Practica 2

Macarena Palomares, Pablo Moreno

2022-05-26

Contents

Cargamos las librerías

```
if(!require('dplyr')) install.packages('dplyr'); library('dplyr')
if(!require('readr')) install.packages('readr'); library('readr')
if(!require('plotrix')) install.packages('plotrix');library(plotrix)
if (!require('kableExtra')) install.packages('kableExtra');library(kableExtra)
if(!require('Rmisc')) install.packages('Rmisc'); library('Rmisc')
if(!require('ggplot2')) install.packages('ggplot2'); library(ggplot2)
if(!require('car')) install.packages('car'); library(car)
if(!require('corrplot')) install.packages('corrplot'); library(corrplot)
if(!require('randomForest')) install.packages('randomForest'); library(randomForest)
if(!require('caret')) install.packages('caret'); library(caret)
if(!require('ROCR')) install.packages('ROCR'); library(ROCR)
```

Lectura del fichero y preparación de los datos

```
path = 'winequality-red.csv'
data_wine <- read_csv(path)

#Comprobamos la estrucutra del dataframe
str(data_wine)
```

```
## spec_tbl_df [1,599 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ..
##  $ free sulfur dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
```

```
##  $ total sulfur dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `fixed acidity` = col_double(),
##   ..   `volatile acidity` = col_double(),
##   ..   `citric acid` = col_double(),
##   ..   `residual sugar` = col_double(),
##   ..   chlorides = col_double(),
##   ..   `free sulfur dioxide` = col_double(),
##   ..   `total sulfur dioxide` = col_double(),
##   ..   density = col_double(),
##   ..   pH = col_double(),
##   ..   sulphates = col_double(),
##   ..   alcohol = col_double(),
##   ..   quality = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
#Visualizamos los 6 primeros elementos cargados
head(data_wine)
```

```
## # A tibble: 6 x 12
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##             <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
## 1             7.4               0.7           0                 1.9     0.076
## 2             7.8               0.88          0                 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               0.7           0                 1.9     0.076
## 6             7.4               0.66          0                 1.8     0.075
## # ... with 7 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <dbl>
```

Nuestros datos tienen 1599 observaciones de 12 variables.

Limpieza de los datos

```
# Reemplazar espacios vacios en los nombres de columnas
names(data_wine) <- gsub(" ", "_", names(data_wine))
```

- ¿Los datos contienen ceros o elementos vacíos?

```
sum(is.na(data_wine))
```

```
## [1] 0
```

Los datos no contienen valores NA. Vemos que ocurre con los valores 0:

```
sum(data_wine[,]==0)
```

```
## [1] 132
```

Tenemos 132 valores a 0. Vemos la distribución por columnas de estos valores:

```
apply(X = data_wine[,1:12] == 0, MARGIN = 2, FUN = sum)
```

```
##        fixed_acidity     volatile_acidity          citric_acid
##                    0                    0                  132
##       residual_sugar            chlorides  free_sulfur_dioxide
##                    0                    0                    0
## total_sulfur_dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
```

Los 132 valores a 0 se encuentran en la columna **citric_acid**.

Vemos la distribución de valores 0 de la variable **citric_acid** con respecto a los niveles de calidad del vino:

```
valores_0 <-data_wine[data_wine$citric_acid==0,]
mytable<-table(valores_0$quality)
kable(mytable, digits=5) %>%
  kable_styling(full_width = T) %>%
  column_spec(col = 1, background="steelblue", bold=T, color="white") %>%
  row_spec(row = 0,color="blue")
```

Var1	Freq
	3
	10
	57
	54
	8

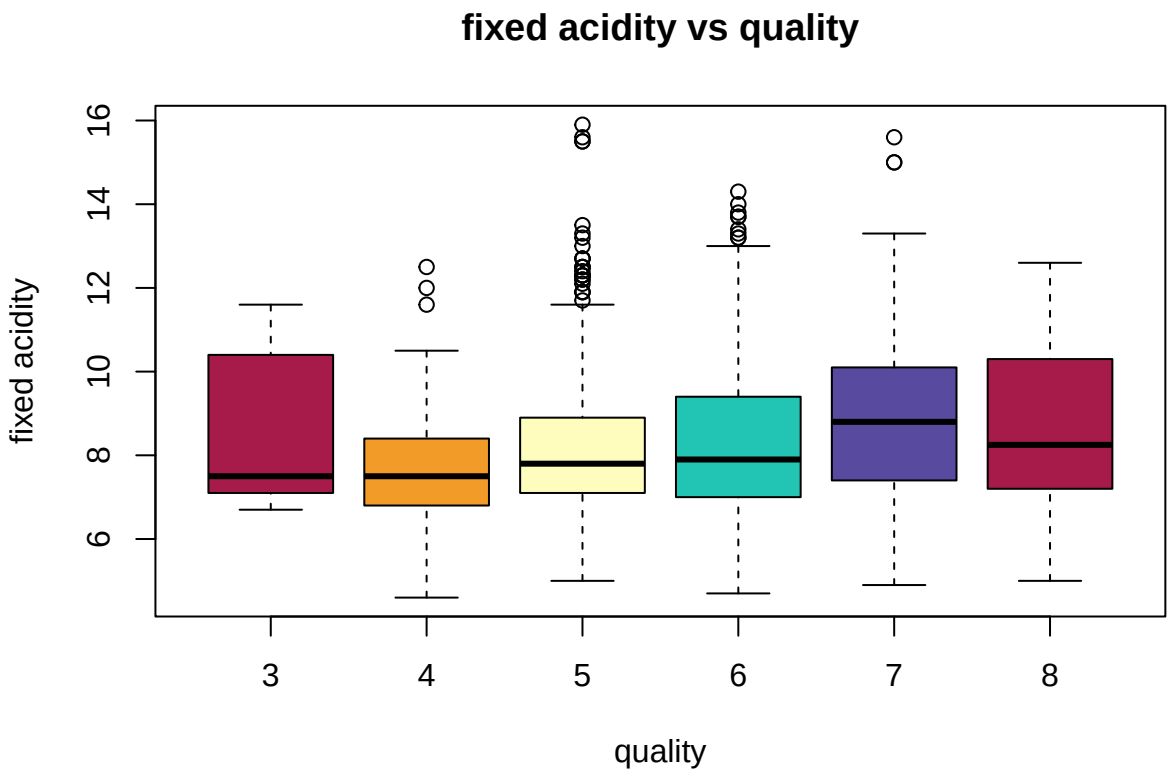
Valores extremos para cada variable:

- Fixed acidity:

```
summary(data_wine$`fixed acidity`)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
g_caja<-boxplot(data_wine$fixed_acidity~data_wine$quality ,main="fixed acidity vs quality", xlab="quali
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##      [,1] [,2] [,3] [,4] [,5]  [,6]
## [1,]  6.7  4.6  5.0  4.7  4.9  5.00
## [2,]  7.1  6.8  7.1  7.0  7.4  7.20
## [3,]  7.5  7.5  7.8  7.9  8.8  8.25
## [4,] 10.4  8.4  8.9  9.4 10.1 10.30
## [5,] 11.6 10.5 11.6 13.0 13.3 12.60
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##          [,1]     [,2]     [,3]     [,4]     [,5]    [,6]
## [1,] 5.851188 7.152752 7.691018 7.749873 8.497591 7.09553
## [2,] 9.148812 7.847248 7.908982 8.050127 9.102409 9.40447
##
## $out
##  [1] 12.5 11.6 12.0 12.5 13.5 11.9 12.5 12.7 12.3 12.3 12.5 13.0 12.4 11.9 11.9
## [16] 15.5 15.5 15.6 12.7 12.7 12.3 12.3 11.7 13.2 15.9 12.1 13.3 12.2 12.2 13.3
## [31] 13.4 13.8 14.0 13.7 13.7 14.3 13.2 13.2 15.0 15.0 15.6
```

```
##
## $group
##  [1] 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4
## [39] 5 5 5
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

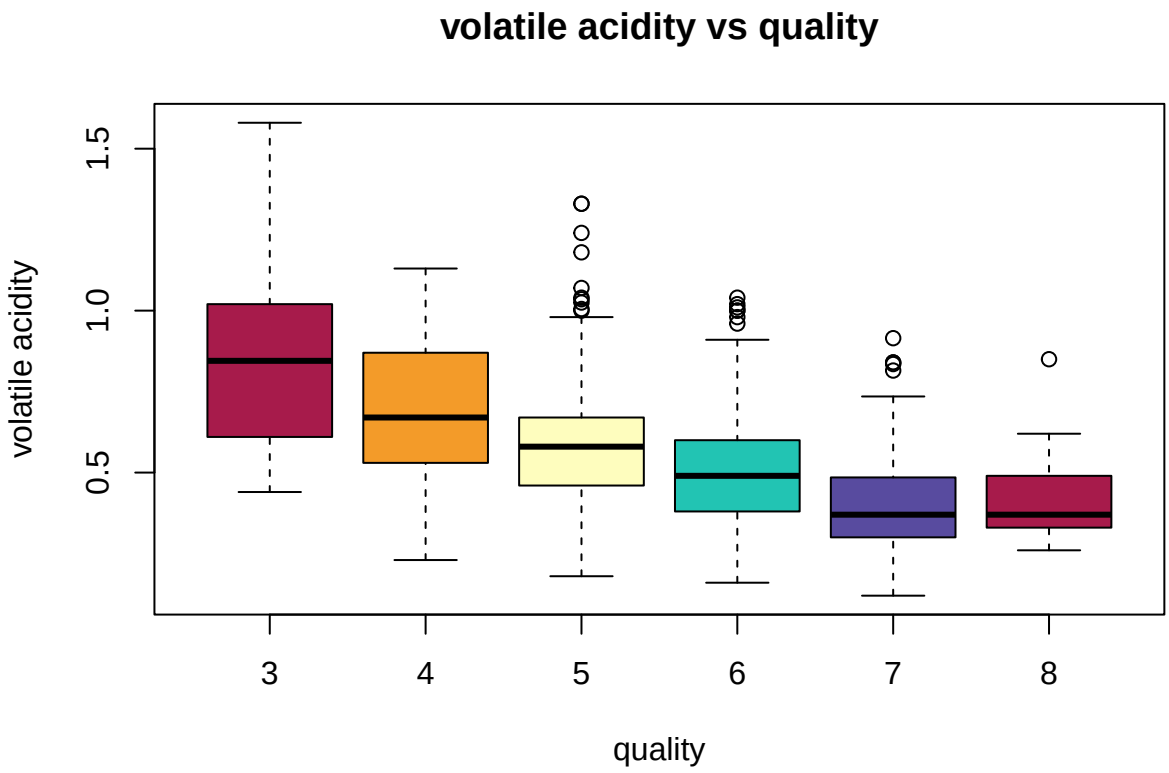
```
##  [1] 12.5 11.6 12.0 12.5 13.5 11.9 12.5 12.7 12.3 12.3 12.5 13.0 12.4 11.9 11.9
## [16] 15.5 15.5 15.6 12.7 12.7 12.3 12.3 11.7 13.2 15.9 12.1 13.3 12.2 12.2 13.3
## [31] 13.4 13.8 14.0 13.7 13.7 14.3 13.2 13.2 15.0 15.0 15.6
```

- Volatile acidity:

```
summary(data_wine$volatile_acidity )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1200  0.3900  0.5200  0.5278  0.6400  1.5800
```

```
g_caja<-boxplot(data_wine$volatile_acidity~data_wine$quality ,main="volatile acidity vs quality", xlab=
```



```r
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##       [,1] [,2] [,3] [,4]  [,5] [,6]
## [1,] 0.440 0.23 0.18 0.16 0.120 0.26
## [2,] 0.610 0.53 0.46 0.38 0.300 0.33
## [3,] 0.845 0.67 0.58 0.49 0.370 0.37
## [4,] 1.020 0.87 0.67 0.60 0.485 0.49
## [5,] 1.580 1.13 0.98 0.91 0.735 0.62
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.6401477 0.5962099 0.5672854 0.4762384 0.3492794 0.3104145
## [2,] 1.0498523 0.7437901 0.5927146 0.5037616 0.3907206 0.4295855
##
## $out
##  [1] 1.070 1.330 1.330 1.040 1.240 1.035 1.025 1.000 1.005 1.180 1.040 1.000
## [13] 1.000 1.020 0.980 1.010 0.960 0.835 0.815 0.840 0.840 0.915 0.850
##
## $group
##  [1] 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5 6
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```r
#outliers
print(valores_outlier)
```

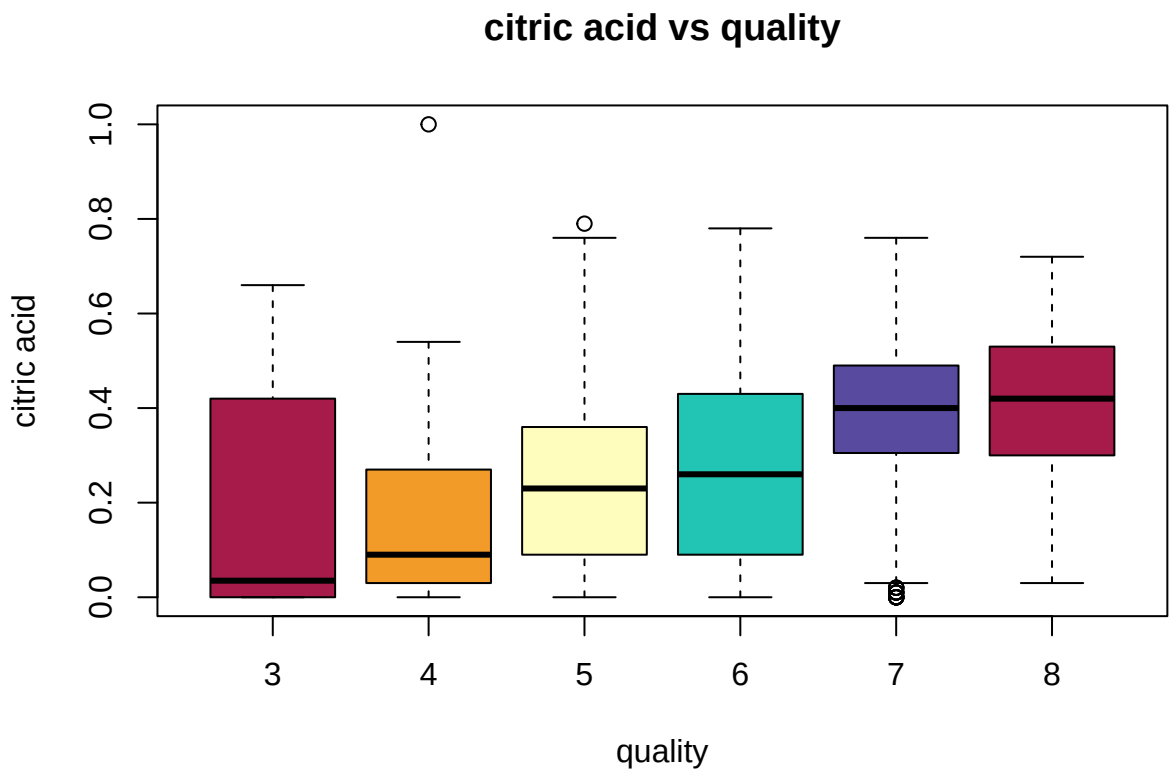
```
##  [1] 1.070 1.330 1.330 1.040 1.240 1.035 1.025 1.000 1.005 1.180 1.040 1.000
## [13] 1.000 1.020 0.980 1.010 0.960 0.835 0.815 0.840 0.840 0.915 0.850
```

- Citric acid:

```r
summary(data_wine$citric_acid )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.090   0.260   0.271   0.420   1.000
```

```r
g_caja<-boxplot(data_wine$citric_acid~data_wine$quality ,main="citric acid vs quality", xlab="quality",
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##       [,1] [,2] [,3] [,4]  [,5] [,6]
## [1,] 0.000 0.00 0.00 0.00 0.030 0.03
## [2,] 0.000 0.03 0.09 0.09 0.305 0.30
## [3,] 0.035 0.09 0.23 0.26 0.400 0.42
## [4,] 0.420 0.27 0.36 0.43 0.490 0.53
## [5,] 0.660 0.54 0.76 0.78 0.760 0.72
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##            [,1]       [,2]      [,3]     [,4]      [,5]      [,6]
## [1,] -0.1748487 0.03791287 0.2136527 0.238732 0.3792794 0.3343458
## [2,]  0.2448487 0.14208713 0.2463473 0.281268 0.4207206 0.5056542
##
## $out
##  [1] 1.00 0.79 0.00 0.02 0.01 0.02 0.01 0.00 0.00 0.02 0.00 0.00 0.01 0.01 0.00
## [16] 0.01 0.00 0.00 0.02
##
## $group
##  [1] 2 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
```

```
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

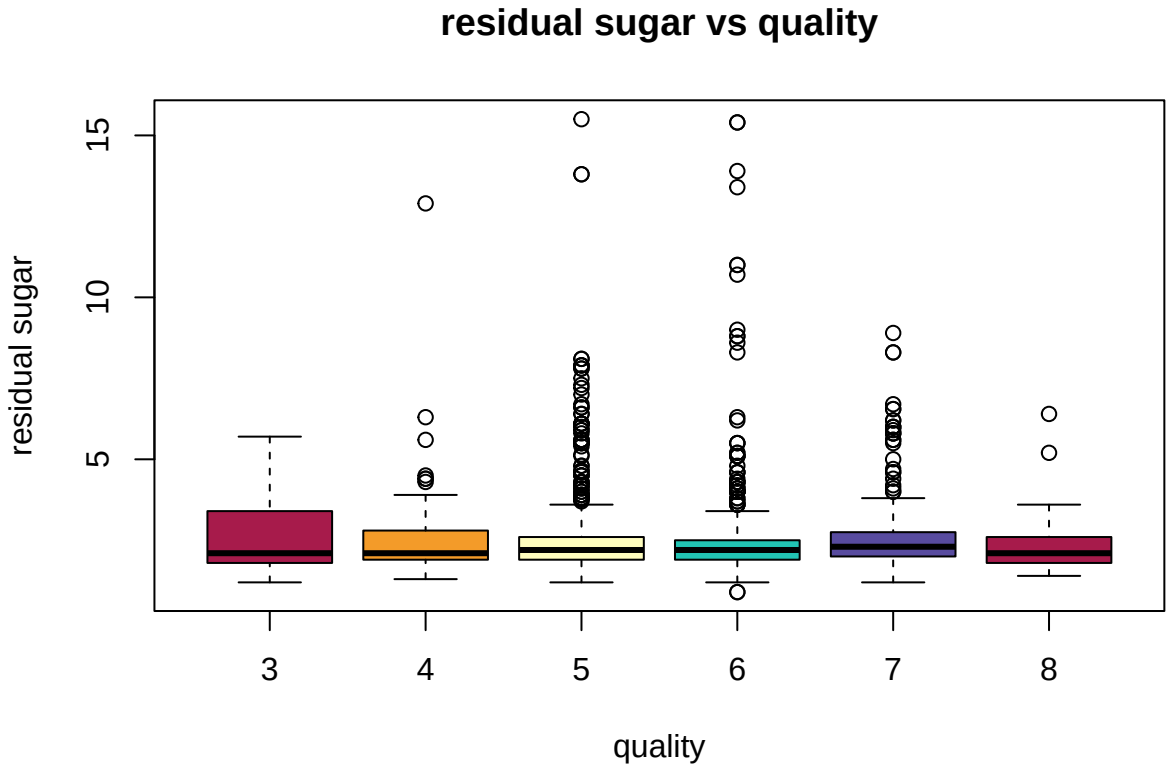
```
##  [1] 1.00 0.79 0.00 0.02 0.01 0.02 0.01 0.00 0.00 0.02 0.00 0.00 0.01 0.01 0.00
## [16] 0.01 0.00 0.00 0.02
```

- Residual sugar:

```
summary(data_wine$residual_sugar)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.900   1.900   2.200   2.539   2.600  15.500
```

```
g_caja<-boxplot(data_wine$residual_sugar~data_wine$quality ,main="residual sugar vs quality", xlab="qua
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]  1.2  1.3  1.2  1.2 1.20  1.4
## [2,]  1.8  1.9  1.9  1.9 2.00  1.8
## [3,]  2.1  2.1  2.2  2.2 2.30  2.1
## [4,]  3.4  2.8  2.6  2.5 2.75  2.6
## [5,]  5.7  3.9  3.6  3.4 3.80  3.6
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 1.300576 1.904673 2.157618 2.162468 2.215998 1.802072
## [2,] 2.899424 2.295327 2.242382 2.237532 2.384002 2.397928
##
## $out
##   [1]  4.40  4.50  4.40  5.60 12.90  4.30  6.30  6.10  6.10  3.80  3.90  5.90
##  [13]  5.90  3.80  5.10  4.65  4.65  5.50  5.50  5.50  5.50  7.30  7.20  3.80
##  [25]  5.60  4.00  7.00  4.50  4.80  7.90  7.90  6.60 15.50  4.60  6.10  5.15
##  [37]  4.20  4.20  4.20  7.90  5.60  5.60  7.50  6.00  3.90  4.00  6.60  4.10
##  [49]  3.90  8.10  8.10  6.40  6.40  3.70  7.80  5.80  4.10  4.50  4.50  4.30
##  [61]  4.30  3.80  5.40  3.80  6.10  3.90  4.80  3.75 13.80 13.80  4.30  6.70
##  [73]  7.80 10.70  5.50  4.00  4.00  4.00  4.00  4.00  3.60 11.00 11.00  3.80
##  [85]  6.20  3.70  4.10  8.30  4.30  6.30  4.60  4.60  4.30  4.30  5.10  8.60
##  [97]  4.20  3.60  3.60  4.00  4.00  3.80  9.00  8.80  8.80  0.90  0.90  5.50
## [109]  5.50  4.30  4.60 13.40  4.80  5.10  5.10 15.40 15.40  5.20  5.20  4.10
## [121]  4.10  4.40  3.70 13.90  5.10  3.60  5.60  5.60  5.80  5.80  4.40  4.20
## [133]  6.70  6.55  6.55  5.80  4.60  6.00  6.00  6.00  4.60  5.00  4.10  5.90
## [145]  6.20  8.90  4.00  4.00  8.30  8.30  4.70  5.50  6.20  6.40  5.20
##
## $group
##   [1] 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4
##  [75] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [112] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [149] 5 5 5 5 5 6 6
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

```
##   [1]  4.40  4.50  4.40  5.60 12.90  4.30  6.30  6.10  6.10  3.80  3.90  5.90
##  [13]  5.90  3.80  5.10  4.65  4.65  5.50  5.50  5.50  5.50  7.30  7.20  3.80
##  [25]  5.60  4.00  7.00  4.50  4.80  7.90  7.90  6.60 15.50  4.60  6.10  5.15
##  [37]  4.20  4.20  4.20  7.90  5.60  5.60  7.50  6.00  3.90  4.00  6.60  4.10
##  [49]  3.90  8.10  8.10  6.40  6.40  3.70  7.80  5.80  4.10  4.50  4.50  4.30
##  [61]  4.30  3.80  5.40  3.80  6.10  3.90  4.80  3.75 13.80 13.80  4.30  6.70
##  [73]  7.80 10.70  5.50  4.00  4.00  4.00  4.00  4.00  3.60 11.00 11.00  3.80
##  [85]  6.20  3.70  4.10  8.30  4.30  6.30  4.60  4.60  4.30  4.30  5.10  8.60
##  [97]  4.20  3.60  3.60  4.00  4.00  3.80  9.00  8.80  8.80  0.90  0.90  5.50
## [109]  5.50  4.30  4.60 13.40  4.80  5.10  5.10 15.40 15.40  5.20  5.20  4.10
```

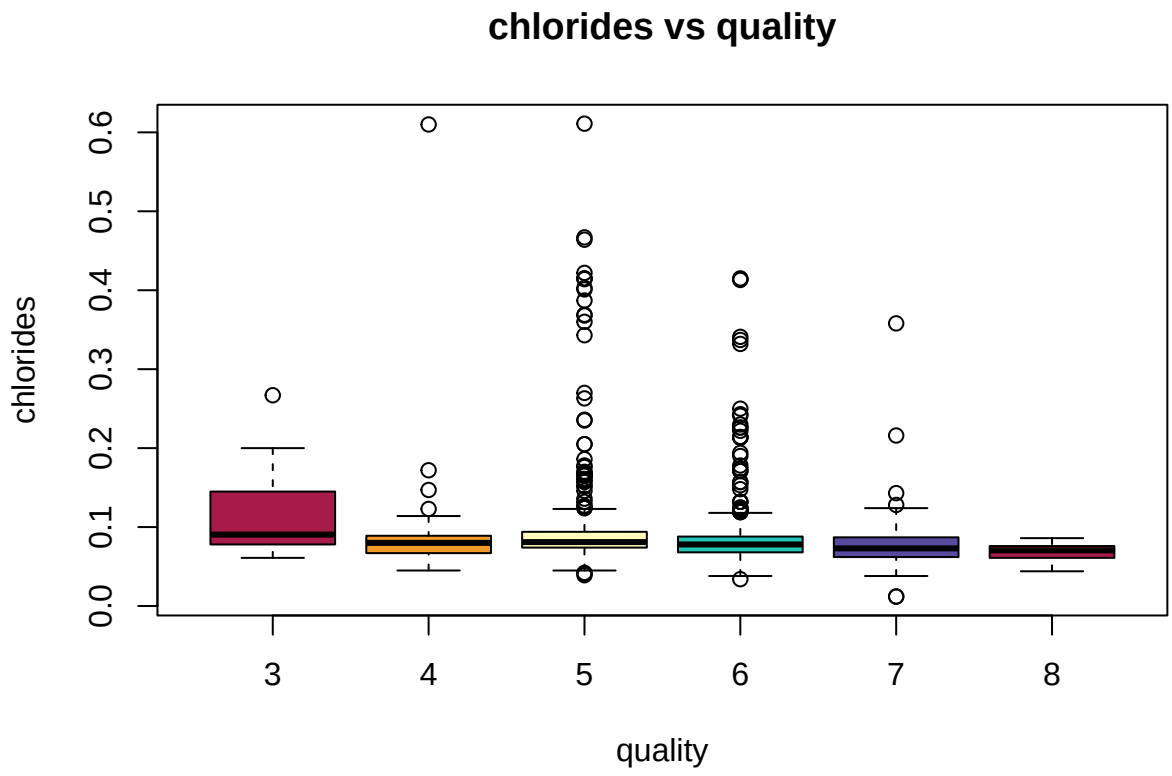
```
## [121]  4.10  4.40  3.70 13.90  5.10  3.60  5.60  5.60  5.80  5.80  4.40  4.20
## [133]  6.70  6.55  6.55  5.80  4.60  6.00  6.00  6.00  4.60  5.00  4.10  5.90
## [145]  6.20  8.90  4.00  4.00  8.30  8.30  4.70  5.50  6.20  6.40  5.20
```

- Chlorides :

```
summary(data_wine$chlorides )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01200 0.07000 0.07900 0.08747 0.09000 0.61100
```

```
g_caja<-boxplot(data_wine$chlorides~data_wine$quality ,main="chlorides vs quality", xlab="quality", ylab
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##        [,1]  [,2]  [,3]  [,4]  [,5]   [,6]
## [1,] 0.0610 0.045 0.045 0.038 0.038 0.0440
## [2,] 0.0780 0.067 0.074 0.068 0.062 0.0610
## [3,] 0.0905 0.080 0.081 0.078 0.073 0.0705
## [4,] 0.1450 0.089 0.094 0.088 0.087 0.0760
## [5,] 0.2000 0.114 0.123 0.118 0.124 0.0860
##
```

```
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] 0.05702413 0.07522535 0.07978909 0.07674894 0.07019992 0.06491386
## [2,] 0.12397587 0.08477465 0.08221091 0.07925106 0.07580008 0.07608614
##
## $out
##  [1] 0.267 0.172 0.610 0.147 0.123 0.176 0.170 0.368 0.464 0.401 0.467 0.178
## [13] 0.146 0.236 0.360 0.270 0.263 0.611 0.343 0.186 0.159 0.127 0.152 0.152
## [25] 0.124 0.124 0.039 0.157 0.422 0.387 0.132 0.126 0.165 0.161 0.414 0.369
## [37] 0.041 0.166 0.166 0.136 0.403 0.166 0.168 0.415 0.415 0.169 0.205 0.205
## [49] 0.042 0.235 0.341 0.332 0.119 0.119 0.337 0.213 0.214 0.122 0.122 0.174
## [61] 0.121 0.413 0.125 0.171 0.226 0.226 0.250 0.148 0.222 0.034 0.415 0.157
## [73] 0.157 0.243 0.241 0.190 0.119 0.194 0.132 0.120 0.123 0.123 0.171 0.178
## [85] 0.132 0.132 0.123 0.123 0.414 0.153 0.214 0.214 0.230 0.358 0.128 0.143
## [97] 0.012 0.012 0.216
##
## $group
##  [1] 1 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [77] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

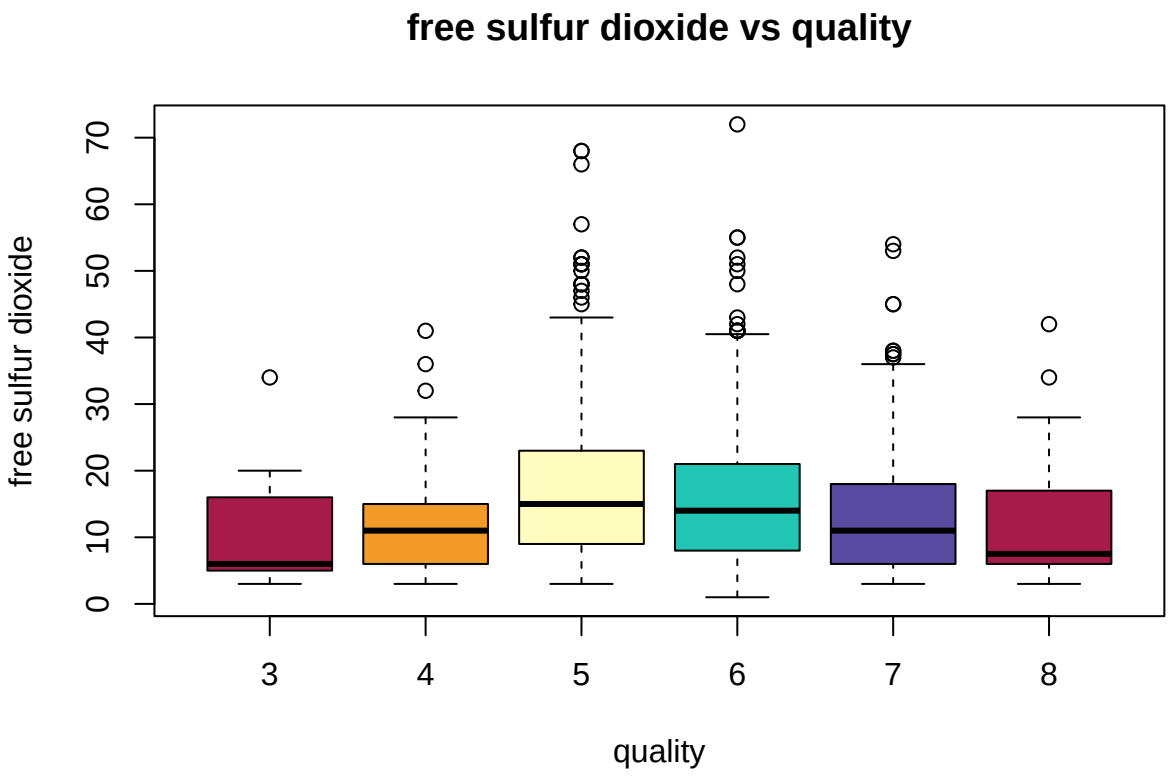
```
##  [1] 0.267 0.172 0.610 0.147 0.123 0.176 0.170 0.368 0.464 0.401 0.467 0.178
## [13] 0.146 0.236 0.360 0.270 0.263 0.611 0.343 0.186 0.159 0.127 0.152 0.152
## [25] 0.124 0.124 0.039 0.157 0.422 0.387 0.132 0.126 0.165 0.161 0.414 0.369
## [37] 0.041 0.166 0.166 0.136 0.403 0.166 0.168 0.415 0.415 0.169 0.205 0.205
## [49] 0.042 0.235 0.341 0.332 0.119 0.119 0.337 0.213 0.214 0.122 0.122 0.174
## [61] 0.121 0.413 0.125 0.171 0.226 0.226 0.250 0.148 0.222 0.034 0.415 0.157
## [73] 0.157 0.243 0.241 0.190 0.119 0.194 0.132 0.120 0.123 0.123 0.171 0.178
## [85] 0.132 0.132 0.123 0.123 0.414 0.153 0.214 0.214 0.230 0.358 0.128 0.143
## [97] 0.012 0.012 0.216
```

- Free sulfur dioxide :

```
summary(data_wine$free_sulfur_dioxide )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    7.00   14.00   15.87   21.00   72.00
```

```
g_caja<-boxplot(data_wine$free_sulfur_dioxide~data_wine$quality ,main="free sulfur dioxide vs quality",
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    3    3    3  1.0    3  3.0
## [2,]    5    6    9  8.0    6  6.0
## [3,]    6   11   15 14.0   11  7.5
## [4,]   16   15   23 21.0   18 17.0
## [5,]   20   28   43 40.5   36 28.0
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##            [,1]      [,2]     [,3]     [,4]      [,5]      [,6]
## [1,]  0.5039614  9.046733 14.15236 13.18681  9.655961  3.403495
## [2,] 11.4960386 12.953267 15.84764 14.81319 12.344039 11.596505
##
## $out
##  [1] 34.0 41.0 32.0 36.0 52.0 51.0 50.0 68.0 68.0 47.0 46.0 45.0 57.0 48.0 51.0
## [16] 51.0 52.0 48.0 48.0 66.0 41.0 41.0 41.0 52.0 51.0 41.0 50.0 48.0 41.0 43.0
## [31] 72.0 55.0 55.0 42.0 38.0 38.0 54.0 53.0 45.0 37.5 37.5 45.0 37.0 34.0 42.0
##
## $group
```

```
##  [1] 1 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5
## [39] 5 5 5 5 5 6 6
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

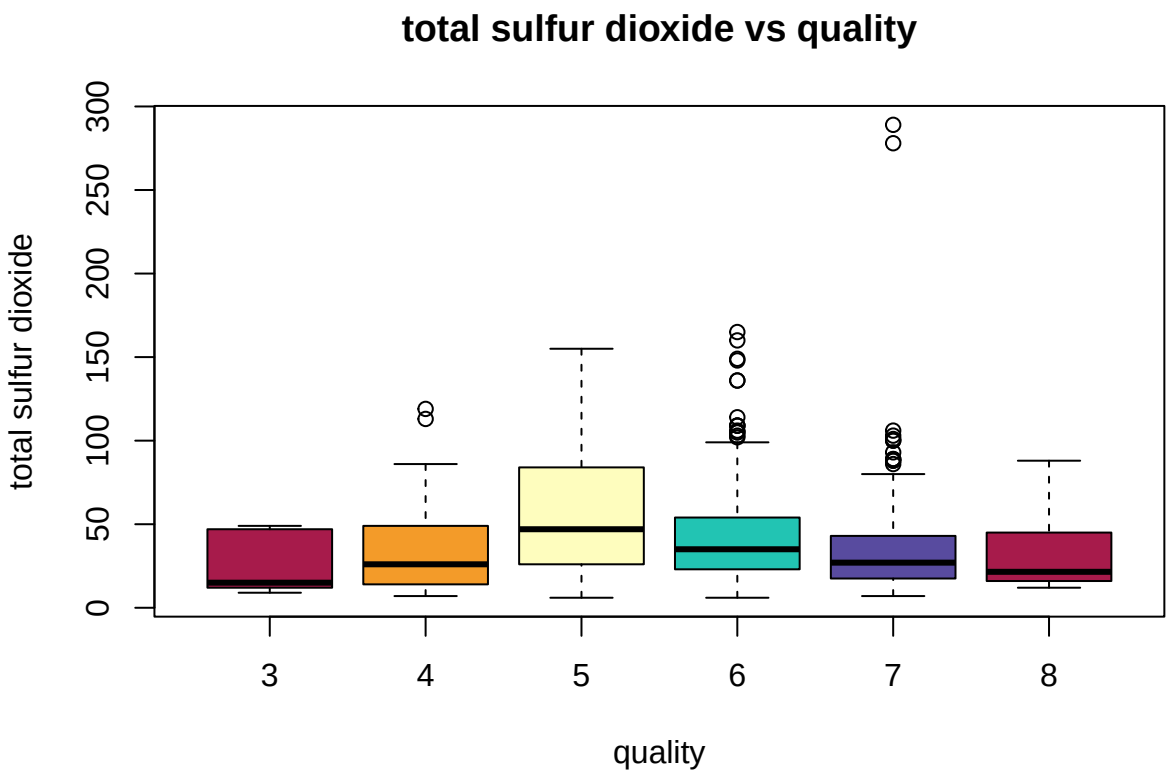
```
##  [1] 34.0 41.0 32.0 36.0 52.0 51.0 50.0 68.0 68.0 47.0 46.0 45.0 57.0 48.0 51.0
## [16] 51.0 52.0 48.0 48.0 66.0 41.0 41.0 41.0 52.0 51.0 41.0 50.0 48.0 41.0 43.0
## [31] 72.0 55.0 55.0 42.0 38.0 38.0 54.0 53.0 45.0 37.5 37.5 45.0 37.0 34.0 42.0
```

- Total sulfur dioxide :

```
summary(data_wine$total_sulfur_dioxide )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   22.00   38.00   46.47   62.00  289.00
```

```
g_caja<-boxplot(data_wine$total_sulfur_dioxide~data_wine$quality ,main="total sulfur dioxide vs quality
```



```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    9    7    6    6  7.0 12.0
## [2,]   12   14   26   23 17.5 16.0
## [3,]   15   26   47   35 27.0 21.5
## [4,]   47   49   84   54 43.0 45.0
## [5,]   49   86  155   99 80.0 88.0
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] -2.487395 18.40396 43.48835 33.06086 24.14392 10.70012
## [2,] 32.487395 33.59604 50.51165 36.93914 29.85608 32.29988
##
## $out
##  [1] 119 113 136 136 106 109 105 114 165 149 103 148 109 109 160 105 102 103  93
## [20] 106  86 100 100 278 289  88 101  88  89  89
##
## $group
##  [1] 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

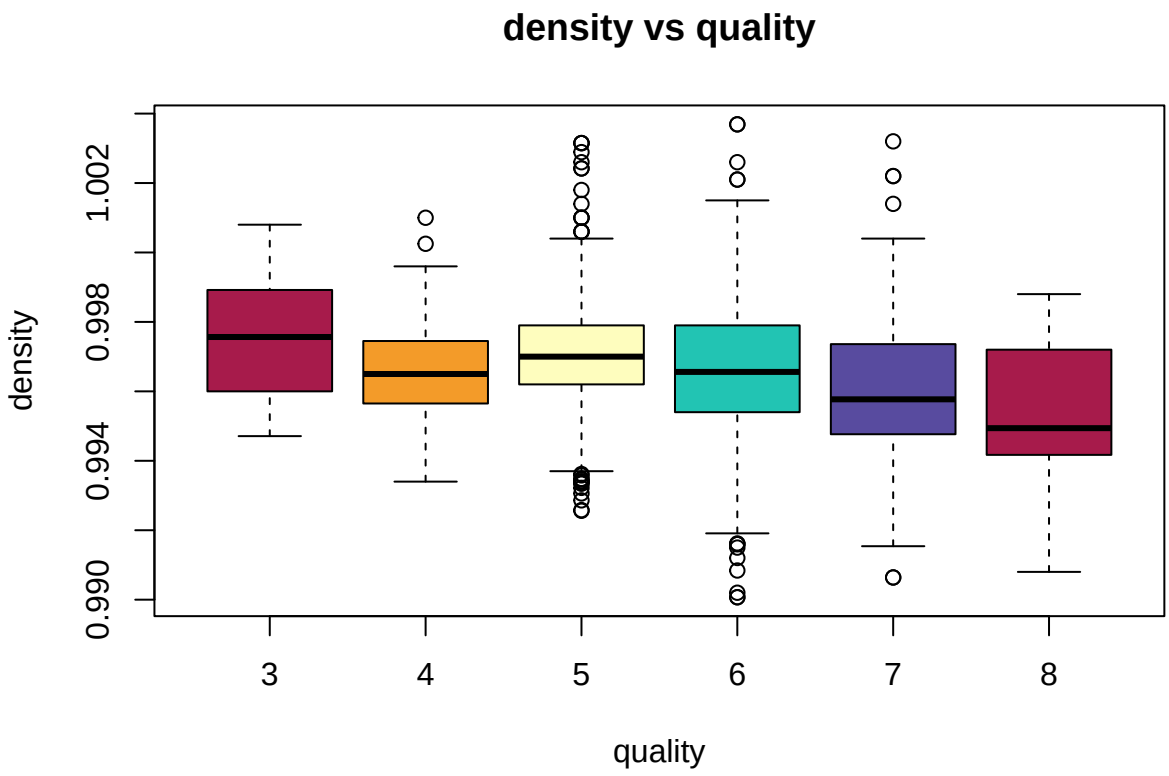
```
##  [1] 119 113 136 136 106 109 105 114 165 149 103 148 109 109 160 105 102 103  93
## [20] 106  86 100 100 278 289  88 101  88  89  89
```

- Density :

```
summary(data_wine$density )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9901  0.9956  0.9968  0.9967  0.9978  1.0037
```

```
g_caja<-boxplot(data_wine$density~data_wine$quality ,main="density vs quality", xlab="quality", ylab="d
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##          [,1]    [,2]   [,3]    [,4]     [,5]    [,6]
## [1,] 0.994710 0.99340 0.9937 0.99191 0.991540 0.99080
## [2,] 0.996000 0.99565 0.9962 0.99540 0.994765 0.99417
## [3,] 0.997565 0.99650 0.9970 0.99656 0.995770 0.99494
## [4,] 0.998920 0.99745 0.9979 0.99790 0.997360 0.99720
## [5,] 1.000800 0.99960 1.0004 1.00150 1.000400 0.99880
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.9961061 0.9961093 0.9968971 0.9964036 0.9954794 0.9938116
## [2,] 0.9990239 0.9968907 0.9971029 0.9967164 0.9960606 0.9960684
##
## $out
##  [1] 1.00025 1.00100 1.00100 1.00180 1.00140 1.00060 1.00260 1.00100 0.99340
## [10] 1.00315 1.00315 1.00315 1.00060 1.00060 1.00100 1.00289 0.99258 0.99256
## [19] 0.99341 0.99346 0.99358 0.99286 0.99322 0.99334 0.99336 1.00242 1.00242
## [28] 0.99348 0.99306 0.99362 0.99160 0.99160 0.99120 1.00210 1.00210 1.00260
## [37] 0.99162 0.99007 0.99007 0.99020 0.99150 0.99084 1.00369 1.00369 1.00220
```

```
## [46] 1.00220 1.00320 1.00140 0.99064 0.99064
##
## $group
##  [1] 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4
## [39] 4 4 4 4 4 4 5 5 5 5 5 5
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

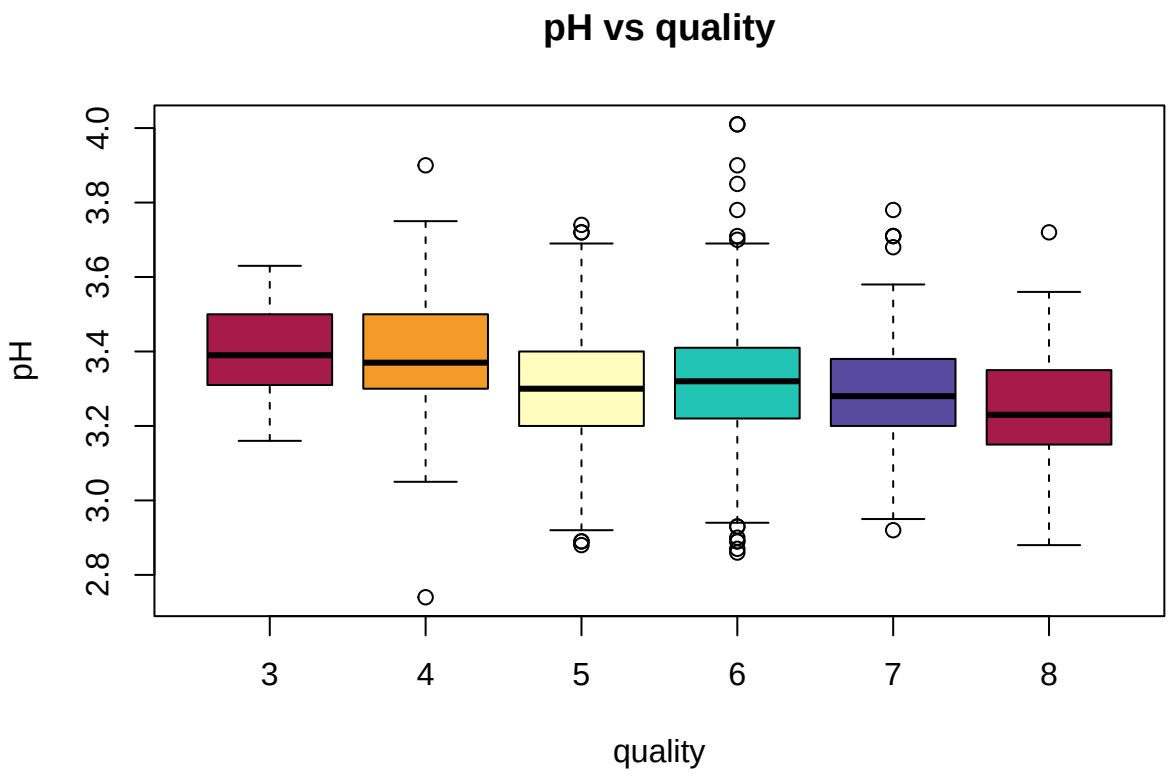
```
##  [1] 1.00025 1.00100 1.00100 1.00180 1.00140 1.00060 1.00260 1.00100 0.99340
## [10] 1.00315 1.00315 1.00315 1.00060 1.00060 1.00100 1.00289 0.99258 0.99256
## [19] 0.99341 0.99346 0.99358 0.99286 0.99322 0.99334 0.99336 1.00242 1.00242
## [28] 0.99348 0.99306 0.99362 0.99160 0.99160 0.99120 1.00210 1.00210 1.00260
## [37] 0.99162 0.99007 0.99007 0.99020 0.99150 0.99084 1.00369 1.00369 1.00220
## [46] 1.00220 1.00320 1.00140 0.99064 0.99064
```

- PH :

```
summary(data_wine$pH )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.740   3.210   3.310   3.311   3.400   4.010
```

```
g_caja<-boxplot(data_wine$pH~data_wine$quality ,main="pH vs quality", xlab="quality", ylab="pH", col=hc
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 3.16 3.05 2.92 2.94 2.95 2.88
## [2,] 3.31 3.30 3.20 3.22 3.20 3.15
## [3,] 3.39 3.37 3.30 3.32 3.28 3.23
## [4,] 3.50 3.50 3.40 3.41 3.38 3.35
## [5,] 3.63 3.75 3.69 3.69 3.58 3.56
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 3.295068 3.326594 3.287891 3.308115 3.259839 3.155518
## [2,] 3.484932 3.413406 3.312109 3.331885 3.300161 3.304482
##
## $out
##  [1] 3.90 2.74 3.74 2.89 2.89 2.88 3.72 3.72 2.93 2.93 3.85 2.86 2.87 3.90 2.89
## [16] 2.89 3.70 3.78 4.01 2.90 4.01 3.71 2.92 3.71 3.71 3.71 3.78 3.68 3.72
##
## $group
##  [1] 2 2 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 6
```

```
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

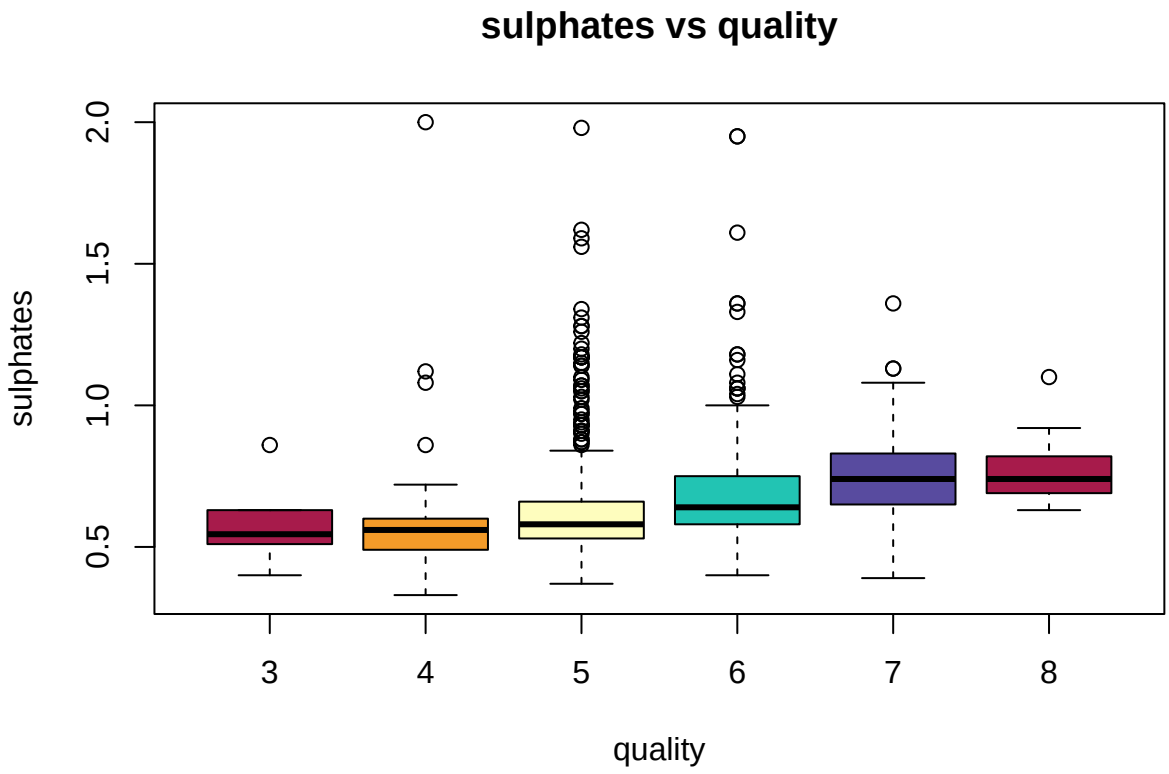
```
##  [1] 3.90 2.74 3.74 2.89 2.89 2.88 3.72 3.72 2.93 2.93 3.85 2.86 2.87 3.90 2.89
## [16] 2.89 3.70 3.78 4.01 2.90 4.01 3.71 2.92 3.71 3.71 3.71 3.78 3.68 3.72
```

- Sulphates :

```
summary(data_wine$sulphates )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3300  0.5500  0.6200  0.6581  0.7300  2.0000
```

```
g_caja<-boxplot(data_wine$sulphates~data_wine$quality ,main="sulphates vs quality", xlab="quality", ylab
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##       [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.400 0.33 0.37 0.40 0.39 0.63
## [2,] 0.510 0.49 0.53 0.58 0.65 0.69
## [3,] 0.545 0.56 0.58 0.64 0.74 0.74
## [4,] 0.630 0.60 0.66 0.75 0.83 0.82
## [5,] 0.630 0.72 0.84 1.00 1.08 0.92
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##           [,1]      [,2]      [,3]     [,4]      [,5]      [,6]
## [1,] 0.4850432 0.5361267 0.5721291 0.629366 0.7198394 0.6915868
## [2,] 0.6049568 0.5838733 0.5878709 0.650634 0.7601606 0.7884132
##
## $out
##  [1] 0.86 1.12 2.00 1.08 0.86 1.56 0.88 0.93 1.28 0.91 0.91 1.20 1.28 1.14 1.22
## [16] 1.98 1.31 0.93 0.93 1.59 1.02 1.03 1.09 1.26 0.87 0.95 0.88 0.87 1.07 0.91
## [31] 1.05 1.06 1.14 0.99 1.05 1.17 1.62 1.18 0.86 1.07 1.34 1.10 0.98 0.94 1.15
## [46] 1.17 1.17 1.17 1.17 0.90 0.93 0.97 0.97 1.08 1.95 1.95 1.61 1.18 1.04 1.11
## [61] 1.06 1.06 1.36 1.36 1.06 1.16 1.33 1.18 1.03 1.36 1.13 1.13 1.10
##
## $group
##  [1] 1 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 6
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

```
#outliers
print(valores_outlier)
```

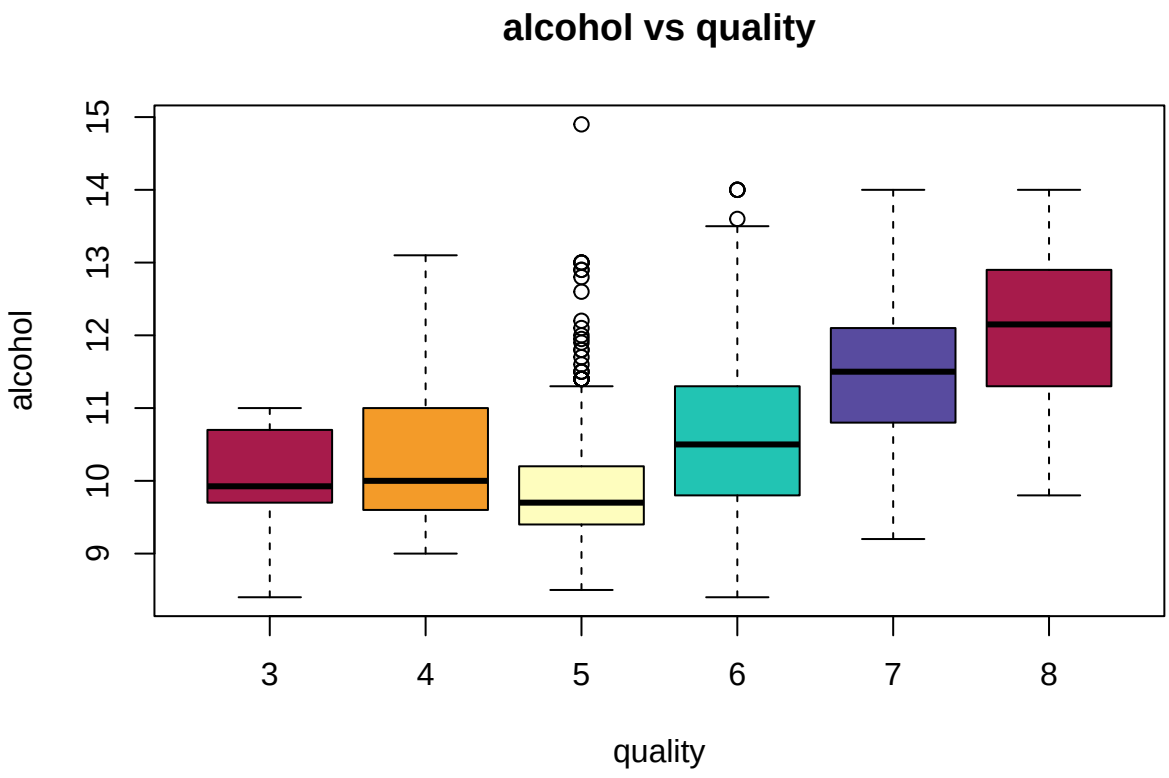
```
##  [1] 0.86 1.12 2.00 1.08 0.86 1.56 0.88 0.93 1.28 0.91 0.91 1.20 1.28 1.14 1.22
## [16] 1.98 1.31 0.93 0.93 1.59 1.02 1.03 1.09 1.26 0.87 0.95 0.88 0.87 1.07 0.91
## [31] 1.05 1.06 1.14 0.99 1.05 1.17 1.62 1.18 0.86 1.07 1.34 1.10 0.98 0.94 1.15
## [46] 1.17 1.17 1.17 1.17 0.90 0.93 0.97 0.97 1.08 1.95 1.95 1.61 1.18 1.04 1.11
## [61] 1.06 1.06 1.36 1.36 1.06 1.16 1.33 1.18 1.03 1.36 1.13 1.13 1.10
```

- Sulphates :

```
summary(data_wine$alcohol )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.40    9.50   10.20   10.42   11.10   14.90
```

```
g_caja<-boxplot(data_wine$alcohol~data_wine$quality ,main="alcohol vs quality", xlab="quality", ylab="al
```




```
valores_outlier<-g_caja$out
g_caja
```

```
## $stats
##        [,1] [,2] [,3] [,4] [,5]  [,6]
## [1,]  8.400  9.0  8.5  8.4  9.2  9.80
## [2,]  9.700  9.6  9.4  9.8 10.8 11.30
## [3,]  9.925 10.0  9.7 10.5 11.5 12.15
## [4,] 10.700 11.0 10.2 11.3 12.1 12.90
## [5,] 11.000 13.1 11.3 13.5 14.0 14.00
##
## $n
## [1]  10  53 681 638 199  18
##
## $conf
##          [,1]      [,2]     [,3]     [,4]    [,5]     [,6]
## [1,]  9.42536  9.696158 9.651563 10.40617 11.3544 11.55414
## [2,] 10.42464 10.303842 9.748437 10.59383 11.6456 12.74586
##
## $out
##  [1] 13.00 13.00 11.50 13.00 11.80 11.50 11.40 11.40 14.90 11.50 11.80 11.40
## [13] 11.60 12.90 12.80 12.90 11.40 11.40 12.20 13.00 11.40 11.40 11.70 12.60
## [25] 11.40 11.40 11.95 12.00 12.10 11.90 14.00 14.00 14.00 13.60 14.00
##
## $group
```

```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4
##
## $names
## [1] "3" "4" "5" "6" "7" "8"
```

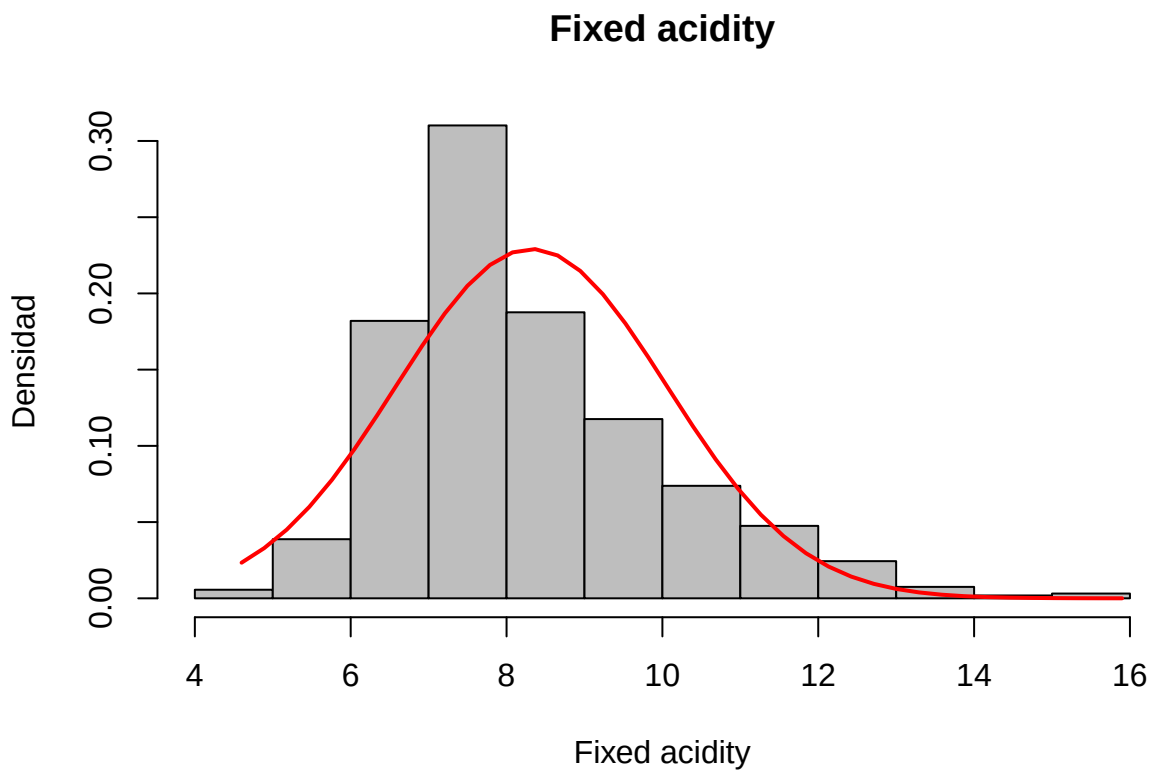
```
#outliers
print(valores_outlier)
```

```
##  [1] 13.00 13.00 11.50 13.00 11.80 11.50 11.40 11.40 14.90 11.50 11.80 11.40
## [13] 11.60 12.90 12.80 12.90 11.40 11.40 12.20 13.00 11.40 11.40 11.70 12.60
## [25] 11.40 11.40 11.95 12.00 12.10 11.90 14.00 14.00 14.00 13.60 14.00
```

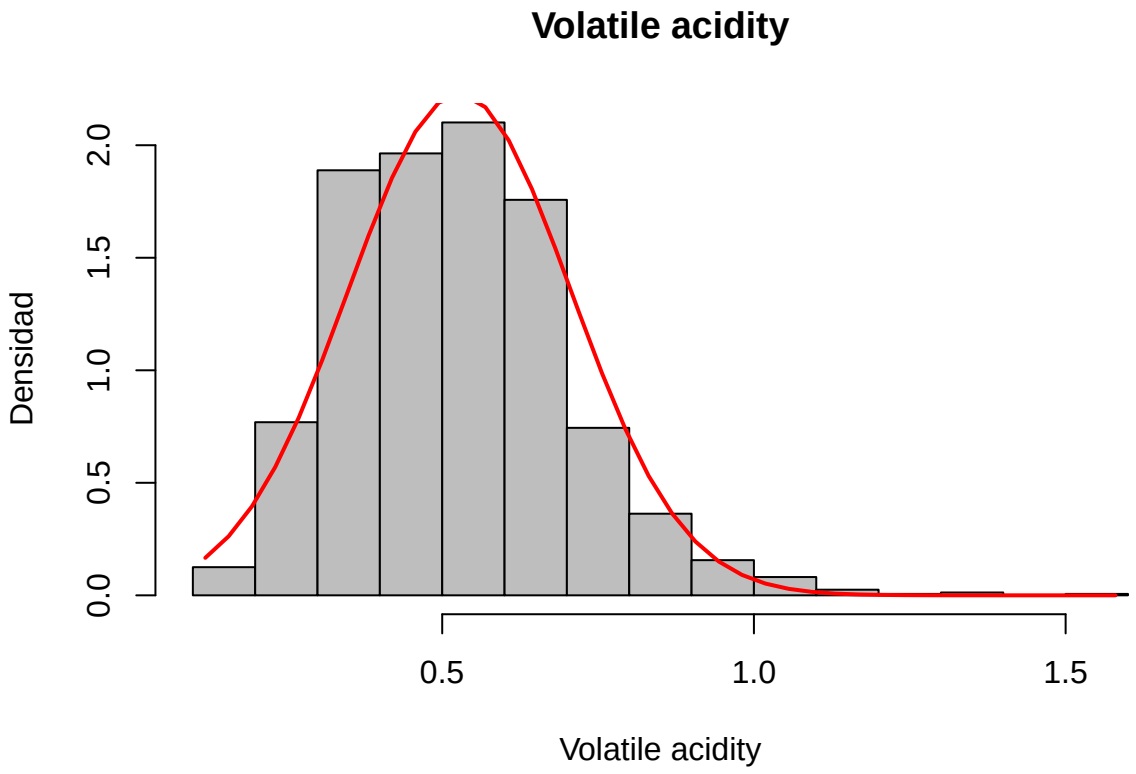
Análisis de los datos

Analizamos la distribución de las variables:

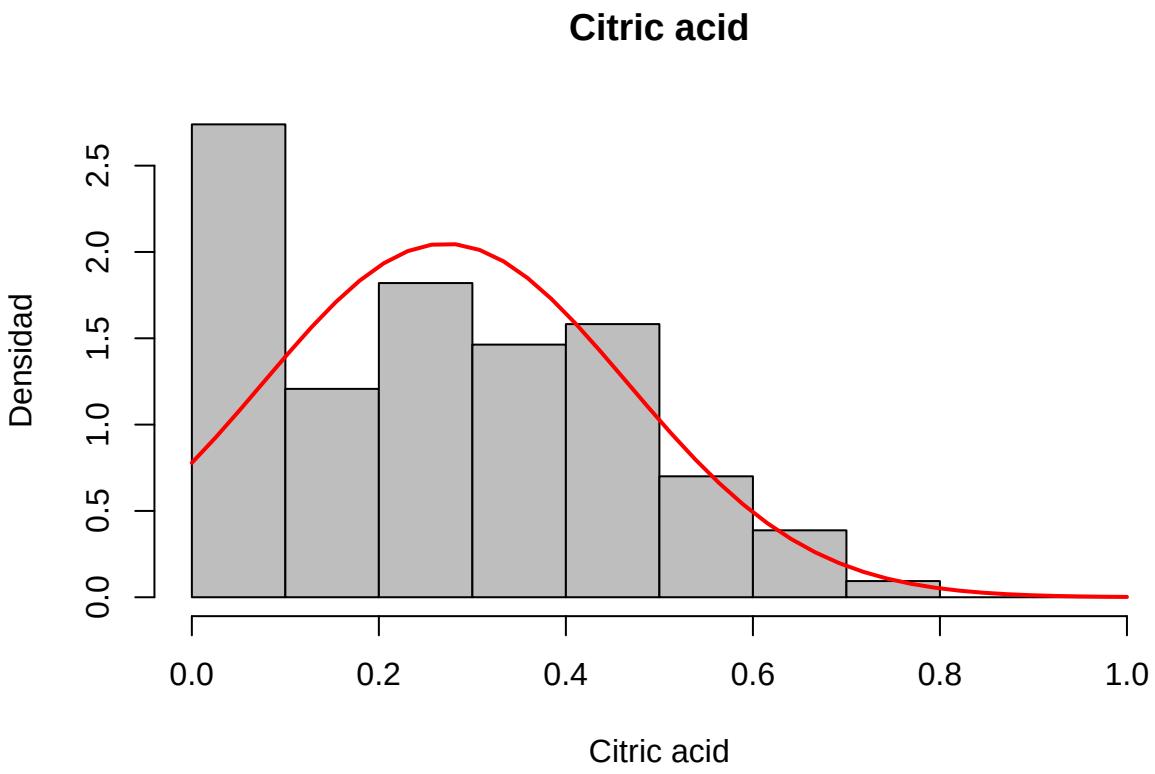
```
hist(data_wine$fixed_acidity, prob = TRUE,
     main = "Fixed acidity", ylab = "Densidad", col="grey",xlab="Fixed acidity")
x <- seq(min(data_wine$fixed_acidity), max(data_wine$fixed_acidity), length = 40)
f <- dnorm(x, mean = mean(data_wine$fixed_acidity), sd = sd(data_wine$fixed_acidity))
lines(x, f, col = "red", lwd = 2)
```



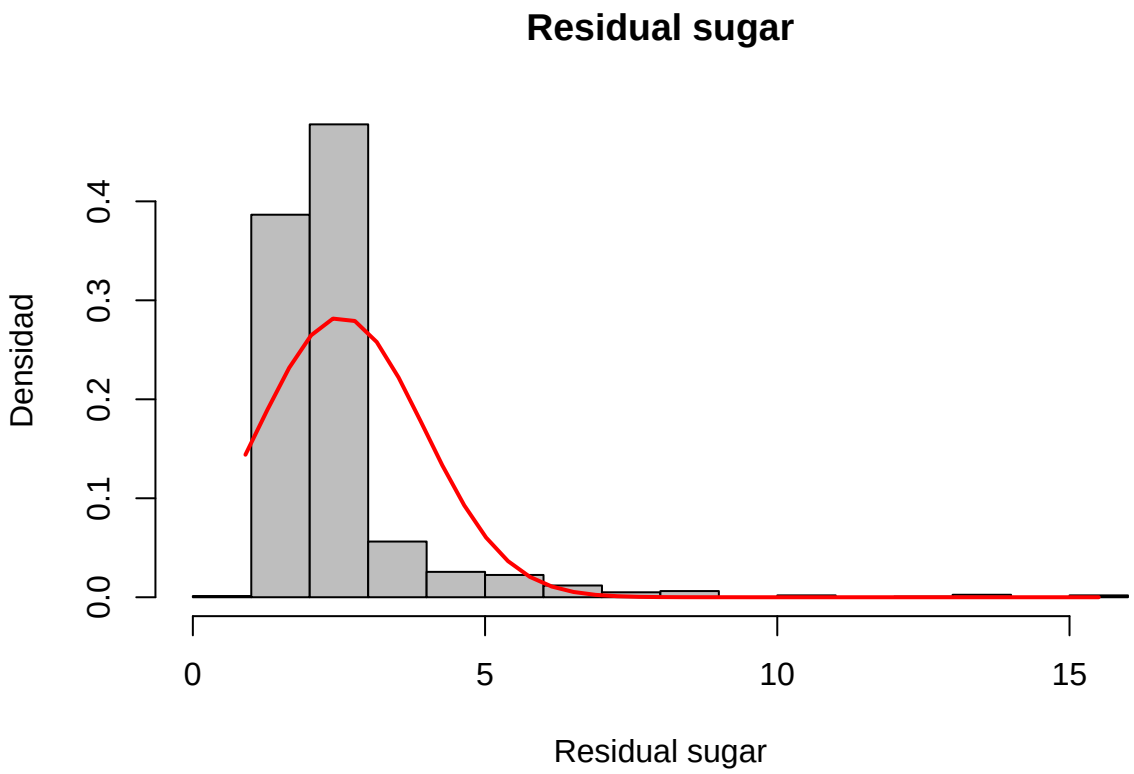
```r
hist(data_wine$volatile_acidity , prob = TRUE,
     main = "Volatile acidity", ylab = "Densidad", col='grey',xlab="Volatile acidity")
x <- seq(min(data_wine$volatile_acidity), max(data_wine$volatile_acidity), length = 40)
f <- dnorm(x, mean = mean(data_wine$volatile_acidity), sd = sd(data_wine$volatile_acidity ))
lines(x, f, col = "red", lwd = 2)
```



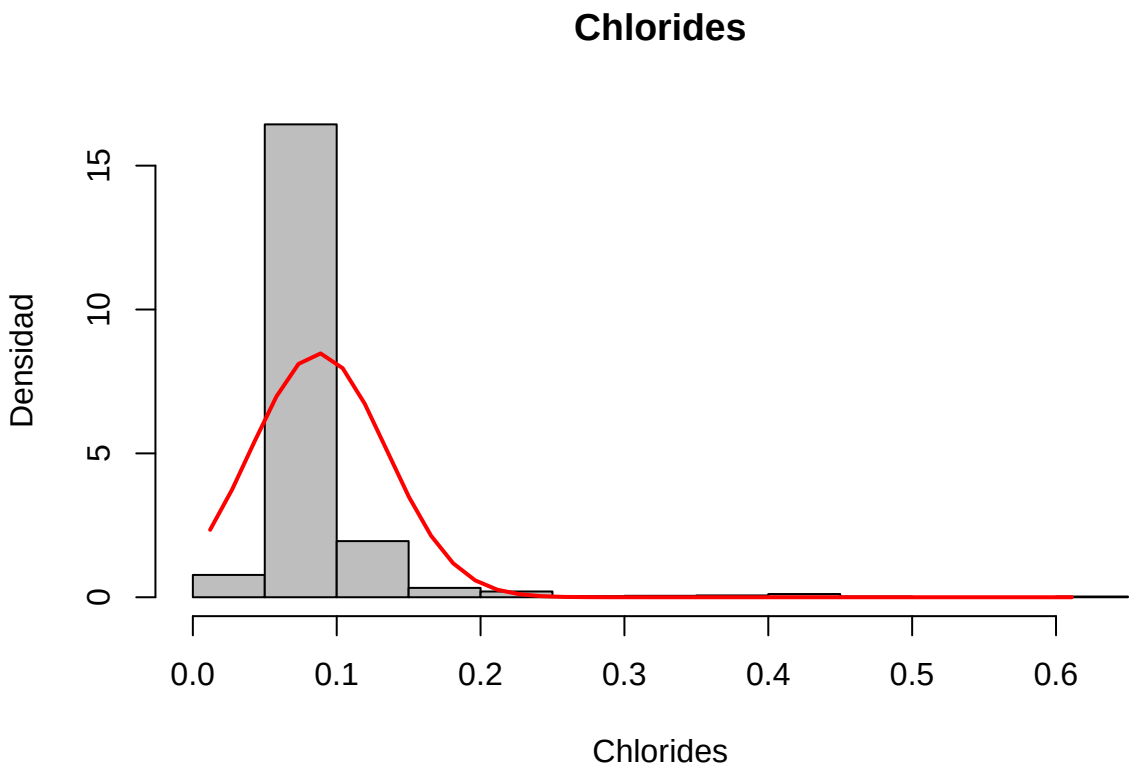

```r
hist(data_wine$citric_acid , prob = TRUE,
     main = "Citric acid", ylab = "Densidad", col='grey',xlab="Citric acid")
x <- seq(min(data_wine$citric_acid), max(data_wine$citric_acid), length = 40)
f <- dnorm(x, mean = mean(data_wine$citric_acid), sd = sd(data_wine$citric_acid))
lines(x, f, col = "red", lwd = 2)
```



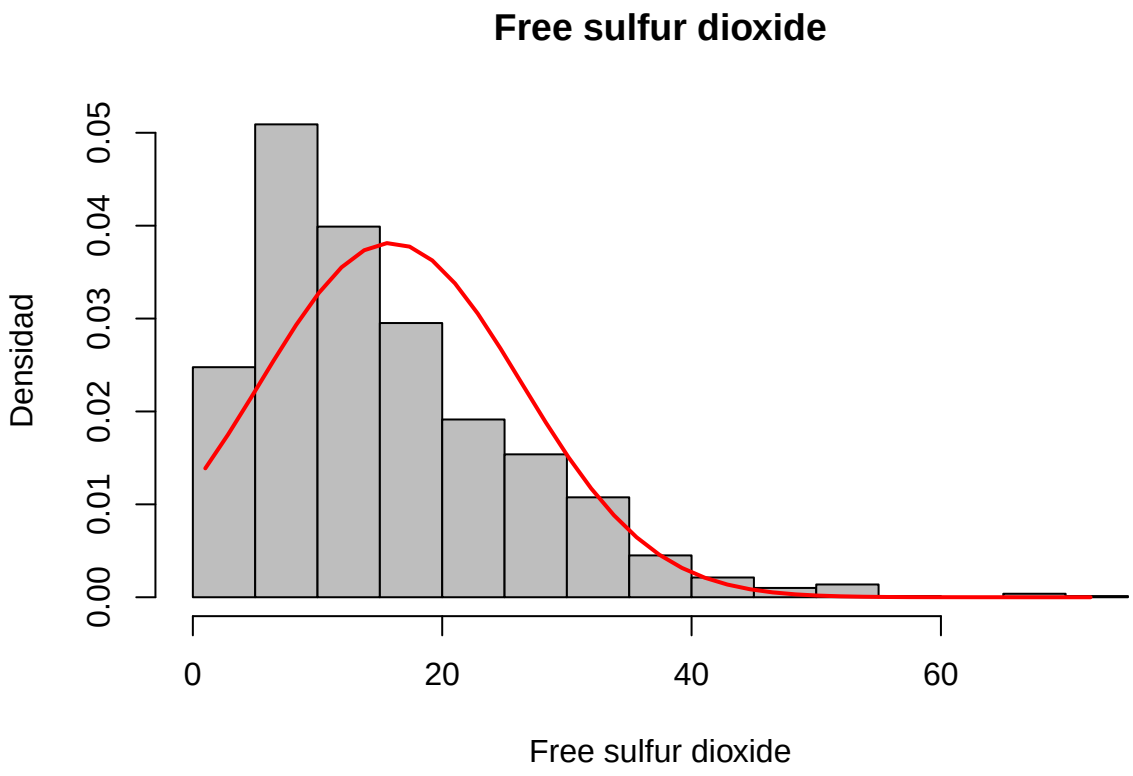

```
hist(data_wine$residual_sugar , prob = TRUE,
     main = "Residual sugar", ylab = "Densidad", col='grey',xlab="Residual sugar")
x <- seq(min(data_wine$residual_sugar), max(data_wine$residual_sugar), length = 40)
f <- dnorm(x, mean = mean(data_wine$residual_sugar), sd = sd(data_wine$residual_sugar))
lines(x, f, col = "red", lwd = 2)
```



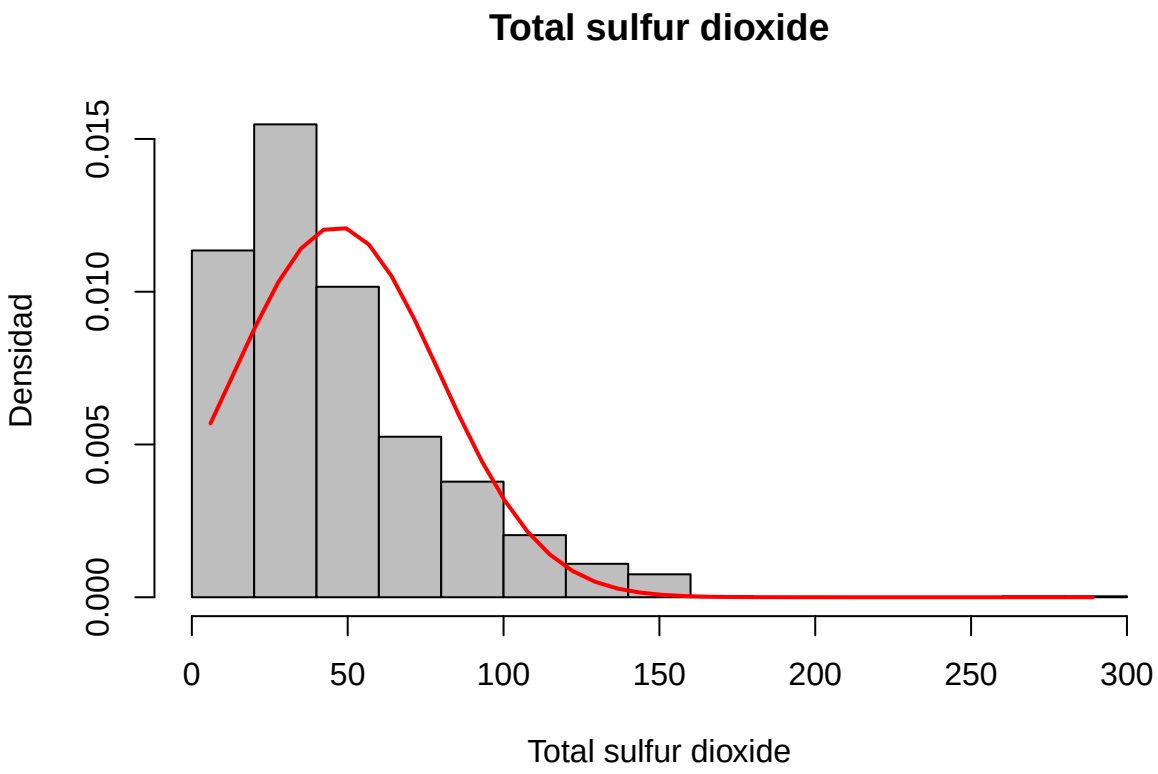

```
hist(data_wine$chlorides , prob = TRUE,
     main = "Chlorides", ylab = "Densidad", col='grey',xlab="Chlorides")
x <- seq(min(data_wine$chlorides), max(data_wine$chlorides), length = 40)
f <- dnorm(x, mean = mean(data_wine$chlorides), sd = sd(data_wine$chlorides))
lines(x, f, col = "red", lwd = 2)
```



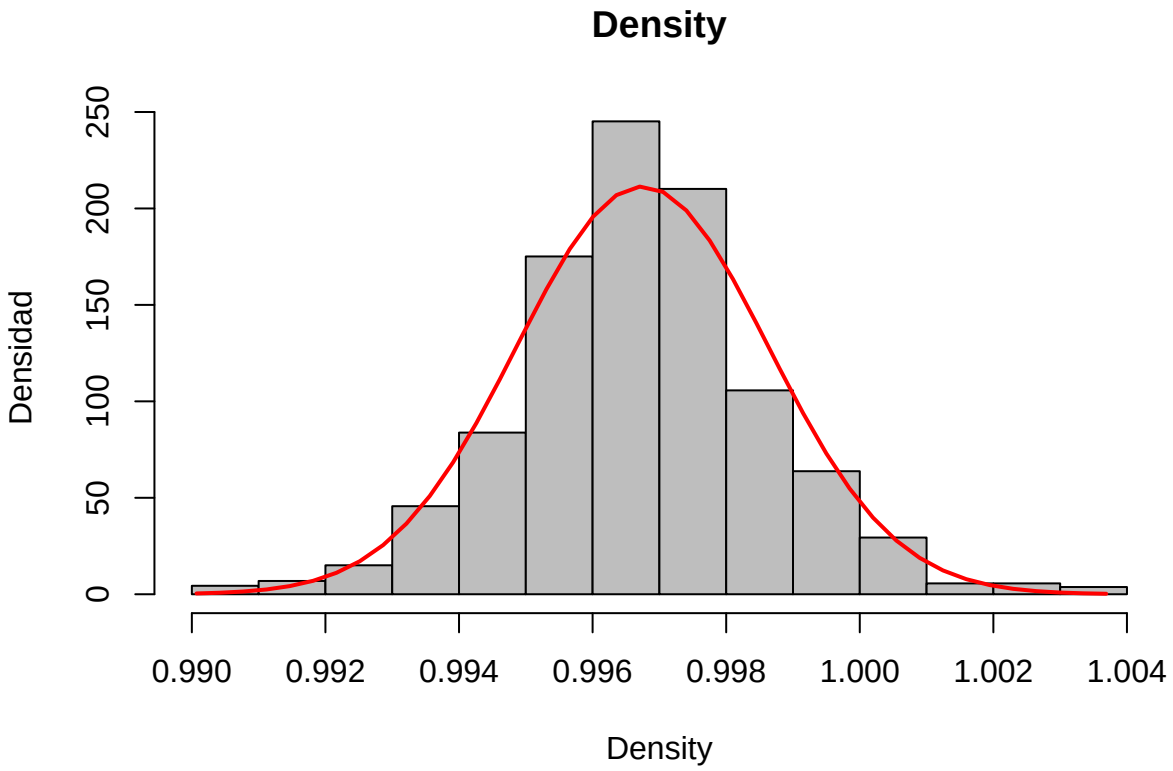

```
hist(data_wine$free_sulfur_dioxide , prob = TRUE,
     main = "Free sulfur dioxide", ylab = "Densidad", col='grey',xlab="Free sulfur dioxide")
x <- seq(min(data_wine$free_sulfur_dioxide), max(data_wine$free_sulfur_dioxide), length = 40)
f <- dnorm(x, mean = mean(data_wine$free_sulfur_dioxide), sd = sd(data_wine$free_sulfur_dioxide))
lines(x, f, col = "red", lwd = 2)
```



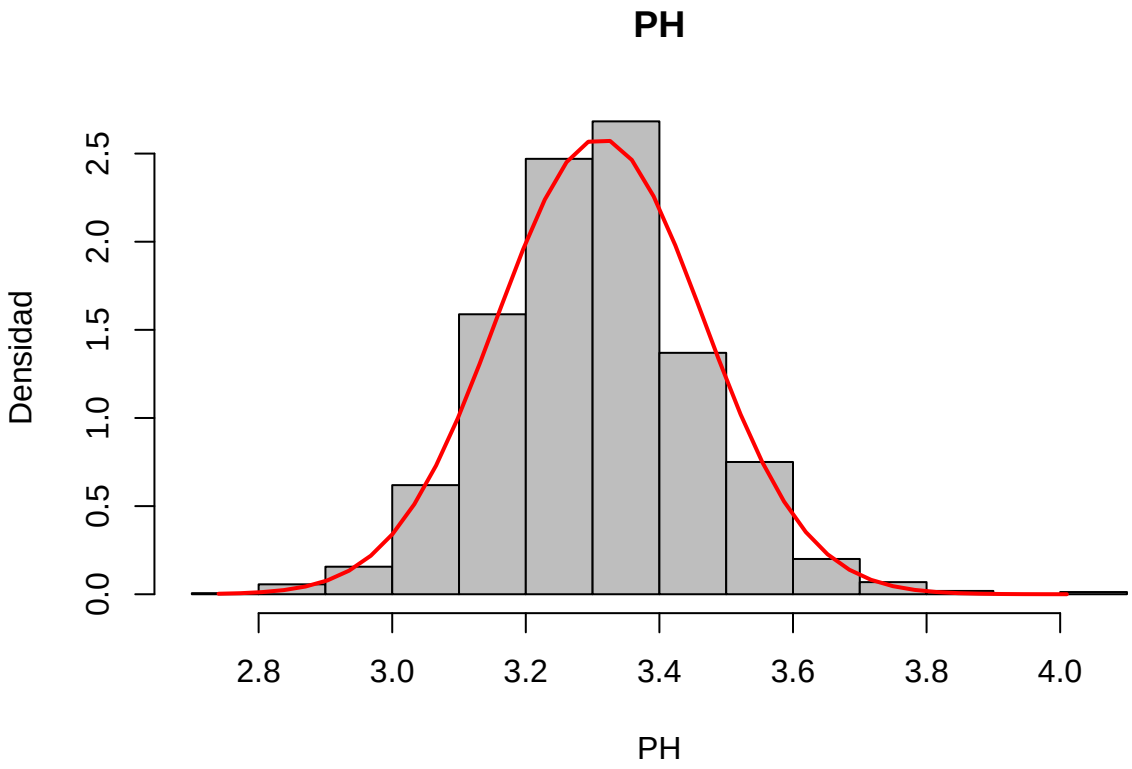

```
hist(data_wine$total_sulfur_dioxide , prob = TRUE,
     main = "Total sulfur dioxide", ylab = "Densidad", col='grey',xlab="Total sulfur dioxide")
x <- seq(min(data_wine$total_sulfur_dioxide), max(data_wine$total_sulfur_dioxide), length = 40)
f <- dnorm(x, mean = mean(data_wine$total_sulfur_dioxide), sd = sd(data_wine$total_sulfur_dioxide))
lines(x, f, col = "red", lwd = 2)
```



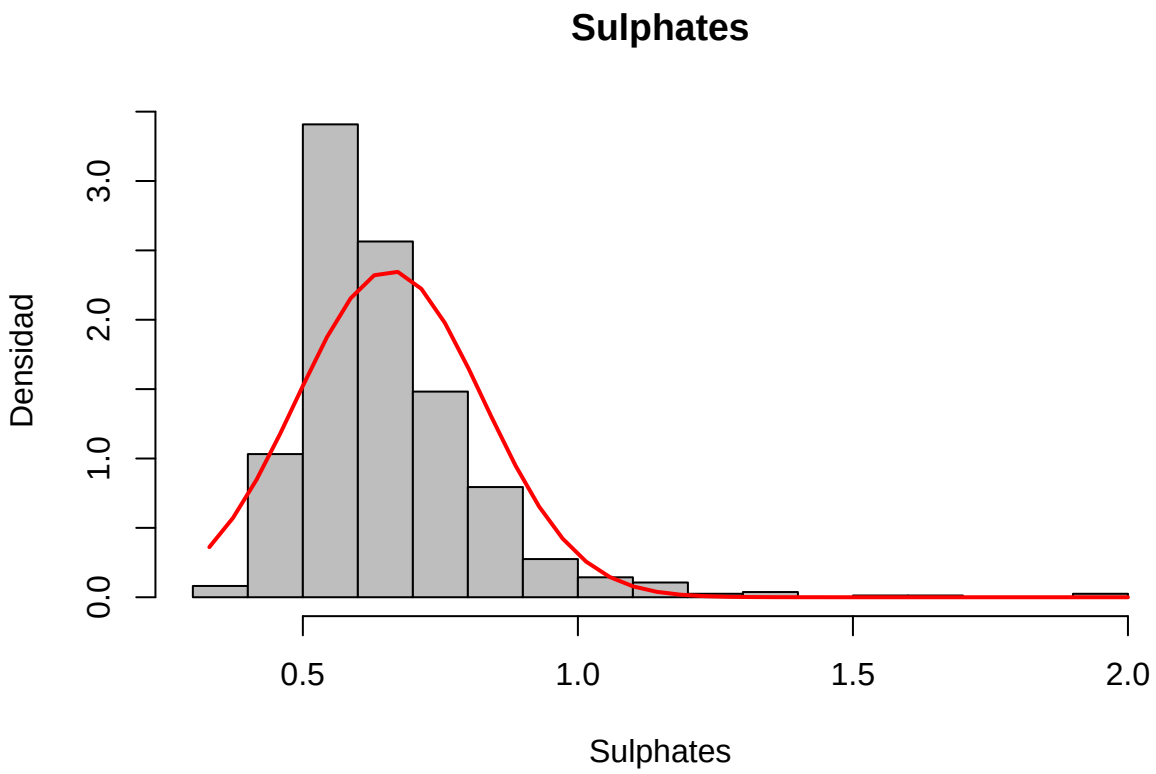

```
hist(data_wine$density , prob = TRUE,
     main = "Density", ylab = "Densidad", col='grey',xlab="Density")
x <- seq(min(data_wine$density), max(data_wine$density), length = 40)
f <- dnorm(x, mean = mean(data_wine$density), sd = sd(data_wine$density))
lines(x, f, col = "red", lwd = 2)
```



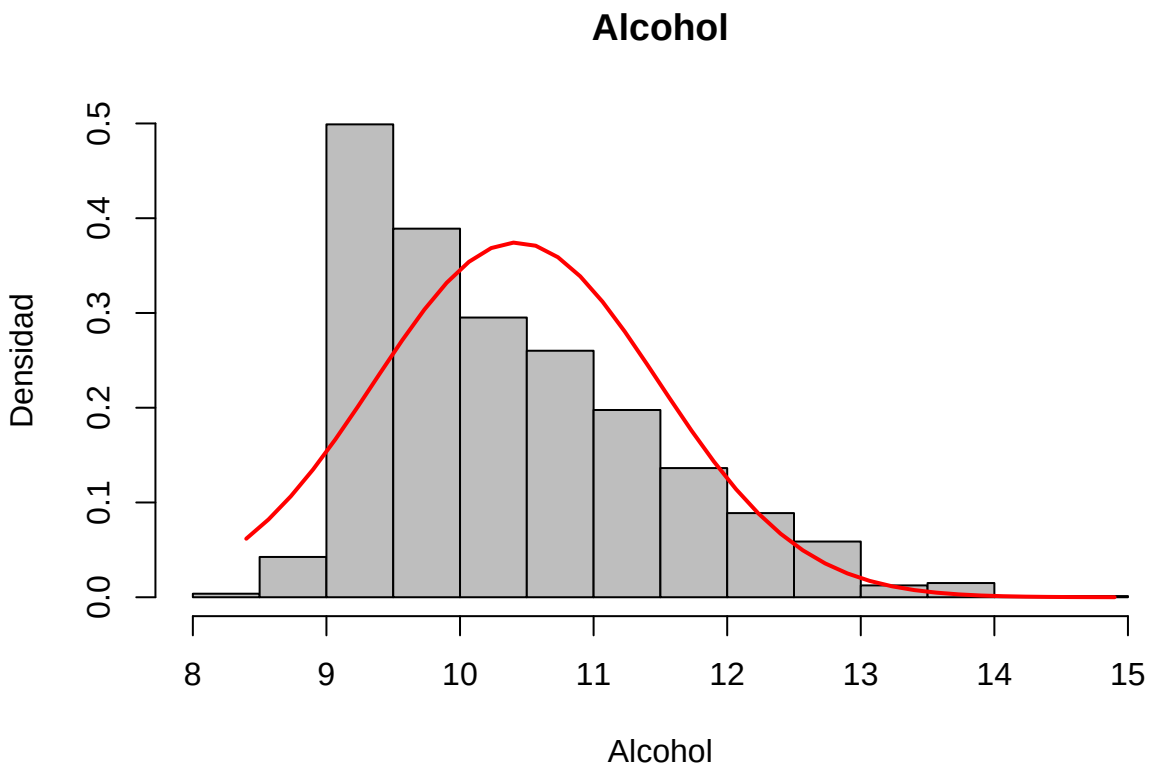

```
hist(data_wine$pH , prob = TRUE,
     main = "PH", ylab = "Densidad", col='grey',xlab="PH")
x <- seq(min(data_wine$pH), max(data_wine$pH), length = 40)
f <- dnorm(x, mean = mean(data_wine$pH), sd = sd(data_wine$pH))
lines(x, f, col = "red", lwd = 2)
```



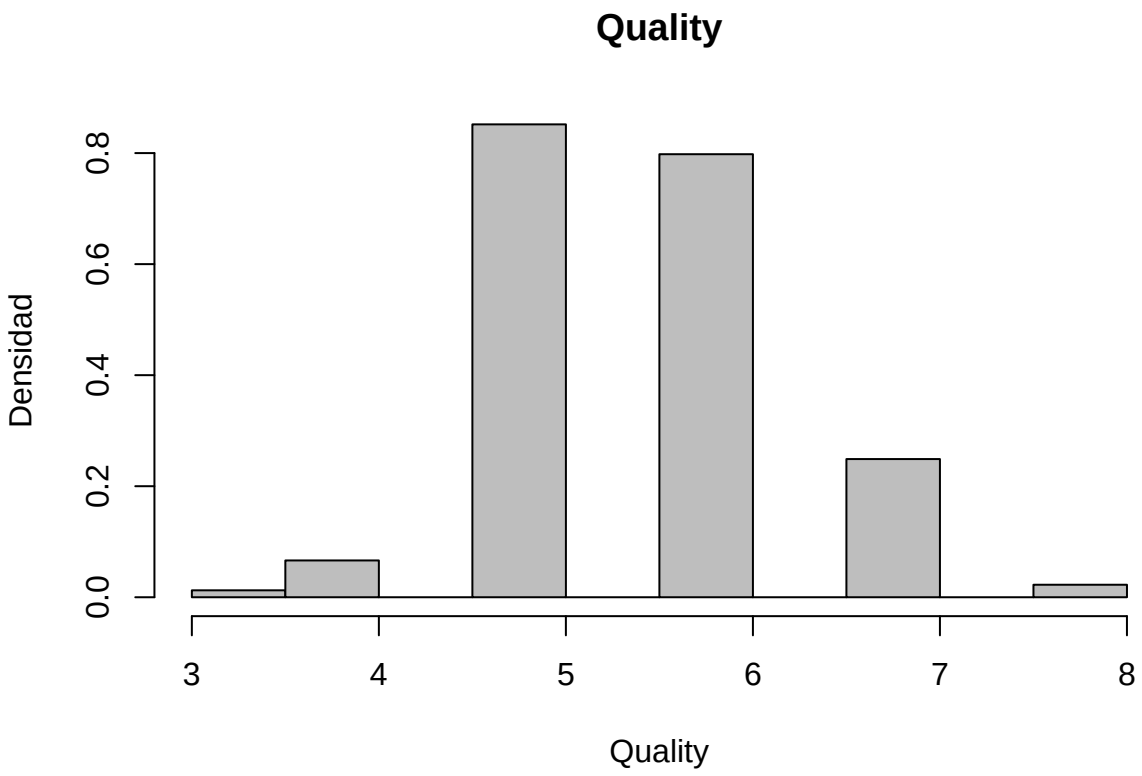

```
hist(data_wine$sulphates , prob = TRUE,
     main = "Sulphates", ylab = "Densidad", col='grey',xlab="Sulphates")
x <- seq(min(data_wine$sulphates), max(data_wine$sulphates), length = 40)
f <- dnorm(x, mean = mean(data_wine$sulphates), sd = sd(data_wine$sulphates))
lines(x, f, col = "red", lwd = 2)
```




```
hist(data_wine$alcohol , prob = TRUE,
     main = "Alcohol", ylab = "Densidad", col='grey',xlab="Alcohol")
x <- seq(min(data_wine$alcohol), max(data_wine$alcohol), length = 40)
f <- dnorm(x, mean = mean(data_wine$alcohol), sd = sd(data_wine$alcohol))
lines(x, f, col = "red", lwd = 2)
```




```
hist(data_wine$quality , prob = TRUE,
     main = "Quality", ylab = "Densidad", col='grey',xlab="Quality")
```




```
prop.table(table(data_wine$quality))
```

```
##
##           3           4           5           6           7           8
## 0.006253909 0.033145716 0.425891182 0.398999375 0.124452783 0.011257036
```

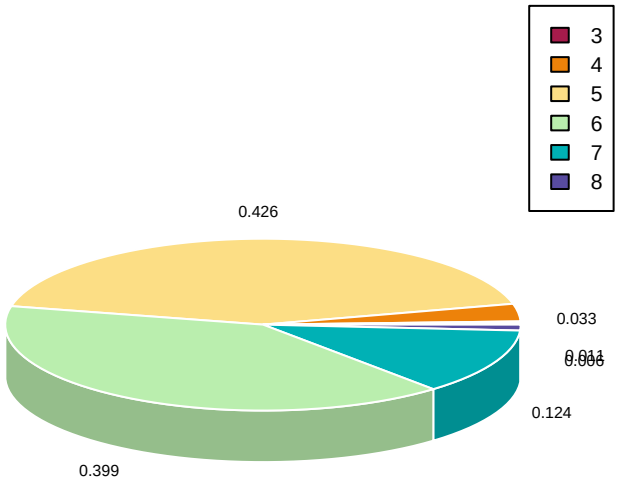
```
mytable <- prop.table(table(data_wine$quality))
mytable <- round(mytable,digits=3)

kable(mytable*100, digits=5) %>%
  kable_styling(full_width = T) %>%
  column_spec(col = 1, background="steelblue", bold=T, color="white") %>%
  row_spec(row = 0,color="blue")
```

Var1	Freq
	0.6
	3.3
	42.6
	39.9
	12.4
	1.1

```
pie3D(mytable,
      col = hcl.colors(length(mytable), "Spectral"),
```

```
      border = "white", labels=mytable,labelcex = 0.50)
par(xpd = TRUE)
legend(1, 0.7, legend = names(mytable), cex=0.7, yjust=0.2, xjust = -0.1,
       fill = hcl.colors(length(mytable), "Spectral"))
```



Comprobamos la normalidad y homegeneidad de la varianza

```
# Test de normalidad
apply(data_wine,2,shapiro.test)
```

```
## $fixed_acidity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.94203, p-value < 2.2e-16
##
##
## $volatile_acidity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.97434, p-value = 2.693e-16
##
```

```
##
## $citric_acid
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.95529, p-value < 2.2e-16
##
##
## $residual_sugar
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.56608, p-value < 2.2e-16
##
##
## $chlorides
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.48425, p-value < 2.2e-16
##
##
## $free_sulfur_dioxide
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.90184, p-value < 2.2e-16
##
##
## $total_sulfur_dioxide
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.87322, p-value < 2.2e-16
##
##
## $density
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.99087, p-value = 1.936e-08
##
##
## $pH
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
```

```
## W = 0.99349, p-value = 1.712e-06
##
##
## $sulphates
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.83304, p-value < 2.2e-16
##
##
## $alcohol
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.92884, p-value < 2.2e-16
##
##
## $quality
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.85759, p-value < 2.2e-16
```

```
# Test de homocedasticidad
data_wine$quality_factor <- factor(data_wine$quality)
leveneTest(data = data_wine, fixed_acidity ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  5.9047 2.042e-05 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data = data_wine, volatile_acidity ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  6.5965 4.364e-06 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data = data_wine, citric_acid ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value  Pr(>F)
## group    5  2.4189 0.03397 *
##       1593
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data = data_wine, residual_sugar ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  1.0049 0.4133
##       1593
```

```
leveneTest(data = data_wine, chlorides ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  1.7306 0.1244
##       1593
```

```
leveneTest(data = data_wine, free_sulfur_dioxide ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  1.6665 0.1395
##       1593
```

```
leveneTest(data = data_wine, total_sulfur_dioxide ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5   20.68 < 2.2e-16 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data = data_wine, density ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  9.7725 3.274e-09 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data = data_wine, pH ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  0.2978 0.9143
##       1593
```

```
leveneTest(data = data_wine, sulphates ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  0.2301 0.9495
##       1593
```

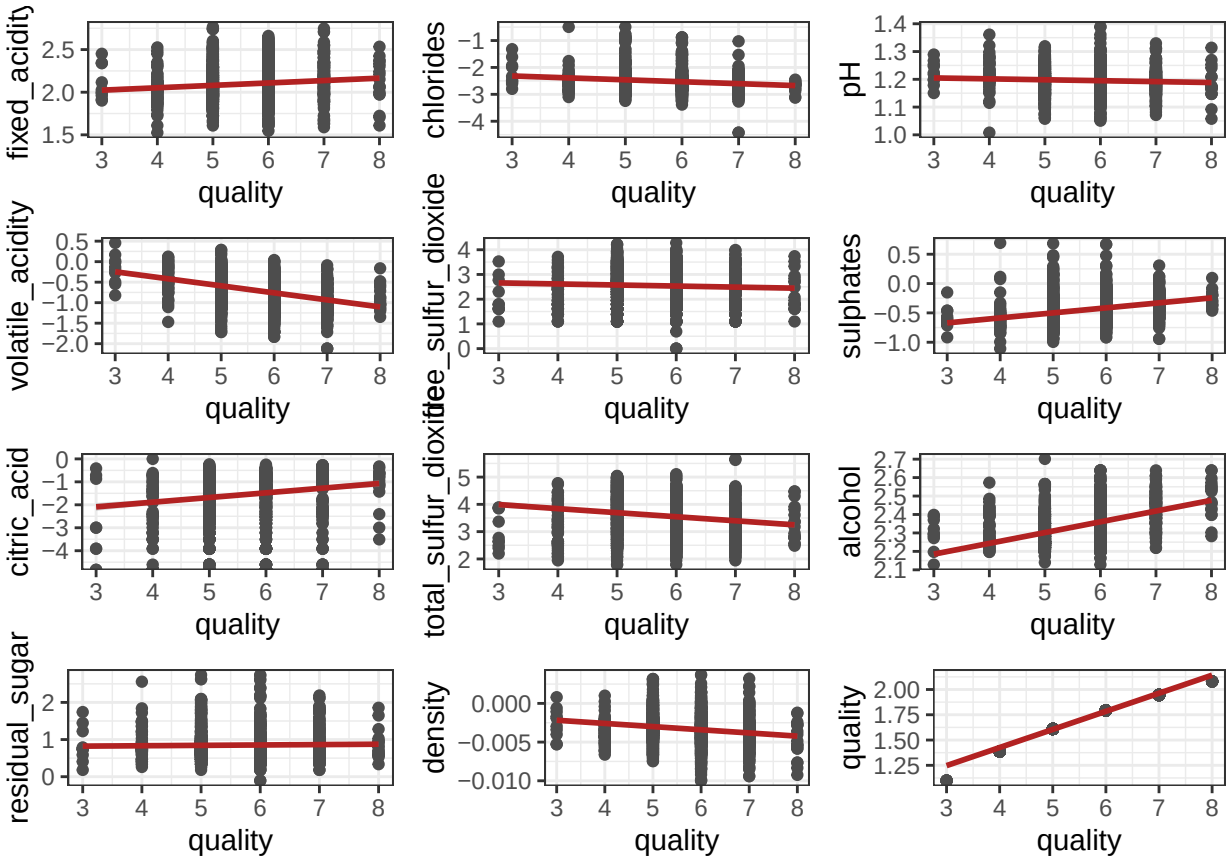
```
leveneTest(data = data_wine, alcohol ~ quality_factor)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  24.226 < 2.2e-16 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
data_wine <- subset(data_wine, select= -c(quality_factor))
```

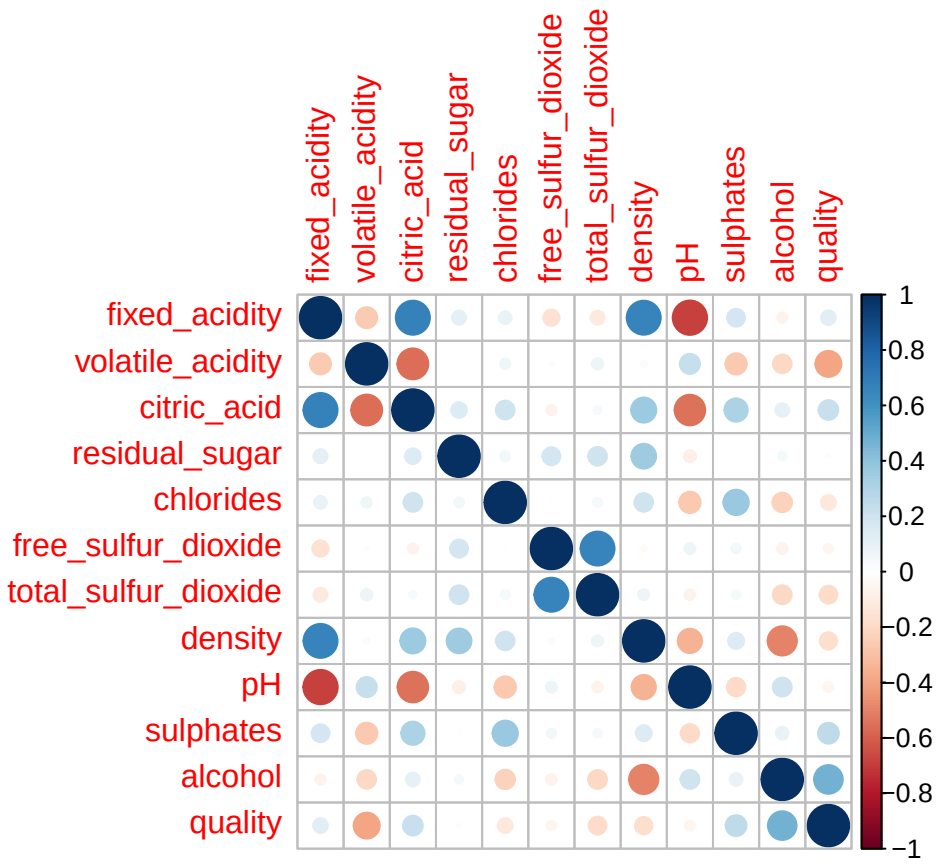
Vamos a intentar buscar una correlación entre la calidad del vino (variable quality) y las distintas variables del juego de datos:

```
histList2<- vector('list', ncol(data_wine))
for(i in seq_along(data_wine)){
  message(i)
histList2[[i]]<-local({
  i<-i
  col <-log(data_wine[[i]])
  ggp<- ggplot(data = data_wine, aes(x = data_wine$quality, y=col)) +
    geom_point(color = "gray30") + geom_smooth(method = lm,color = "firebrick") +
    theme_bw() + xlab("quality") + ylab(names(data_wine)[i])
  })
}
multiplot(plotlist = histList2, cols =3 )
```



Generamos la matriz de correlaciones:

```
# Gráfico de correlaciones
corrplot(cor(data_wine))
```

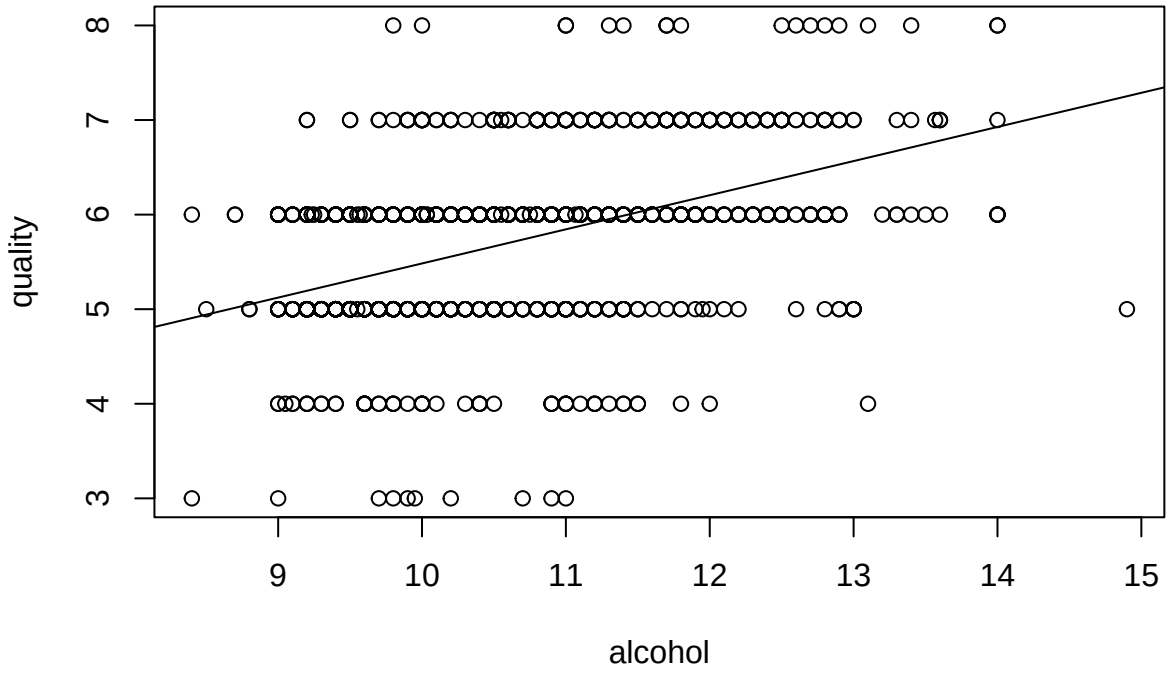


Construimos modelos de minería con los que aplicar métodos de análisis

```
# Modelo de regresión lineal simple con la variable con mayor correlacion
model_1 <- lm(quality ~ alcohol, data = data_wine)
summary(model_1)
```

```
##
## Call:
## lm(formula = quality ~ alcohol, data = data_wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8442 -0.4112 -0.1690  0.5166  2.5888
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.87497    0.17471   10.73   <2e-16 ***
## alcohol      0.36084    0.01668   21.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7104 on 1597 degrees of freedom
## Multiple R-squared:  0.2267, Adjusted R-squared:  0.2263
## F-statistic: 468.3 on 1 and 1597 DF,  p-value: < 2.2e-16
```

```r
plot(data_wine$alcohol, data_wine$quality, xlab="alcohol", ylab="quality")
abline(model_1)
```

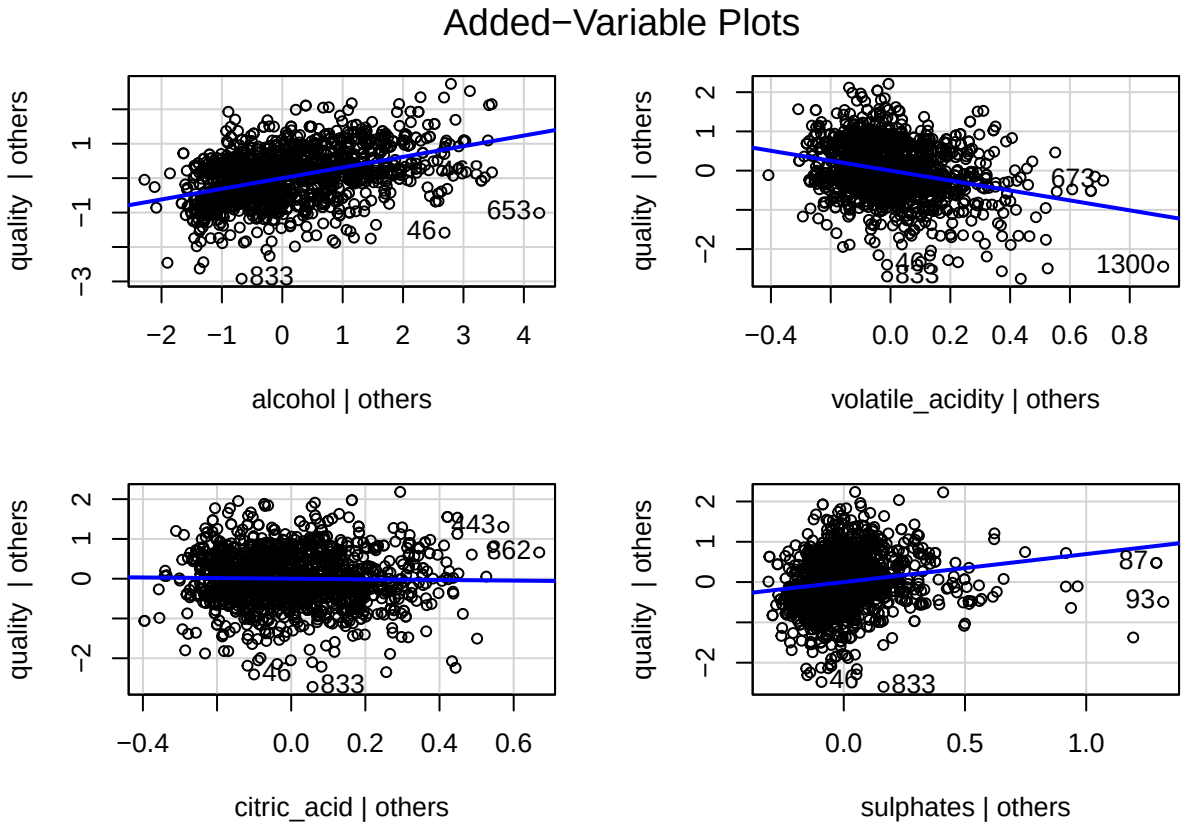



```r
# Modelo de regresión lineal multiple con las variables con mayores correlaciones
model_2 <- lm(quality ~ alcohol + volatile_acidity + citric_acid + sulphates, data = data_wine)
summary(model_2)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile_acidity + citric_acid +
##     sulphates, data = data_wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.71408 -0.38590 -0.06402  0.46657  2.20393
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.64592    0.20106  13.160  < 2e-16 ***
## alcohol            0.30908    0.01581  19.553  < 2e-16 ***
## volatile_acidity -1.26506    0.11266 -11.229  < 2e-16 ***
## citric_acid      -0.07913    0.10381  -0.762    0.446
## sulphates          0.69552    0.10311   6.746 2.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.6588 on 1594 degrees of freedom
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.3345
## F-statistic: 201.8 on 4 and 1594 DF,  p-value: < 2.2e-16
```

```
avPlots(model_2)
```



Construimos un modelo de clasificación de tipo random forest

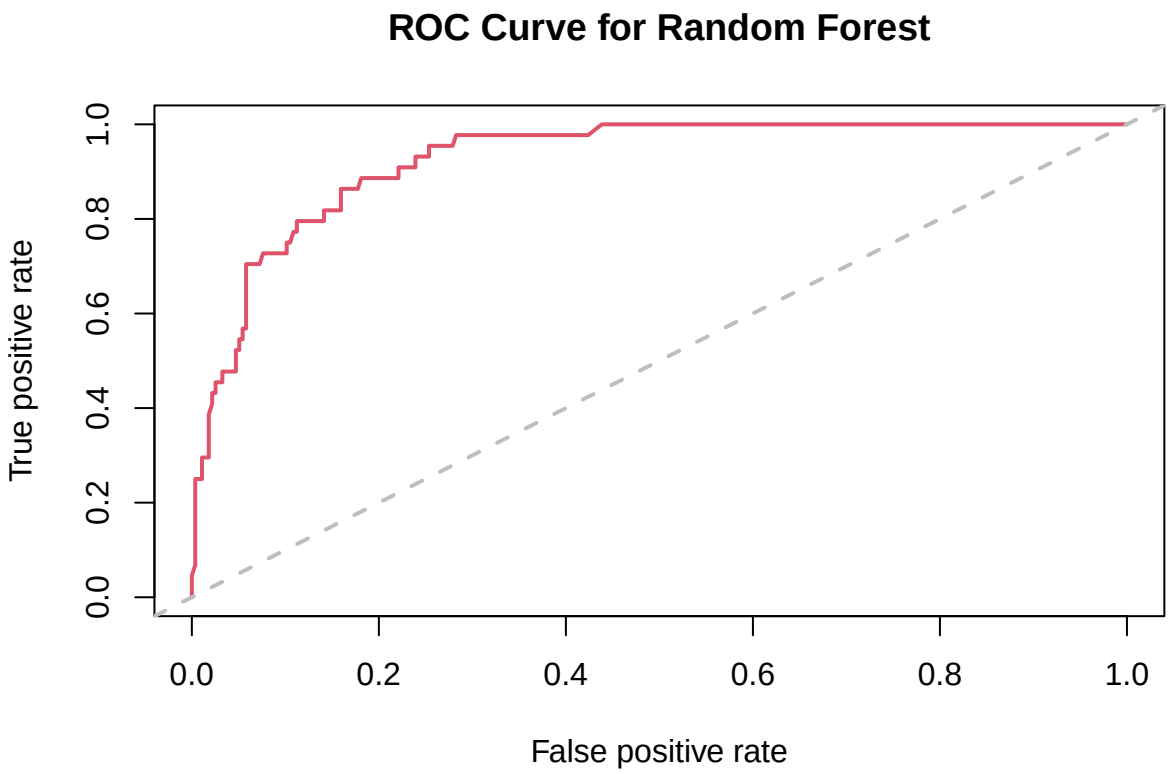
```
# Pasamos la variable respuesta a binario, 0 si mala calidad y 1 si buena calidad
# Esto lo hacemos para construir un modelo de clasificacion
data_wine$quality <- as.factor(ifelse(data_wine$quality < 7, 0,
                               ifelse(data_wine$quality >=7, 1, NA)))
# Dividimos los datos en el grupo de entramiento y el de test
set.seed(27)
index <- sample(1:nrow(data_wine),size = 0.8*nrow(data_wine))
train <- data_wine[index,]
test <- data_wine[-index,]

# Modelo de arbol de decision de tipo random forest
model_3 <- randomForest(quality ~., data = train)
model_3_pred <- predict(model_3, test)
model_3_matrix <- confusionMatrix(model_3_pred, test$quality)
model_3_matrix
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   0   1
##          0 263  23
##          1  13  21
##
##                Accuracy : 0.8875
##                  95% CI : (0.8477, 0.92)
##     No Information Rate : 0.8625
##     P-Value [Acc > NIR] : 0.1096
##
##                   Kappa : 0.4756
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 0.9529
##             Specificity : 0.4773
##          Pos Pred Value : 0.9196
##          Neg Pred Value : 0.6176
##              Prevalence : 0.8625
##          Detection Rate : 0.8219
##    Detection Prevalence : 0.8938
##       Balanced Accuracy : 0.7151
##
##        'Positive' Class : 0
##
```

```r
# Curva ROC
pred1 <- predict(model_3, test, type = "prob")
perf <- prediction(pred1[,2], test$quality)
auc <- performance(perf, "auc")
pred3 <- performance(perf, "tpr","fpr")
plot(pred3,main="ROC Curve for Random Forest",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```




```
cat("AUC:", auc@y.values[[1]])
```

```
## AUC: 0.9265481
```