Predict wine quality based on 11 predecessor variables

```
# Packages
library(knitr)
library(skimr)
library(Hmisc)
```

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

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

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

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(treemapify)
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
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:Hmisc':
##
##     impute
```

```
library(rpart)
library(multiROC)
library(ROCR)
library(RColorBrewer)
library(ggridges)
library(cowplot)
```

```
##
## ************************************************************
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## ************************************************************
```

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:randomForest':
##
##     combine
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(glue)
```

```
##
## Attaching package: 'glue'
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
```

```
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
library(DMwR)
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# loading dataset

setwd("C:/Users/giese/Desktop/outros_projetos/meus_e_outros_projetos prontos/vinhos")
getwd()
```

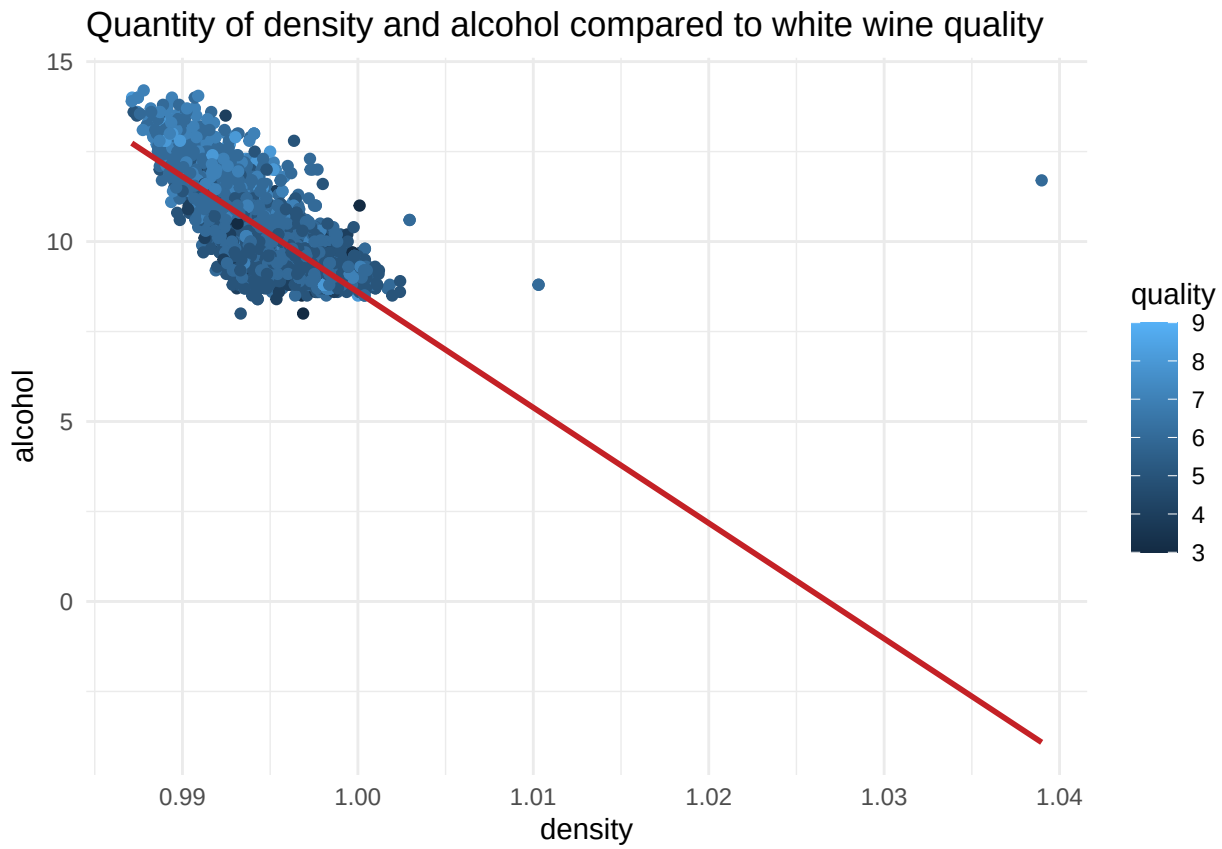
```
## [1] "C:/Users/giese/Desktop/outros_projetos/meus_e_outros_projetos prontos/vinhos"
```

```
vinho_vermelho <- read.csv("winequality-red.csv", header = TRUE, sep = ";")

vinho_branco <- read.csv("winequality-white.csv", header = TRUE, sep = ";")

str(vinho_branco)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
```

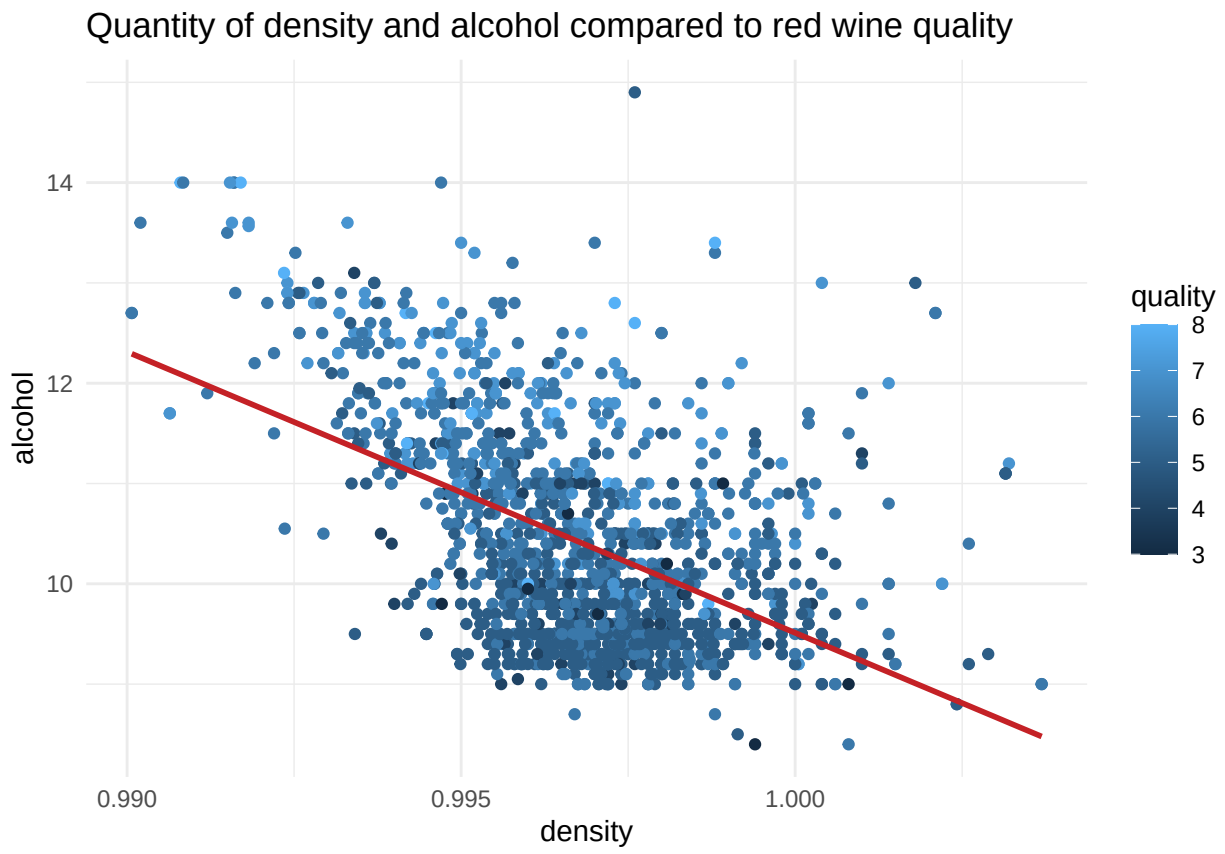
```
#### exploratory analysis ####

# analysis of variables density & alcohol with scatter plot white wine & red wine

a <- ggplot(vinho_branco, aes(x = density, y = alcohol)) +
  geom_point(aes(color = quality)) +
  theme_minimal() +
  stat_smooth(method = "lm",
              col = "#C42126",
              se = FALSE,
              size = 1) +
  ggtitle("Quantity of density and alcohol compared to white wine quality")
a
```

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



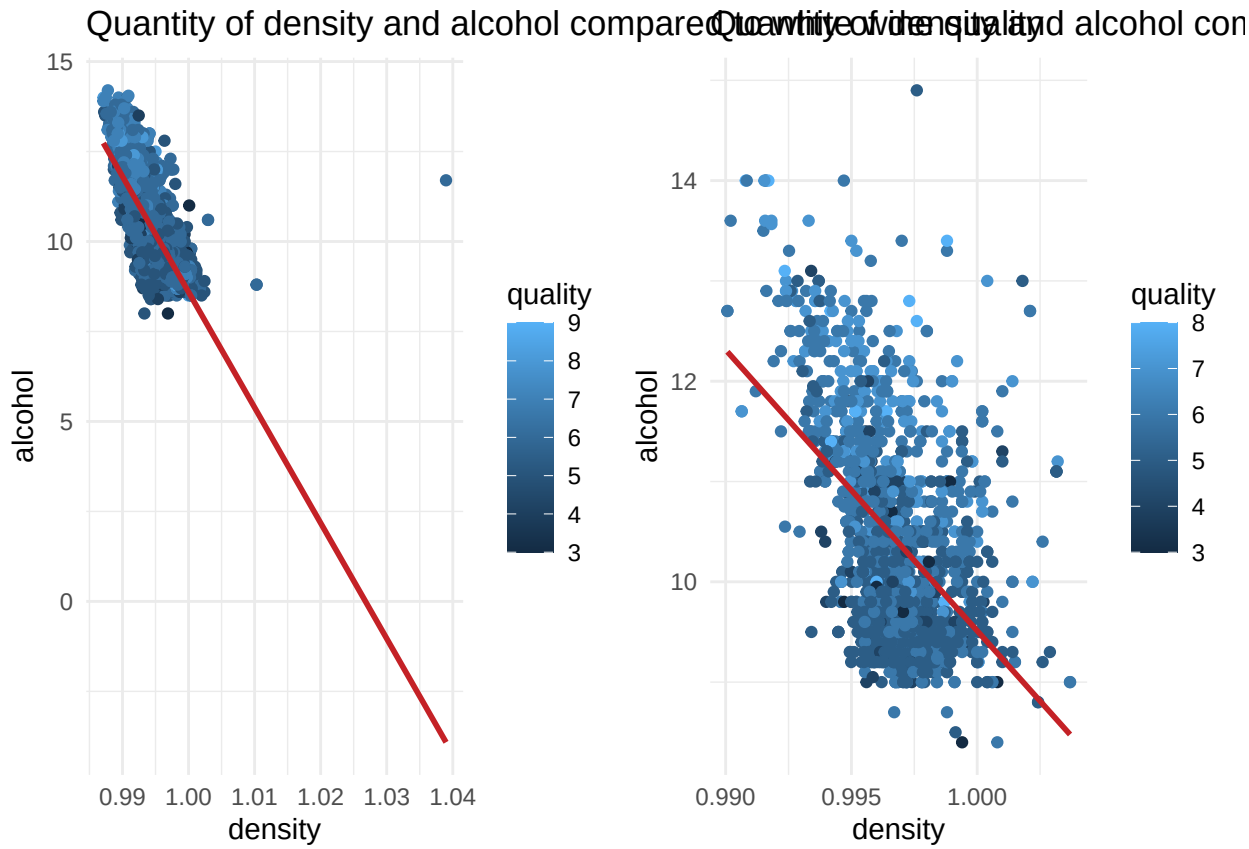

```
b <- ggplot(vinho_vermelho, aes(x = density, y = alcohol)) +
  geom_point(aes(color = quality)) +
  theme_minimal() +
  stat_smooth(method = "lm",
              col = "#C42126",
              se = FALSE,
              size = 1) +
  ggtitle("Quantity of density and alcohol compared to red wine quality")
b
```

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

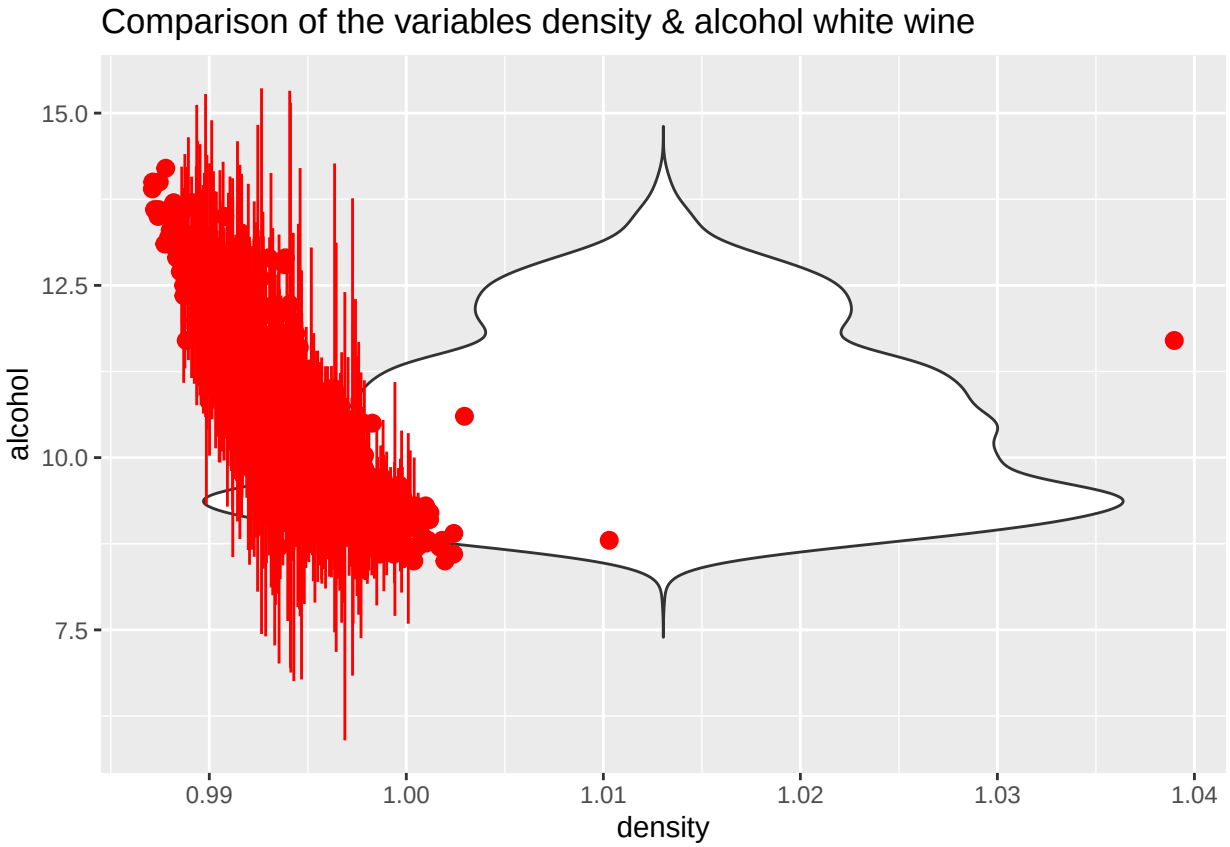



```
plot_grid(a, b, ncol = 2, nrow = 1)
```

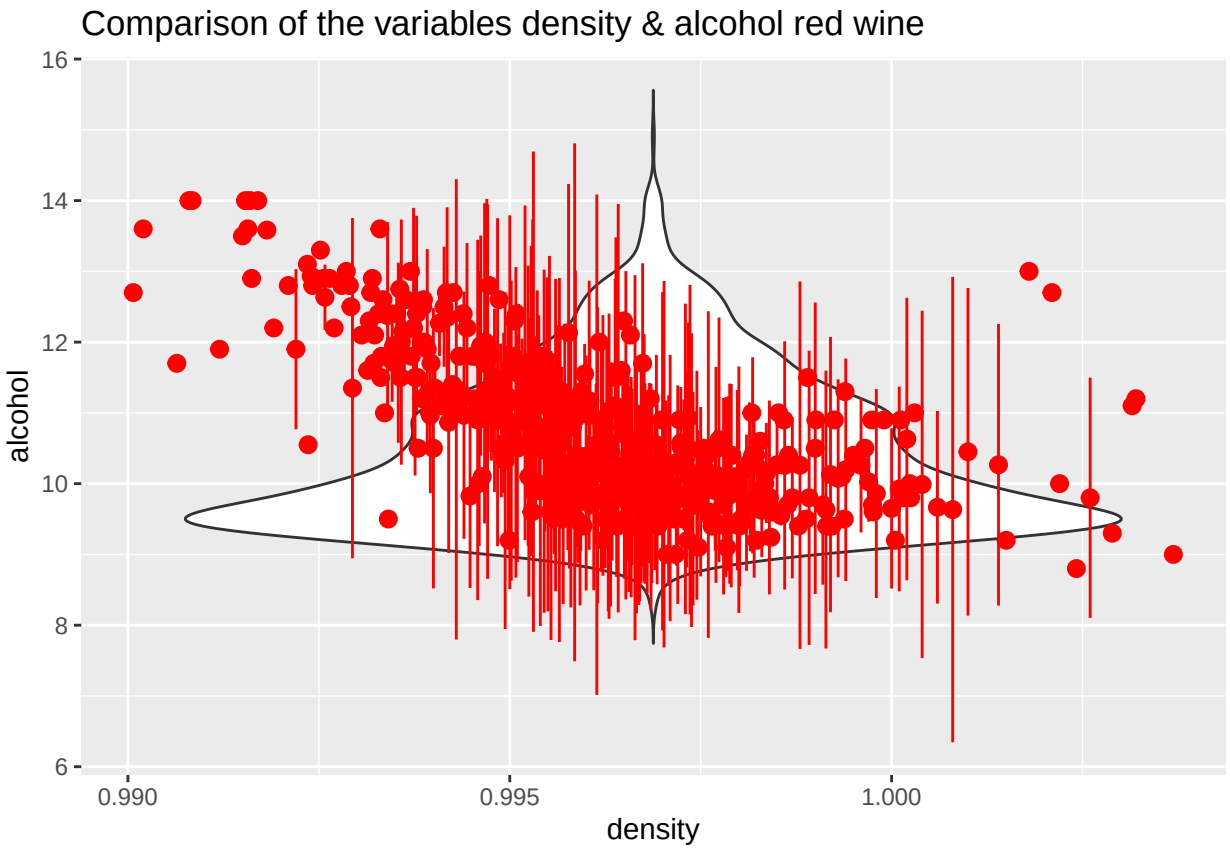
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```




```
# analysis of variables density & alcohol with violin plot white wine & red wine

c <- ggplot(vinho_branco, aes(x=density, y=alcohol)) +
  geom_violin(trim = FALSE) +
  stat_summary(fun.data = mean_sdl,
               geom="pointrange", color="red") +
  ggtitle("Comparison of the variables density & alcohol white wine")
c
```



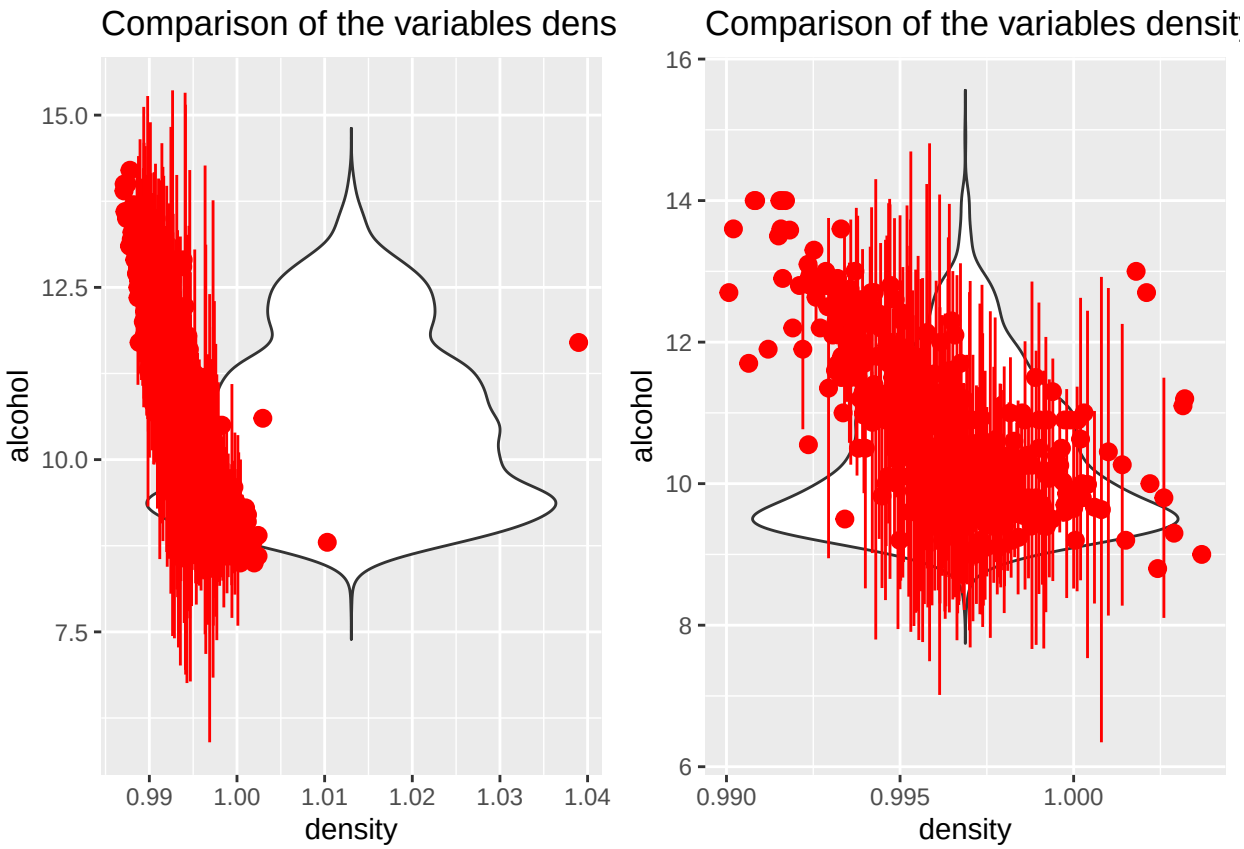

```
d <- ggplot(vinho_vermelho, aes(x=density, y=alcohol)) +
  geom_violin(trim = FALSE) +
  stat_summary(fun.data = mean_sdl,
               geom="pointrange", color="red") +
  ggtitle("Comparison of the variables density & alcohol red wine")
d
```



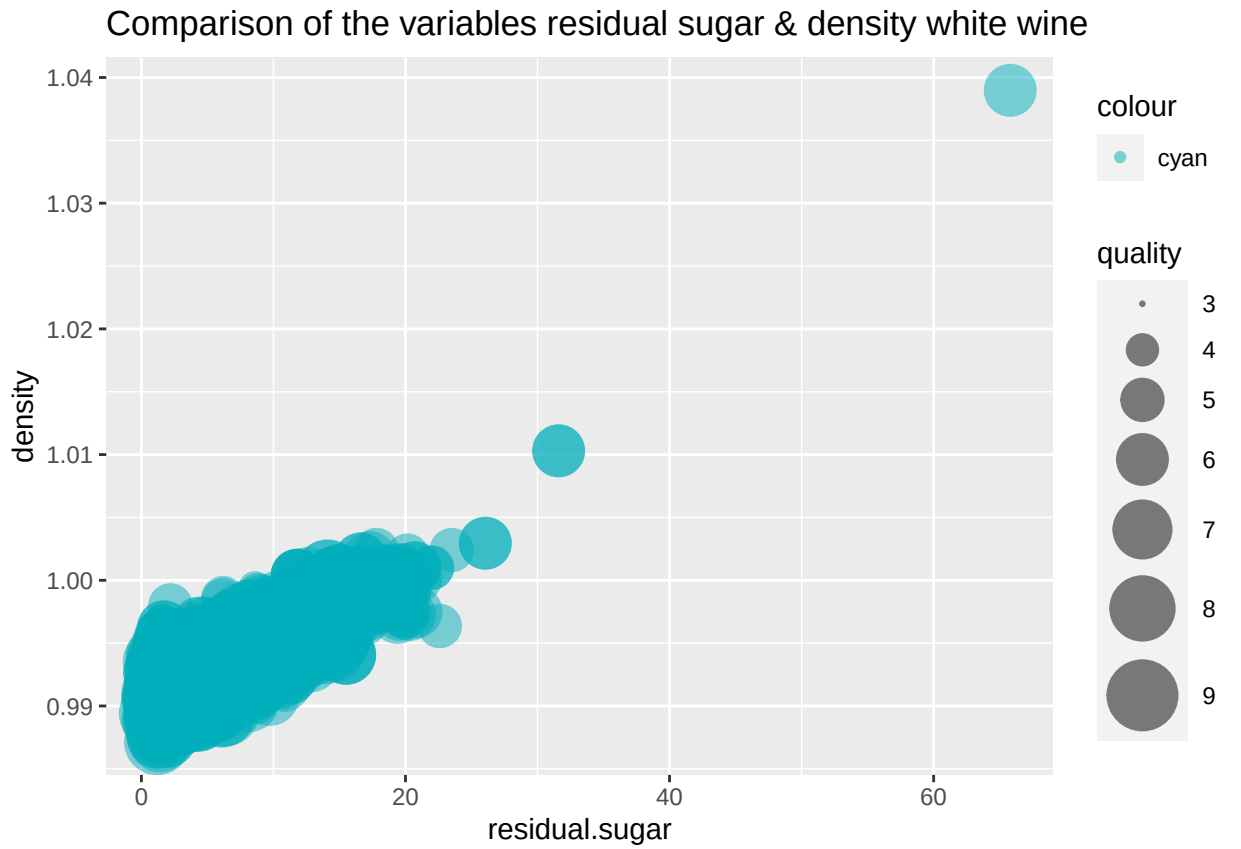

```
plot_grid(c, d, ncol = 2, nrow = 1)
```

```
## Warning: Removed 237 rows containing missing values (geom_segment).
```

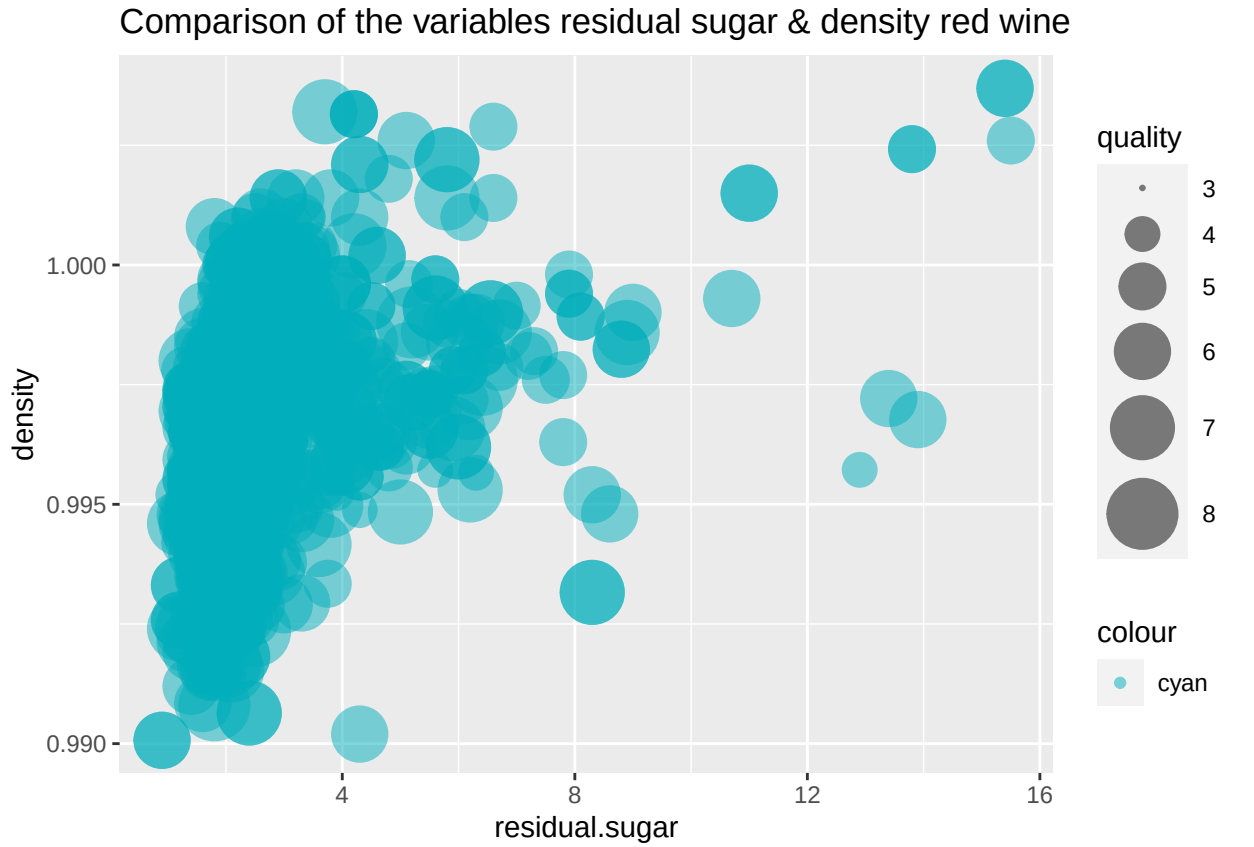
```
## Warning: Removed 167 rows containing missing values (geom_segment).
```



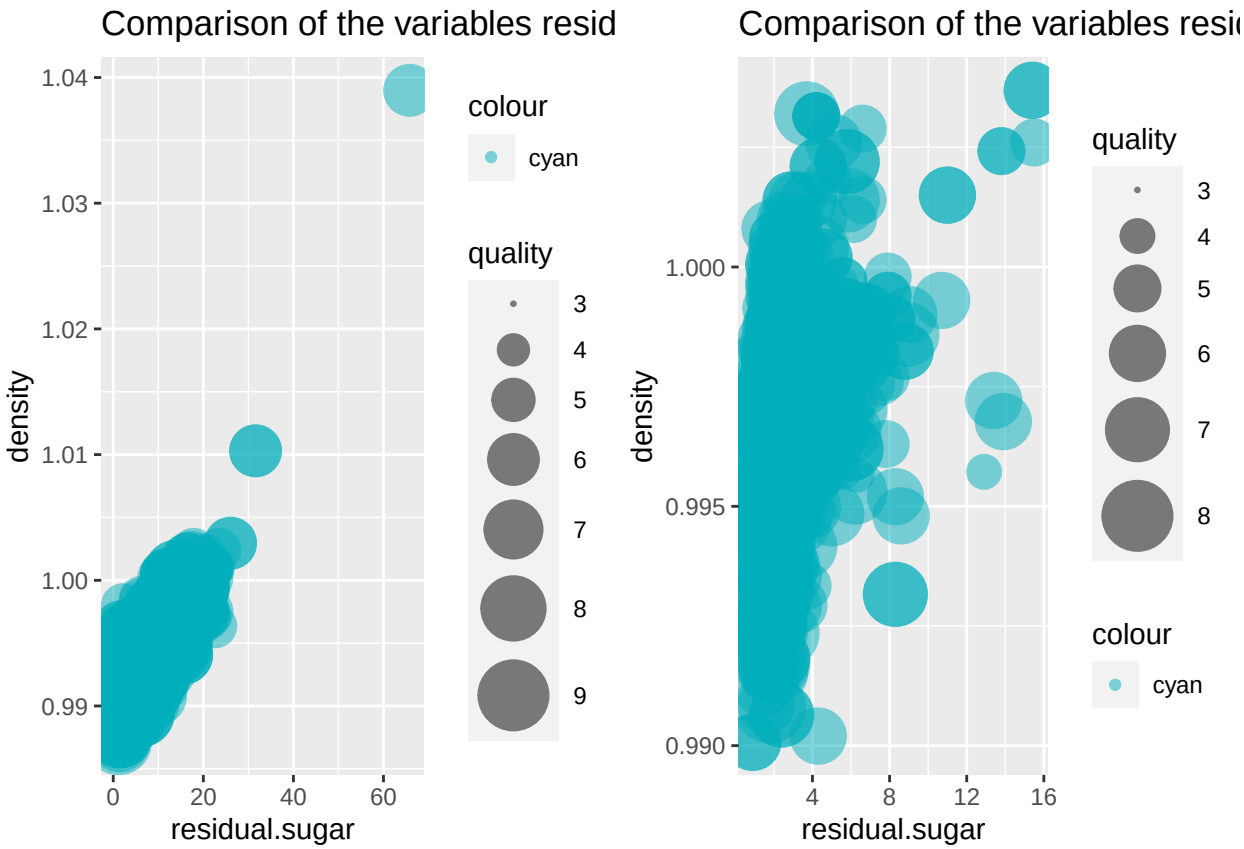

```
# analysis of variables residual.sugar & density with bubbleplot white wine & red wine

e <- ggplot(vinho_branco, aes(x = residual.sugar, y = density)) +
  geom_point(aes(color = "cyan", size = quality), alpha = 0.5) +
  scale_color_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +
  scale_size(range = c(0.5, 12)) +
  ggtitle("Comparison of the variables residual sugar & density white wine")
e
```




```
f <- ggplot(vinho_vermelho, aes(x = residual.sugar, y = density)) +
  geom_point(aes(color = "cyan", size = quality), alpha = 0.5) +
  scale_color_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +
  scale_size(range = c(0.5, 12)) +
  ggtitle("Comparison of the variables residual sugar & density red wine")
f
```

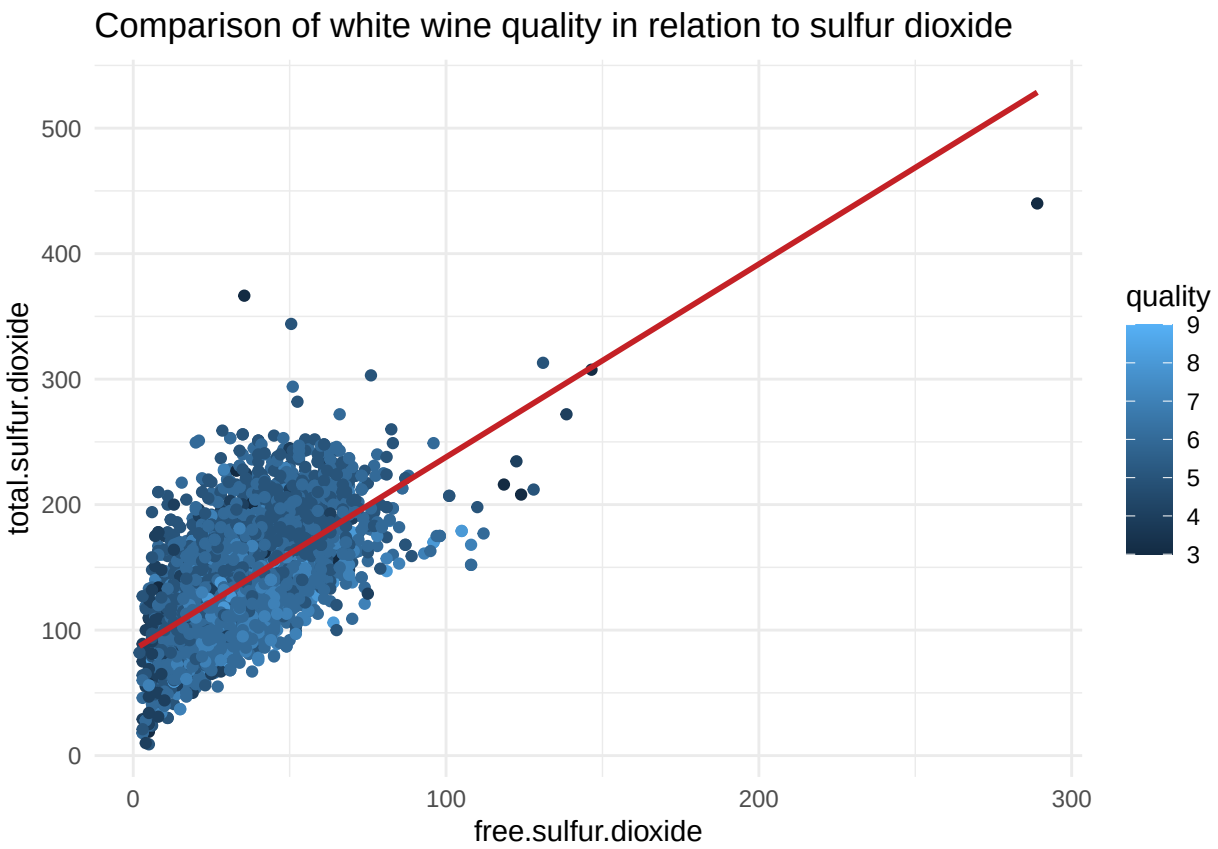



```
plot_grid(e, f, ncol = 2, nrow = 1)
```




```
# analysis of variables free.sulfur.dioxide & total.sulfur.dioxide with scatterplot white & red wine

g <- ggplot(vinho_branco, aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide)) +
  geom_point(aes(color = quality)) +
  theme_minimal() +
  stat_smooth(method = "lm",
              col = "#C42126",
              se = FALSE,
              size = 1) +
  ggtitle("Comparison of white wine quality in relation to sulfur dioxide")

g
```

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

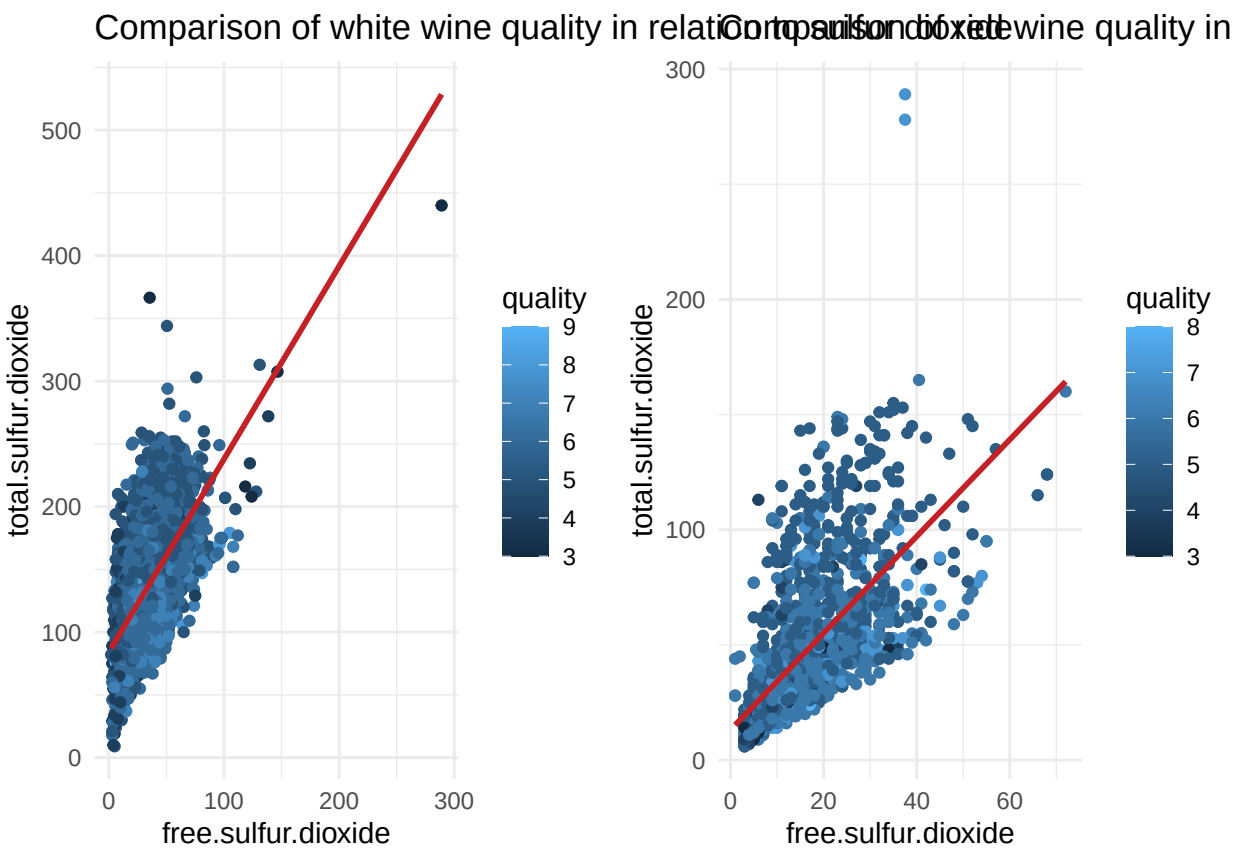



```
h <- ggplot(vinho_vermelho, aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide)) +
  geom_point(aes(color = quality)) +
  theme_minimal() +
  stat_smooth(method = "lm",
              col = "#C42126",
              se = FALSE,
              size = 1) +
  ggtitle("Comparison of red wine quality in relation to sulfur dioxide")

h
```

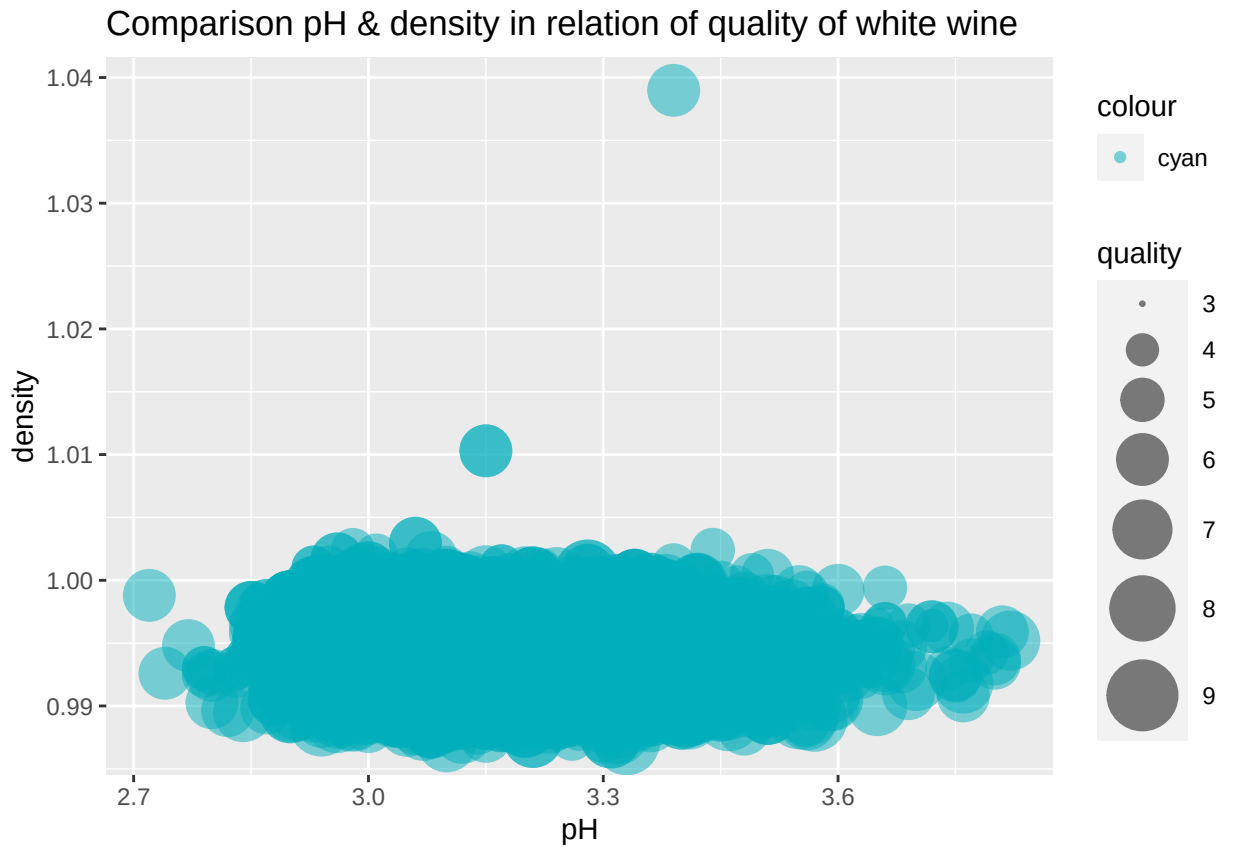
```
## `geom_smooth()` using formula 'y ~ x'
```


```
plot_grid(g, h, ncol = 2, nrow = 1)
```

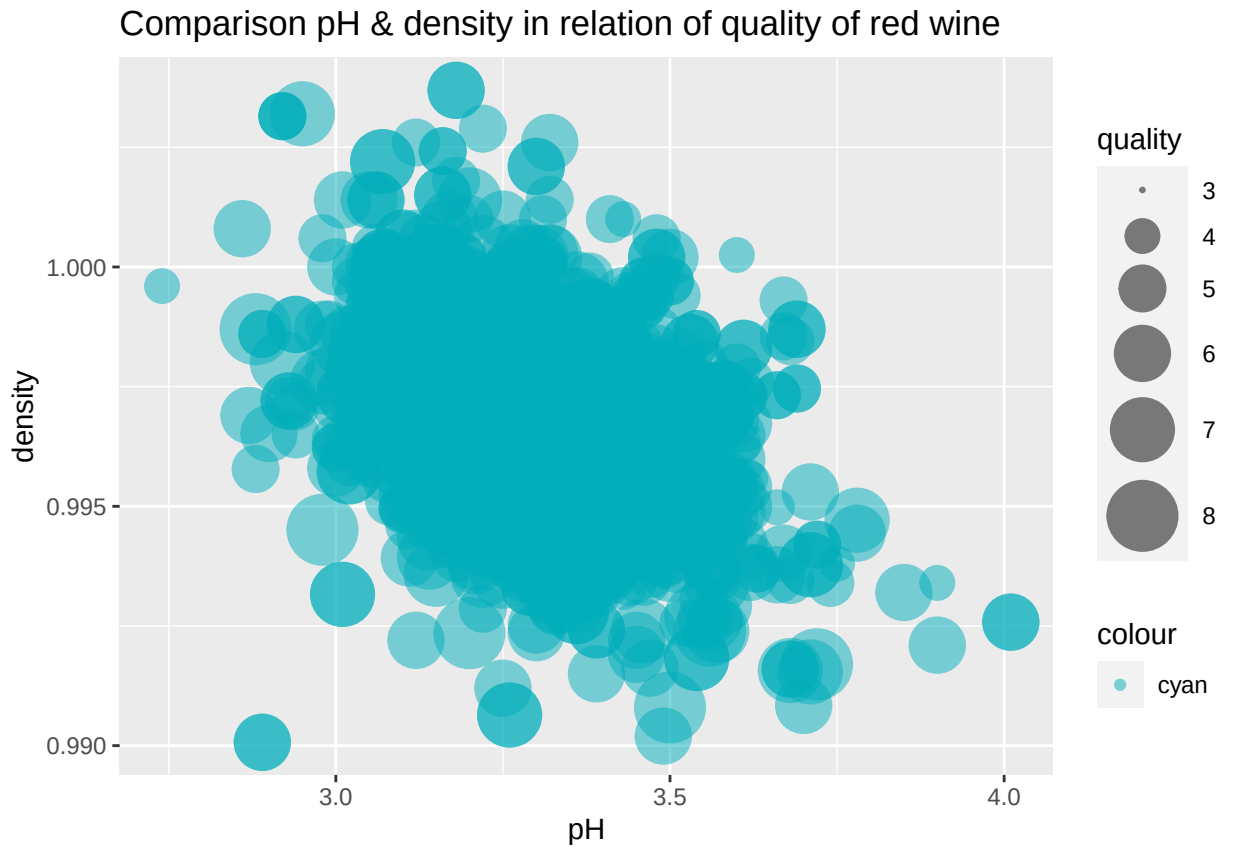
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



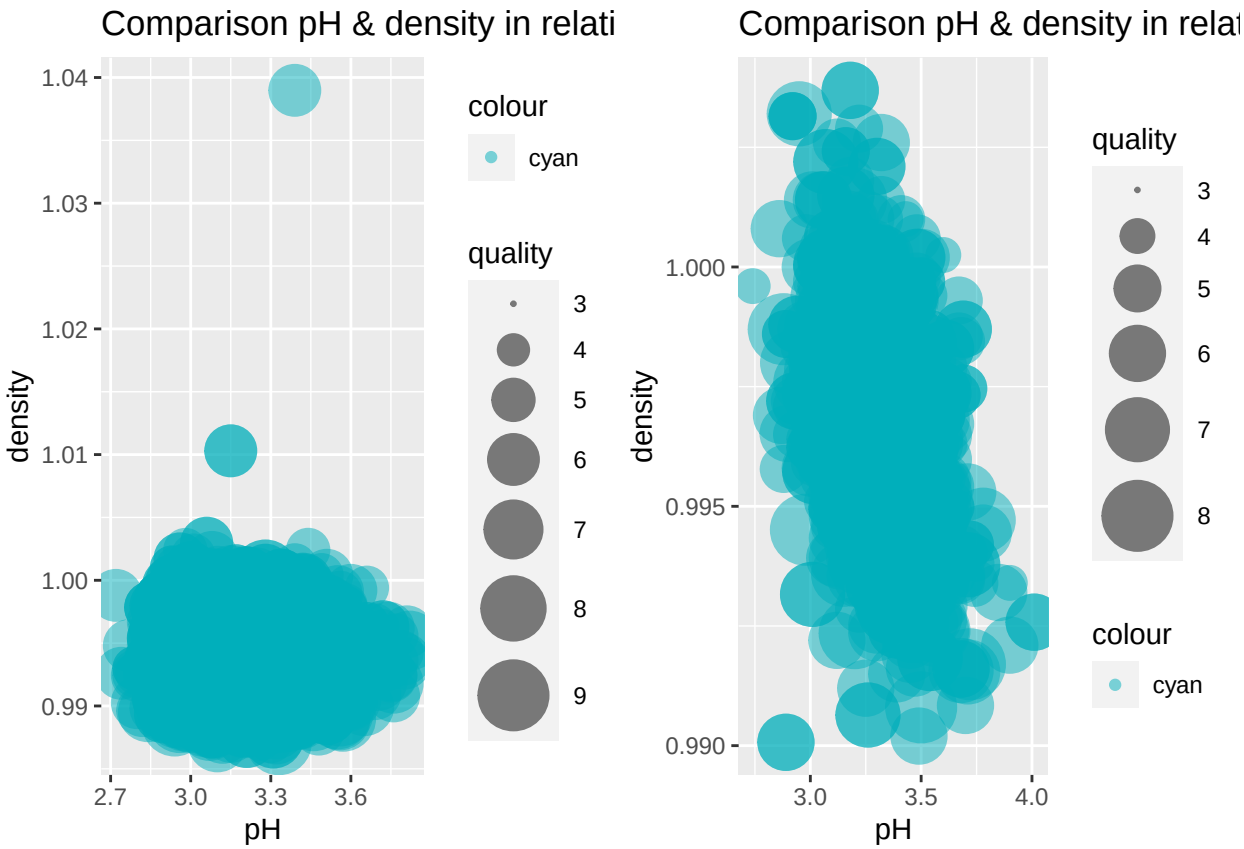

```
# analysis of variables ph & density with bubbleplot white wine & red wine

i <- ggplot(vinho_branco, aes(x = pH, y = density)) +
  geom_point(aes(color = "cyan", size = quality), alpha = 0.5) +
  scale_color_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +
  scale_size(range = c(0.5, 12)) +
  ggtitle("Comparison pH & density in relation of quality of white wine")

i
```




```
j <- ggplot(vinho_vermelho, aes(x = pH, y = density)) +
  geom_point(aes(color = "cyan", size = quality), alpha = 0.5) +
  scale_color_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +
  scale_size(range = c(0.5, 12)) +
  ggtitle("Comparison pH & density in relation of quality of red wine")

j
```




```
plot_grid(i, j, ncol = 2, nrow = 1)
```



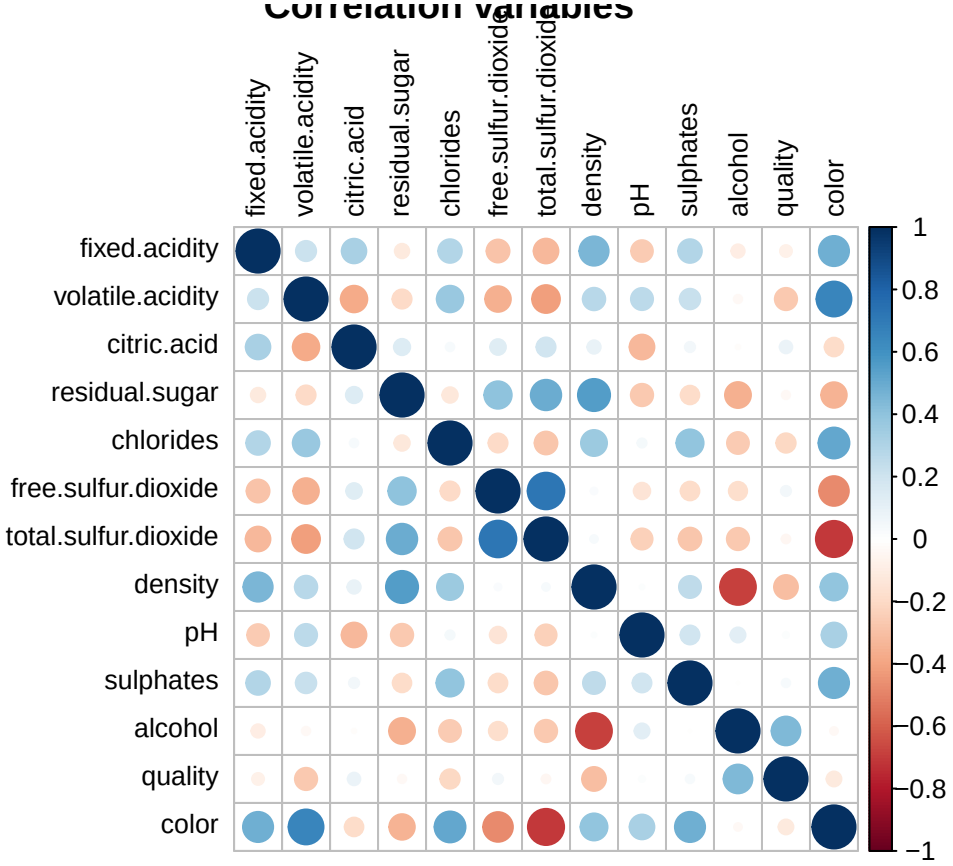

```
#### pre processing ####

# creating new variable for red wine & white wine

color = 0

vinho_branco <- cbind(vinho_branco, color)

color = 1

vinho_vermelho <- cbind(vinho_vermelho, color)
```

```
# merging white wine & red wine datasets

df_vinhos <- merge(vinho_branco, vinho_vermelho, all = TRUE)

str(df_vinhos)
```

```
## 'data.frame':    6497 obs. of  13 variables:
##  $ fixed.acidity       : num  3.8 3.9 4.2 4.2 4.4 4.4 4.4 4.5 4.6 4.6 ...
##  $ volatile.acidity    : num  0.31 0.225 0.17 0.215 0.32 0.46 0.54 0.19 0.445 0.52 ...
##  $ citric.acid         : num  0.02 0.4 0.36 0.23 0.39 0.1 0.09 0.21 0 0.15 ...
##  $ residual.sugar      : num  11.1 4.2 1.8 5.1 4.3 2.8 5.1 0.95 1.4 2.1 ...
##  $ chlorides           : num  0.036 0.03 0.029 0.041 0.03 0.024 0.038 0.033 0.053 0.054 ...
##  $ free.sulfur.dioxide : num  20 29 93 64 31 31 52 89 11 8 ...
##  $ total.sulfur.dioxide: num  114 118 161 157 127 111 97 159 178 65 ...
##  $ density             : num  0.992 0.989 0.99 0.997 0.989 ...
```

```
##  $ pH                  : num  3.75 3.57 3.65 3.42 3.46 3.48 3.41 3.34 3.79 3.9 ...
##  $ sulphates           : num  0.44 0.36 0.89 0.44 0.36 0.34 0.4 0.42 0.55 0.56 ...
##  $ alcohol             : num  12.4 12.8 12 8 12.8 13.1 12.2 8 10.2 13.1 ...
##  $ quality             : int  6 8 7 3 8 6 7 5 5 4 ...
##  $ color               : num  0 0 0 0 0 0 0 0 0 1 ...
```

```
# correlation of variables wine
correlations <- cor(df_vinhos,method="pearson")

corrplot(correlations, number.cex = .9, method = "circle", type = "full", tl.cex=0.8,tl.col = "black",
         title = "Correlation variables")
```

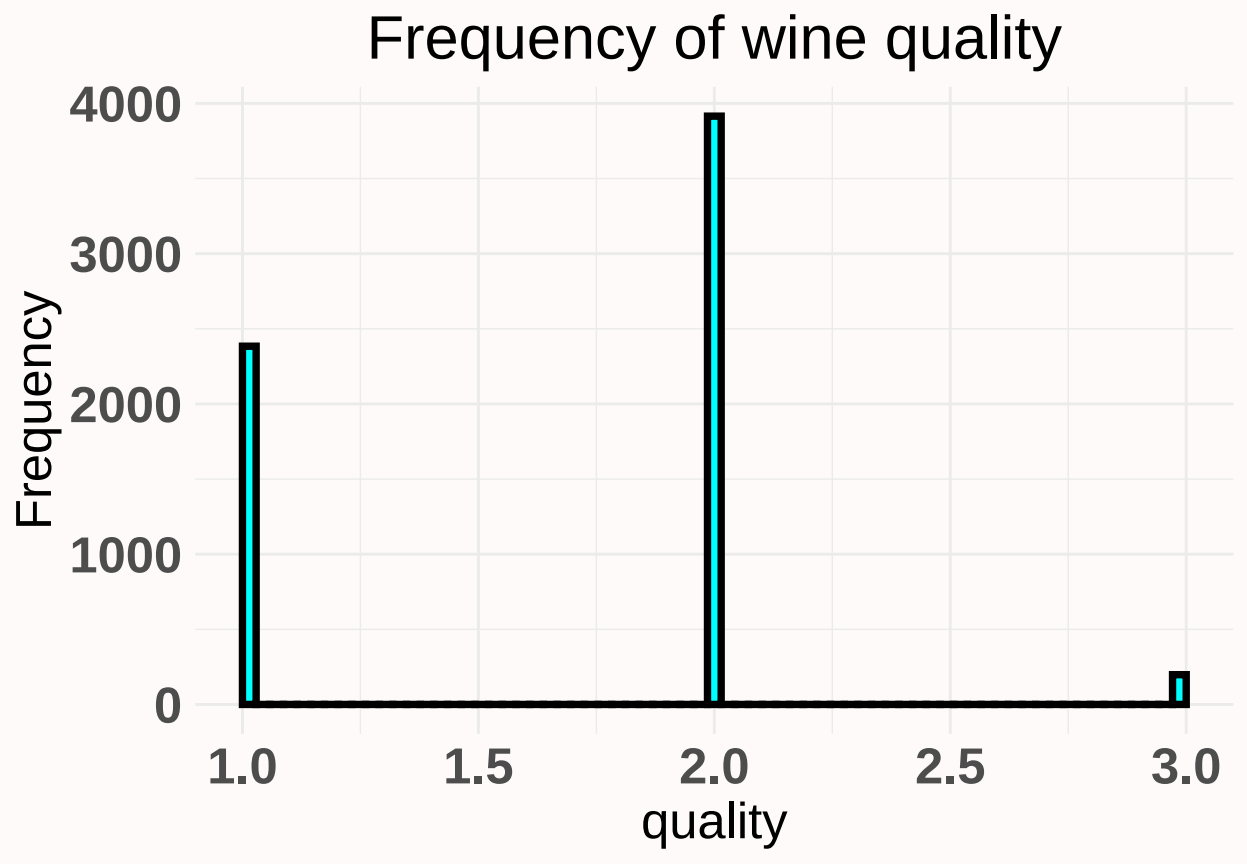



```
# creating new target variable with quality column
quality <- cut(df_vinhos$quality, breaks = c(3), labels = c(0,1,2))

df_vinhos$quality <- NULL

df_vinhos_final <- cbind(df_vinhos, quality)

str(df_vinhos_final)
```

```
## 'data.frame':    6497 obs. of  13 variables:
##  $ fixed.acidity       : num  3.8 3.9 4.2 4.2 4.4 4.4 4.4 4.5 4.6 4.6 ...
##  $ volatile.acidity    : num  0.31 0.225 0.17 0.215 0.32 0.46 0.54 0.19 0.445 0.52 ...
##  $ citric.acid         : num  0.02 0.4 0.36 0.23 0.39 0.1 0.09 0.21 0 0.15 ...
##  $ residual.sugar      : num  11.1 4.2 1.8 5.1 4.3 2.8 5.1 0.95 1.4 2.1 ...
##  $ chlorides           : num  0.036 0.03 0.029 0.041 0.03 0.024 0.038 0.033 0.053 0.054 ...
```

```
##  $ free.sulfur.dioxide : num  20 29 93 64 31 31 52 89 11 8 ...
##  $ total.sulfur.dioxide: num  114 118 161 157 127 111 97 159 178 65 ...
##  $ density             : num  0.992 0.989 0.99 0.997 0.989 ...
##  $ pH                  : num  3.75 3.57 3.65 3.42 3.46 3.48 3.41 3.34 3.79 3.9 ...
##  $ sulphates           : num  0.44 0.36 0.89 0.44 0.36 0.34 0.4 0.42 0.55 0.56 ...
##  $ alcohol             : num  12.4 12.8 12 8 12.8 13.1 12.2 8 10.2 13.1 ...
##  $ color               : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ quality             : Factor w/ 3 levels "0","1","2": 2 3 2 1 3 2 2 1 1 1 ...
```

```r
# Histogram of ratings white wine

tema <- theme(plot.background = element_rect(fill = "#FFFAFA", color = "#FFFAFA"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=14, repr.plot.height=6)
a <- ggplot(data = df_vinhos_final, mapping = aes(x = as.integer(quality))) +
  geom_histogram(fill = "cyan", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("quality") +
  ggtitle("Frequency of wine quality") +
  tema

a
```




```
# analysis of frequency of new variable target

freq_notas_vinho <- data.frame(cbind(frequency = table(df_vinhos_final$quality, useNA = NULL),
                                     percent = prop.table(table(df_vinhos_final$quality)) * 100))

freq_notas_vinho
```

```
##   frequency  percent
## 0      2384 36.69386
## 1      3915 60.25858
## 2       198  3.04756
```

```
# Feature selection

modelo_vinho <- randomForest(quality ~., data = df_vinhos_final)

importance(modelo_vinho)
```

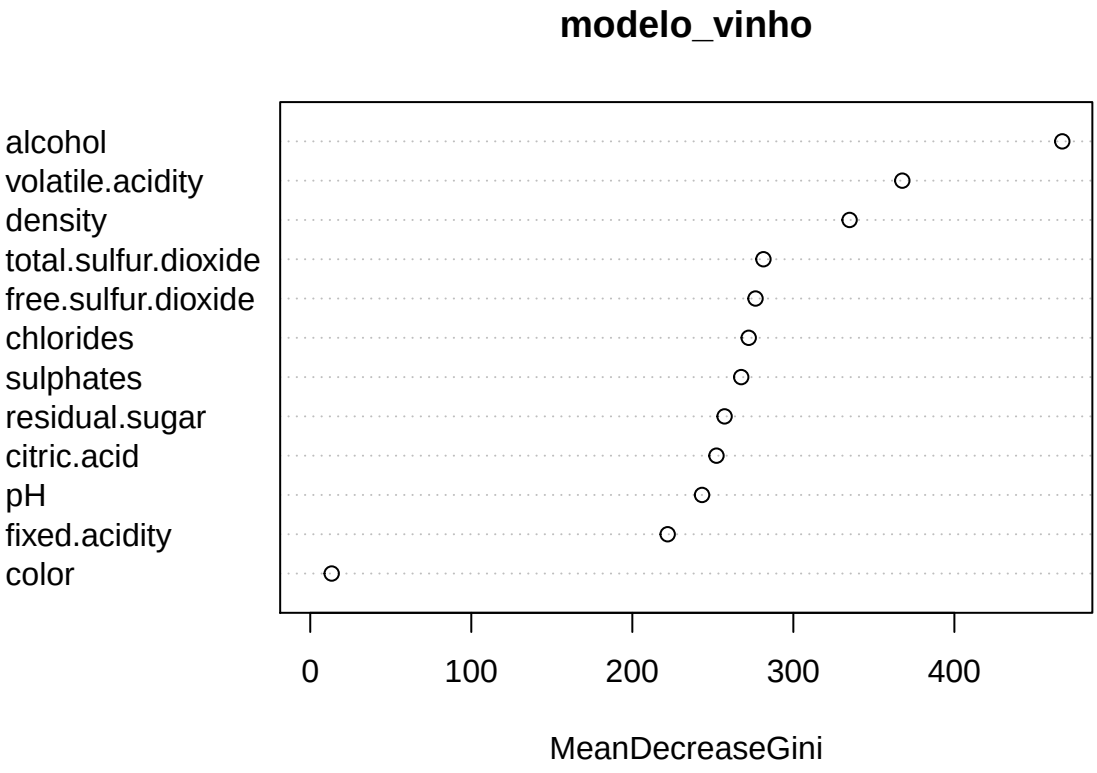
```
##                       MeanDecreaseGini
## fixed.acidity                221.94008
## volatile.acidity             367.65027
## citric.acid                  252.25689
## residual.sugar               257.28581
## chlorides                    272.27262
## free.sulfur.dioxide          276.50214
## total.sulfur.dioxide         281.41568
## density                      334.91078
```

```
## pH                          243.29086
## sulphates                   267.61226
## alcohol                     466.99359
## color                        13.30847
```

```
varImp(modelo_vinho)
```

```
##                        Overall
## fixed.acidity        221.94008
## volatile.acidity     367.65027
## citric.acid          252.25689
## residual.sugar       257.28581
## chlorides            272.27262
## free.sulfur.dioxide  276.50214
## total.sulfur.dioxide 281.41568
## density              334.91078
## pH                   243.29086
## sulphates            267.61226
## alcohol              466.99359
## color                 13.30847
```

```
varImpPlot(modelo_vinho)
```




```
# spliting data into training and test

indexes <- sample(1:nrow(df_vinhos_final), size = 0.7 * nrow(df_vinhos_final))
train.data.vinho <- df_vinhos_final[indexes,]
test.data.vinho <- df_vinhos_final[-indexes,]
```

```r
class(train.data.vinho)
```

```
## [1] "data.frame"
```

```r
class(test.data.vinho)
```

```
## [1] "data.frame"
```

```r
str(train.data.vinho)
```

```
## 'data.frame':    4547 obs. of  13 variables:
##  $ fixed.acidity       : num  7.2 10 5 7.6 7.1 6.9 7.3 6.2 10.2 9.6 ...
##  $ volatile.acidity    : num  0.2 0.32 0.2 0.28 0.34 0.21 0.18 0.22 0.44 0.23 ...
##  $ citric.acid         : num  0.34 0.59 0.4 0.39 0.2 0.33 0.65 0.28 0.88 0.4 ...
##  $ residual.sugar      : num  2.7 2.2 1.9 1.9 6.1 1.4 1.4 2.2 6.2 1.5 ...
##  $ chlorides           : num  0.032 0.077 0.015 0.052 0.063 0.056 0.046 0.04 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  49 3 20 23 47 35 28 24 20 19 ...
##  $ total.sulfur.dioxide: num  151 15 98 116 164 136 157 125 124 135 ...
##  $ density             : num  0.99 0.999 0.99 0.994 0.995 ...
##  $ pH                  : num  3.16 3.2 3.37 3.25 3.17 3.63 3.33 3.19 2.99 2.96 ...
##  $ sulphates           : num  0.39 0.78 0.55 0.4 0.42 0.78 0.62 0.48 0.51 0.49 ...
##  $ alcohol             : num  12.7 9.6 12.1 10.4 10 ...
##  $ color               : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ quality             : Factor w/ 3 levels "0","1","2": 2 1 2 2 1 2 2 2 1 1 ...
```

```r
prop.table(table(train.data.vinho$quality)) * 100
```

```
##
##         0         1         2
## 36.793490 60.149549  3.056961
```

```r
# balancing target variable with SMOTE

train.data.vinho.balanced <- SMOTE(quality ~ ., train.data.vinho, perc.over = 1000, perc.under = 300)

# checking balanced target

train.data.vinho.balanced <- na.omit(train.data.vinho.balanced)

prop.table(table(train.data.vinho.balanced$quality)) * 100
```

```
##
##        0        1        2
## 27.54869 45.62204 26.82927
```

```r
#### Machine learning ####

#svm 64% accuracy with SMOTE 1000 & 300
set.seed(123)

modelo_ma_branco <- svm(quality ~. ,data = train.data.vinho.balanced)
summary(modelo_ma_branco)
```

```
##
## Call:
## svm(formula = quality ~ ., data = train.data.vinho.balanced)
##
##
```

```
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  4320
##
##  ( 1185 2081 1054 )
##
##
## Number of Classes:  3
##
## Levels:
##  0 1 2
```

```
print(modelo_ma_branco)
```

```
##
## Call:
## svm(formula = quality ~ ., data = train.data.vinho.balanced)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  4320
```

```
# prevision of quality of wine svm

modelo_pred_branco <- predict(modelo_ma_branco, test.data.vinho)



table(modelo_pred_branco, test.data.vinho$quality)
```

```
##
## modelo_pred_branco   0   1   2
##                  0 448 191   0
##                  1 250 828  26
##                  2  13 161  33
```

```
confusionMatrix(modelo_pred_branco, test.data.vinho$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2
##          0 448 191   0
##          1 250 828  26
##          2  13 161  33
##
## Overall Statistics
##
##                Accuracy : 0.6713
```

```
##                  95% CI : (0.6499, 0.6921)
##     No Information Rate : 0.6051
##     P-Value [Acc > NIR] : 8.738e-10
##
##                   Kappa : 0.3852
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity            0.6301   0.7017  0.55932
## Specificity            0.8458   0.6416  0.90799
## Pos Pred Value         0.7011   0.7500  0.15942
## Neg Pred Value         0.7994   0.5839  0.98508
## Prevalence             0.3646   0.6051  0.03026
## Detection Rate         0.2297   0.4246  0.01692
## Detection Prevalence   0.3277   0.5662  0.10615
## Balanced Accuracy      0.7380   0.6716  0.73365
```

```
# prevision of quality of wine sum

modelo_pred_branco <- predict(modelo_ma_branco, test.data.vinho)



table(modelo_pred_branco, test.data.vinho$quality)
```

```
##
## modelo_pred_branco   0   1   2
##                  0 448 191   0
##                  1 250 828  26
##                  2  13 161  33
```

```
confusionMatrix(modelo_pred_branco, test.data.vinho$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2
##          0 448 191   0
##          1 250 828  26
##          2  13 161  33
##
## Overall Statistics
##
##                Accuracy : 0.6713
##                  95% CI : (0.6499, 0.6921)
##     No Information Rate : 0.6051
##     P-Value [Acc > NIR] : 8.738e-10
##
##                   Kappa : 0.3852
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity            0.6301   0.7017  0.55932
## Specificity            0.8458   0.6416  0.90799
## Pos Pred Value         0.7011   0.7500  0.15942
## Neg Pred Value         0.7994   0.5839  0.98508
## Prevalence             0.3646   0.6051  0.03026
## Detection Rate         0.2297   0.4246  0.01692
## Detection Prevalence   0.3277   0.5662  0.10615
## Balanced Accuracy      0.7380   0.6716  0.73365
```

```
# naive bayes 51% accuracy

modelo_ma_branco <- naiveBayes(quality~ + alcohol + volatile.acidity + density, train.data.vinho.balanc

summary(modelo_ma_branco)
```

```
##           Length Class  Mode
## apriori   3      table  numeric
## tables    3      -none- list
## levels    3      -none- character
## isnumeric 3      -none- logical
## call      4      -none- call
```

```
# prevision of quality of wine sum

modelo_pred_branco <- predict(modelo_ma_branco, test.data.vinho)



table(modelo_pred_branco, test.data.vinho$quality)
```

```
##
## modelo_pred_branco   0   1   2
##                  0 459 352   7
##                  1 211 485  18
##                  2  41 343  34
```

```
confusionMatrix(modelo_pred_branco, test.data.vinho$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2
##          0 459 352   7
##          1 211 485  18
##          2  41 343  34
##
## Overall Statistics
##
##                Accuracy : 0.5015
##                  95% CI : (0.4791, 0.524)
##     No Information Rate : 0.6051
##     P-Value [Acc > NIR] : 1
##
```

```
##                  Kappa : 0.1947
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity            0.6456   0.4110  0.57627
## Specificity            0.7103   0.7026  0.79693
## Pos Pred Value         0.5611   0.6793  0.08134
## Neg Pred Value         0.7774   0.4377  0.98368
## Prevalence             0.3646   0.6051  0.03026
## Detection Rate         0.2354   0.2487  0.01744
## Detection Prevalence   0.4195   0.3662  0.21436
## Balanced Accuracy      0.6779   0.5568  0.68660
```

```
# svm 2 38% accuracy

modelo_ma_branco <- tune(svm,
                     quality ~.,
                     data = train.data.vinho.balanced,
                     kernel = 'linear',
                     ranges = list(cost = c(0.05, 0.1, 0.5, 1, 2)))

summary(modelo_ma_branco)
```

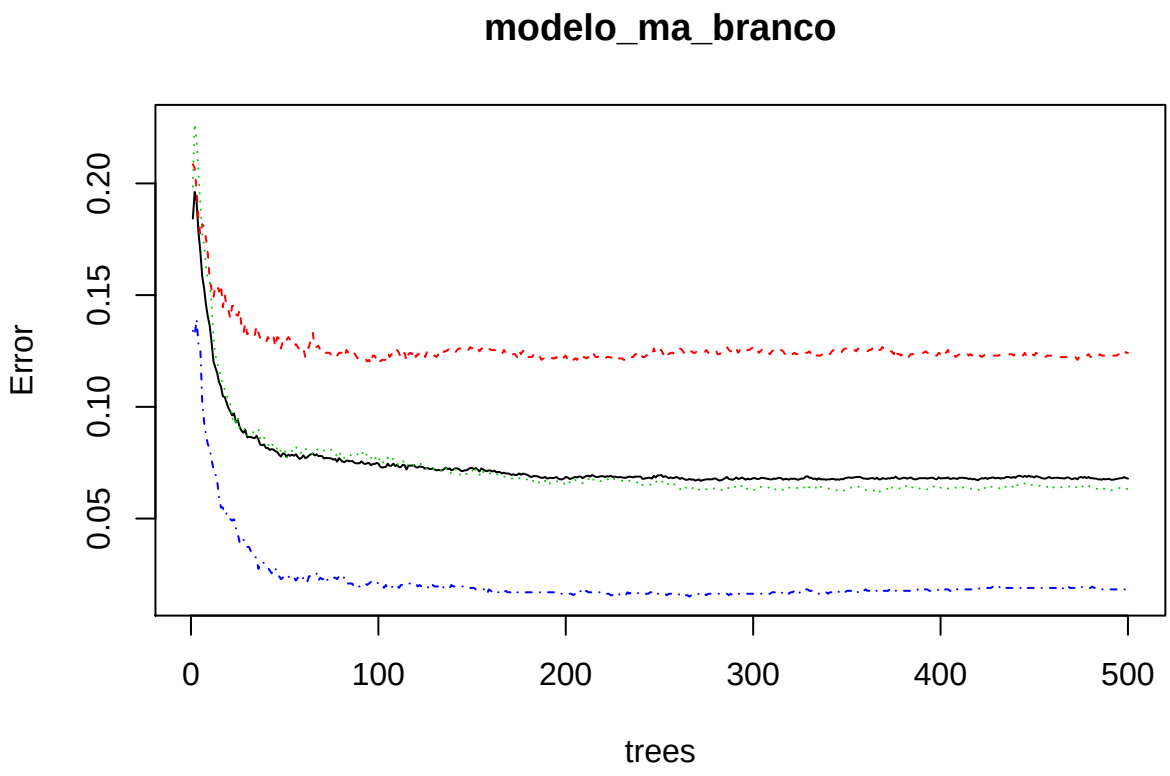
```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     1
##
## - best performance: 0.3888364
##
## - Detailed performance results:
##   cost     error dispersion
## 1 0.05 0.3953282 0.01283380
## 2 0.10 0.3921694 0.01298836
## 3 0.50 0.3891869 0.01240029
## 4 1.00 0.3888364 0.01156750
## 5 2.00 0.3891872 0.01184580
```

```
# RandomForest 76% accuracy

modelo_ma_branco <- randomForest(quality ~., data = train.data.vinho.balanced)

print(modelo_ma_branco)
```

```
##
## Call:
##  randomForest(formula = quality ~ ., data = train.data.vinho.balanced)
##                Type of random forest: classification
```

```
##                       Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 6.79%
## Confusion matrix:
##      0    1    2 class.error
## 0 1375  193    2  0.12420382
## 1  123 2436   41  0.06307692
## 2    0   28 1501  0.01831262
```

```
plot(modelo_ma_branco)
```




```
# prevision of quality of wine

modelo_pred_branco <- predict(modelo_ma_branco, test.data.vinho)



table(modelo_pred_branco, test.data.vinho$quality)
```

```
##
## modelo_pred_branco   0   1   2
##                  0 477 166   0
##                  1 234 963  32
##                  2   0  51  27
```

```
confusionMatrix(modelo_pred_branco, test.data.vinho$quality)
```

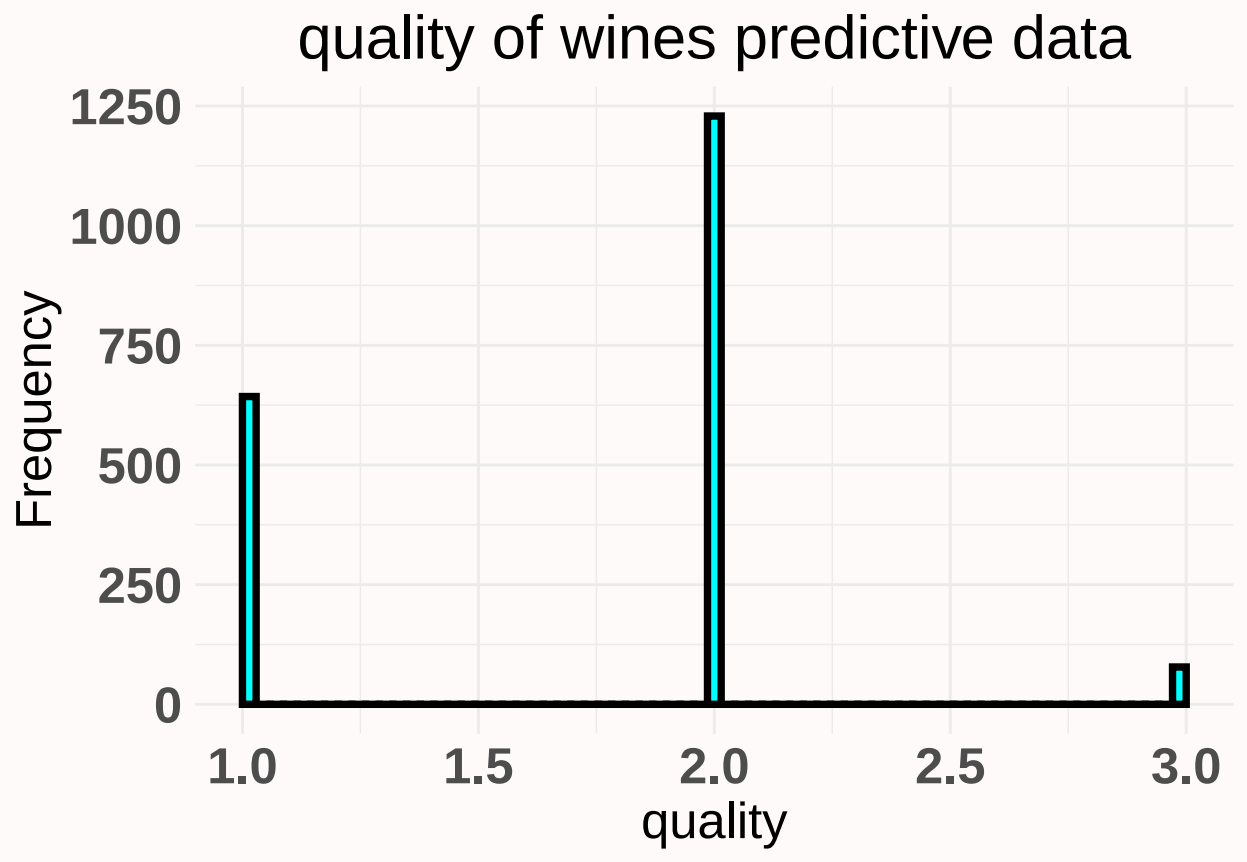
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2
##          0 477 166   0
##          1 234 963  32
##          2   0  51  27
##
## Overall Statistics
##
##                Accuracy : 0.7523
##                  95% CI : (0.7325, 0.7713)
##     No Information Rate : 0.6051
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5018
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity            0.6709   0.8161  0.45763
## Specificity            0.8660   0.6545  0.97303
## Pos Pred Value         0.7418   0.7836  0.34615
## Neg Pred Value         0.8210   0.6990  0.98291
## Prevalence             0.3646   0.6051  0.03026
## Detection Rate         0.2446   0.4938  0.01385
## Detection Prevalence   0.3297   0.6303  0.04000
## Balanced Accuracy      0.7685   0.7353  0.71533
```

```
# quality of wine predictive data

modelo_pred_branco_plot <- as.data.frame(modelo_pred_branco)
names(modelo_pred_branco_plot) <- c("quality")

names(modelo_pred_branco_plot)
```

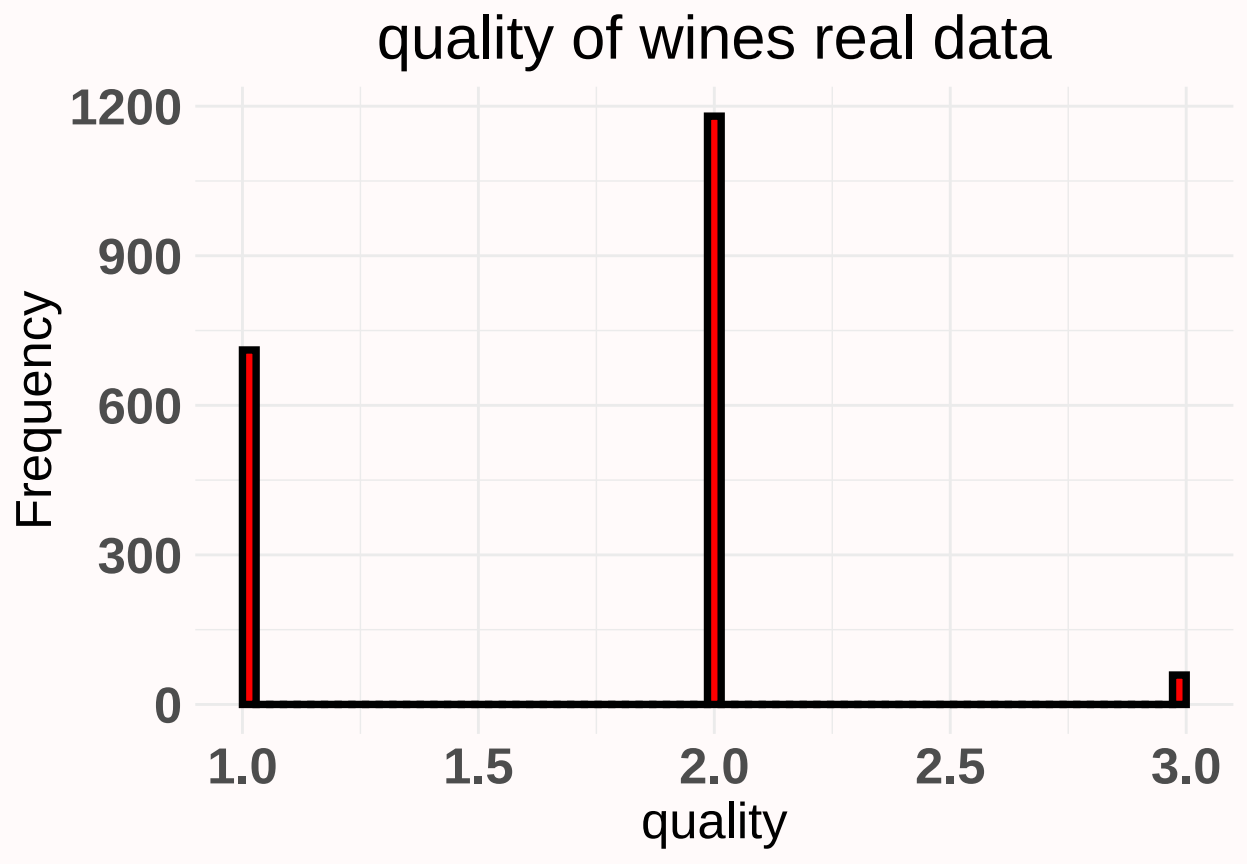
```
## [1] "quality"
```

```
tema <- theme(plot.background = element_rect(fill = "#FFFAFA", color = "#FFFAFA"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=14, repr.plot.height=6)
a <- ggplot(data = modelo_pred_branco_plot , mapping = aes(x = as.numeric(quality))) +
  geom_histogram(fill = "cyan", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("quality") +
  ggtitle("quality of wines predictive data") +
```

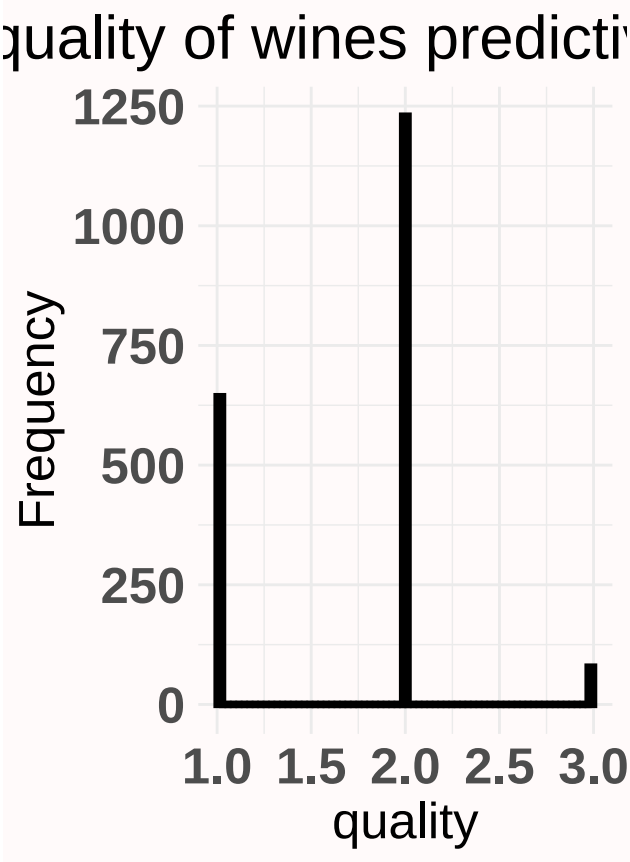
```
  tema

a
```

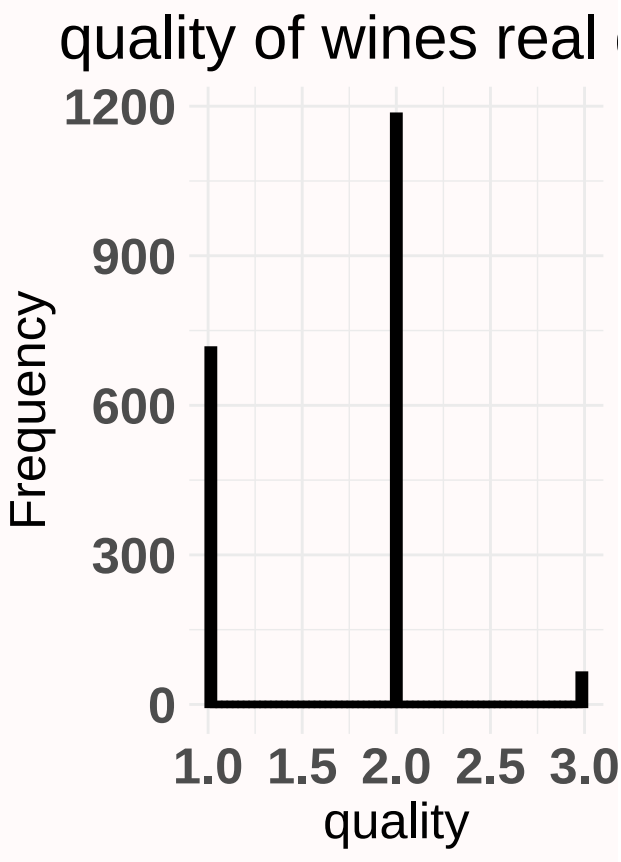



```
# quality of wine real data

options(repr.plot.width=14, repr.plot.height=6)
b <- ggplot(data = test.data.vinho , mapping = aes(x = as.numeric(quality))) +
  geom_histogram(fill = "red", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("quality") +
  ggtitle("quality of wines real data") +
  tema

b
```




```
plot_grid(a, b, ncol = 2, nrow = 1)
```


quality of wines predicti
Frequency
1250
1000
750
500
250
0
1.0
1.5
2.0
2.5
3.0
quality


quality of wines real
Frequency
1200
900
600
300
0
1.0
1.5
2.0
2.5
3.0
quality