Universitat Oberta de Catalunya

uoc.edu

TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

Práctica 2: Limpieza y validación de los datos

Autor: Francisco de Santos Bouzón
Fecha: Enero de 2019

INDICE

1. Detalles de la actividad

1.1 Descripción

En esta actividad se elabora un caso práctico, consistente en el tratamiento de un conjunto de datos (en inglés, dataset), orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

1.2 Objetivos

Los objetivos que se persiguen mediante el desarrollo de esta actividad práctica son los siguientes:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que permita continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.3 Competencias

Así, las competencias del Máster en Data Science que se desarrollan son:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

2. Resolución

2.1 Descripción del dataset

Este conjunto de datos está relacionado con las variantes blancas del vino portugués "Vinho Verde". Para más detalles, consulte la referencia [Cortez et al., 2009]. Debido a cuestiones de privacidad y logística, solo están disponibles las variables fisicoquímicas (entradas) y sensoriales (la salida) (por ejemplo, no hay datos sobre tipos de uva, marca de vino, precio de venta del vino, etc.). El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle, aunque utilizaré el enlace de UCI para realizar la carga de datos, y está constituido por 12 características (columnas) que presentan 4898 valores (filas o registros).

2.2 Definición de atributos

- fixed.acidity: la mayoría de los ácidos relacionados con el vino o fijos o no volátiles (no se evaporan fácilmente).
- volatile.acidity: la cantidad de ácido acético en el vino, que a niveles demasiado altos puede provocar un sabor desagradable a vinagre.
- citric.acid: se encuentra en pequeñas cantidades, el ácido cítrico puede agregar 'frescura' y sabor a los vinos.
- residual.sugar: la cantidad de azúcar restante después de que se detiene la fermentación.
- chlorides: la cantidad de sal en el vino.
- free.sulfur.dioxide: la forma libre de SO2 existe en equilibrio entre el SO2 molecular (como un gas disuelto) y el ion bisulfito; Previene el crecimiento microbiano y la oxidación del vino.
- total.sulfur.dioxide: cantidad de formas libres y ligadas de SO2.
- density: la densidad del agua es cercana a la del agua según el porcentaje de alcohol y contenido de azúcar.
- pH: describe qué tan ácido o básico es un vino en una escala de 0 (muy ácido) a 14 (muy básico).
- sulphates: un aditivo para el vino que puede contribuir a los niveles de gas de dióxido de azufre (S02), que actúa como antimicrobiano y antioxidante.
- alcohol: el porcentaje de alcohol del vino.
- quality: calidad del vino.

2.3 Importancia y objetivos del análisis

Nuestro objetivo será analizar el conjunto de datos indicado, para poder obtener los atributos que más influyen de forma positiva en la elaboración del Vinho Verde en la variedad de vino blanco.

También pretendemos establecer un modelo predictivo que nos ayude a evaluar la calidad del vino en función de los atributos que hayamos destacado como más influyentes para conseguir un mejor vino.

2.4 Proceso del estudio del juego de datos

El primero paso consistirá en instalar y cargar las librerías necesarias

```
# Cargamos las librerías
if(!require("ggplot2")){
  install.packages("ggplot2", repos='http://cran.us.r-project.org')
  library("ggplot2")
}
if(!require("dplyr")){
  install.packages("dplyr", repos='http://cran.us.r-project.org')
  library("dplyr")
}
if(!require("corrplot")){
  install.packages("corrplot", repos='http://cran.us.r-project.org')
  library("corrplot")
}
if(!require("nortest")){
  install.packages("nortest", repos='http://cran.us.r-project.org')
  library("nortest")
}
if(!require("grid")){
  install.packages("grid", repos='http://cran.us.r-project.org')
  library("grid")
}
if(!require("psych")){
  install.packages("psych", repos='http://cran.us.r-project.org')
  library("psych")
}
if(!require("Hmisc")){
  install.packages("Hmisc", repos='http://cran.us.r-project.org')
  library("Hmisc")
}
```

Cargamos el juego de datos

```
# Carga de datos
wine <- read.csv('http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv',sep=";",header=TRUE)

# Creamos la variable filas con la dimensión del dataset
filas=dim(wine)[1]

# Hacemos una copia del dataset para luego mostrar la correlación con ella
winecor <- wine
```

Comprobamos la estructura del dataset.

```
# Estructura y resumen
str(wine)
```

```
## '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 3.2
2 ...
##  $ 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 ...
```

```
summary(wine)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
```

```
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0
##  Median :0.04300   Median : 34.00      Median :134.0
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0
##     density             pH          sulphates         alcohol
##  Min.   :0.9871   Min.   :2.720   Min.   :0.2200   Min.   : 8.00
##  1st Qu.:0.9917   1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50
##  Median :0.9937   Median :3.180   Median :0.4700   Median :10.40
##  Mean   :0.9940   Mean   :3.188   Mean   :0.4898   Mean   :10.51
##  3rd Qu.:0.9961   3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40
##  Max.   :1.0390   Max.   :3.820   Max.   :1.0800   Max.   :14.20
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.878
##  3rd Qu.:6.000
##  Max.   :9.000
```

2.4.1 Selección de datos

El dataset que estamos tratando agrupa distintos factores que intervienen en la elaboración del vino y que, en función de las cantidades de cada uno de esos componentes, hará que los resultados nos den unos vinos de mejor o de peor calidad. Por tanto, ya que todos los atributos influirán a la hora de la elaboración, no vamos a prescindir de ninguno de ellos.

2.4.2 Ceros y elementos vacíos

Habitualmente, los datos desconocidos se rellenan con el valor '0', con una '?' o dejando espacios en blanco. Por ello, el primer análisis que vamos a realizar es comprobar si efectivamente se encuentran valores que no se hayan rellenado a la hora de realizar este estudio.

Buscamos si hay valores vacíos.

```
# Comprobamos si hay valores que sean '0'
colSums(is.na(wine))
```

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

```
##            sulphates              alcohol              quality
##                    0                    0                    0

# Comprobamos si hay valores que sean '?'
colSums(wine=="?")

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

# Comprobamos si hay valores que sean ' '
colSums(wine=="")

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

Tal y como podemos comprobar en los resultados obtenidos, no encontramos valores desconocidos en nuestro conjunto de datos. En caso de haberlos encontrado, deberíamos de decidir si prescindimos de ellos, si utilizamos su moda,...

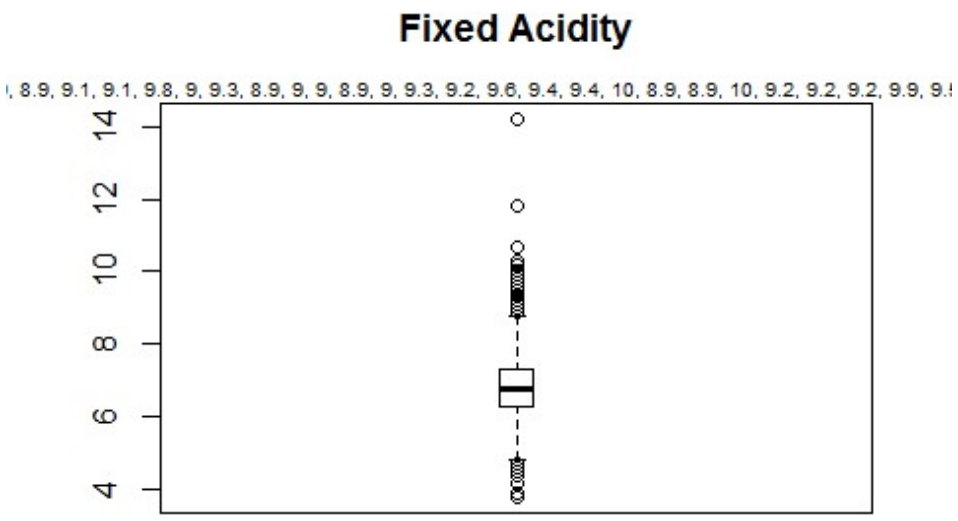
2.4.3 Localización de outliers

Los outliers, o valores extremos o atípicos, tal y como se puede encontrar en la definición de la wikipedia, es "una observación que es numéricamente distante del resto de los datos. Las estadísticas derivadas de los conjuntos de datos que incluyen valores atípicos serán frecuentemente engañosas". Voy a comprobar si efectivamente encontramos outliers en nuestro conjunto de datos. Todos nuestros datos son variables continuas, y sabemos que para una variable continua, los valores extremos son aquellas observaciones que se encuentran fuera de 1.5 * IQR, donde IQR, el 'Inter Quartile Range' es la diferencia entre 75 y 25 cuartiles. Voy a utilizar la función boxplot.stats para comprobar estos datos:

```
# Outliers de "fixed.acidity"
outlier_values <- boxplot.stats(wine$fixed.acidity)$out
outlier_values
```

```
##   [1]  9.8  9.8 10.2  9.1 10.0  9.2  9.2  9.0  9.1  9.2 10.3  9.4  9.2
9.8
##  [15]  9.6  9.2  9.0  9.3  9.2  9.1  8.9  9.8  8.9  9.2  9.7  9.4 10.3
9.6
##  [29]  9.0  9.7  9.2  9.4  9.6  9.2  9.0  9.2 10.7 10.7  9.0  9.2  9.8
9.2
##  [43] 14.2  8.9  8.9  9.1  9.1  9.8  9.0  9.3  8.9  9.0  9.0  8.9  9.0
9.3
##  [57]  9.2  9.6  9.4  9.4 10.0  8.9  8.9 10.0  9.2  9.2  9.2  9.9  9.5
9.0
##  [71]  9.0  8.9  9.5 11.8  9.4  9.1  9.8  9.9  9.2  8.9  9.2  9.4  9.4
9.4
##  [85]  4.6  8.9  9.4  9.2  9.2  9.8  9.0  9.0  9.0  8.9  8.9  4.5  9.2
9.6
##  [99]  4.2  9.7  9.7  9.0  4.2  9.4  8.9  8.9  8.9  4.7  4.7  3.8  4.4
4.7
## [113]  9.0  9.0  4.7  4.4  3.9  4.7  4.4
```

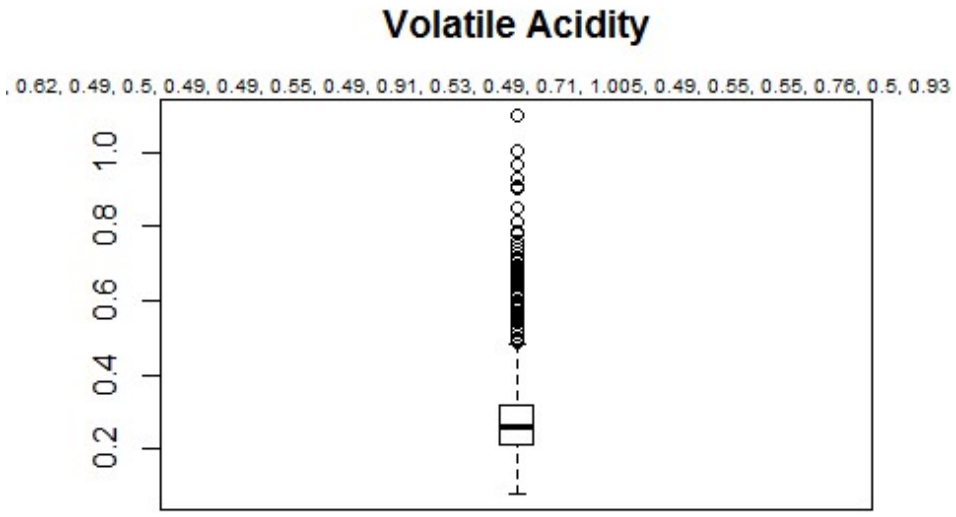
```
boxplot(wine$fixed.acidity, main="Fixed Acidity", boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "volatile.acidity"
outlier_values <- boxplot.stats(wine$volatile.acidity)$out
outlier_values
```

```
##   [1] 0.660 0.660 0.670 0.540 0.595 0.670 0.530 0.540 0.570 0.685 0.49
5
##  [12] 0.640 0.520 0.580 0.585 0.590 0.600 0.580 0.590 0.550 0.905 0.55
0
##  [23] 0.490 0.550 0.520 0.600 0.550 0.510 0.620 0.510 0.560 0.570 0.67
0
##  [34] 0.500 0.560 0.560 0.655 0.595 0.705 0.520 0.550 0.600 0.640 0.68
0
##  [45] 0.490 0.510 0.550 0.520 0.500 0.550 0.600 0.610 0.610 0.610 0.66
0
##  [56] 0.570 0.500 0.500 0.590 0.580 0.540 0.580 0.570 0.640 0.560 0.49
0
##  [67] 0.490 0.670 0.550 0.560 0.520 0.520 0.850 0.510 0.620 0.510 0.53
0
##  [78] 0.640 0.550 0.490 0.490 0.610 0.545 0.620 0.490 0.500 0.490 0.49
0
##  [89] 0.550 0.490 0.910 0.530 0.490 0.710 1.005 0.490 0.550 0.550 0.76
0
## [100] 0.500 0.930 0.490 0.495 0.695 0.705 0.815 0.560 0.560 0.560 0.51
0
## [111] 0.540 0.540 0.500 0.615 0.500 0.520 0.600 0.680 0.655 0.510 0.51
0
## [122] 0.615 0.615 0.965 0.740 0.530 0.780 0.680 0.640 0.540 0.750 0.64
0
## [133] 0.640 0.655 0.580 0.520 0.530 0.600 0.530 0.580 0.670 0.610 0.73
0
## [144] 0.650 0.580 1.100 0.500 0.500 0.500 0.650 0.520 0.550 0.585 0.56
0
## [155] 0.555 0.555 0.540 0.610 0.550 0.530 0.660 0.615 0.500 0.620 0.50
0
## [166] 0.490 0.510 0.510 0.540 0.610 0.695 0.695 0.630 0.630 0.690 0.69
0
## [177] 0.590 0.620 0.785 0.760 0.500 0.540 0.520 0.600 0.540 0.530
```

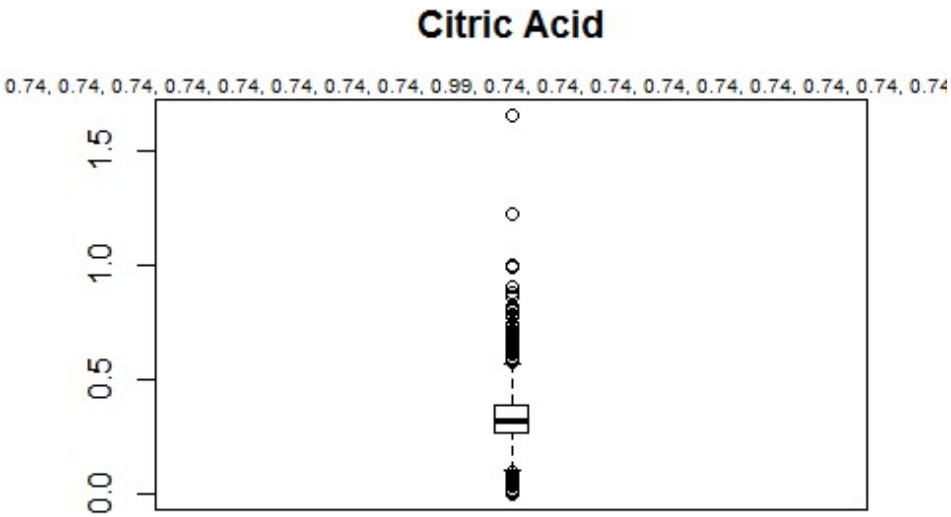
```
boxplot(wine$volatile.acidity,   main="Volatile   Acidity",   boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```r
# Outliers de "citric.acid"
outlier_values <- boxplot.stats(wine$citric.acid)$out
outlier_values
```

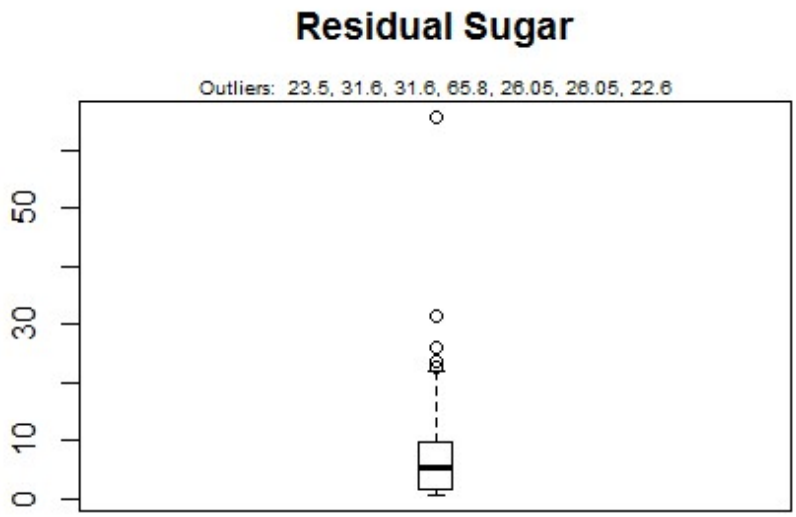
```
##   [1] 0.62 0.04 0.59 0.07 0.03 0.61 0.62 0.63 0.61 0.62 0.63 0.66 0.66
0.00
##  [15] 0.04 0.67 0.67 0.04 0.04 0.07 0.88 0.08 0.59 0.07 0.07 0.07 0.07
0.58
##  [29] 0.70 0.00 0.00 0.60 0.07 0.09 0.04 0.62 0.58 0.62 0.70 0.62 0.62
0.58
##  [43] 0.02 0.65 0.65 0.71 0.66 0.66 0.07 0.06 0.07 0.06 0.68 0.68 0.68
0.68
##  [57] 0.06 0.72 0.69 0.58 0.70 1.66 0.04 0.63 0.60 0.00 0.08 0.58 0.58
0.05
##  [71] 0.58 0.00 0.00 0.65 0.58 0.00 0.05 0.05 0.62 0.62 0.58 0.58 1.00
0.09
##  [85] 0.01 0.71 0.71 0.60 0.06 0.74 0.81 0.69 0.58 0.69 0.00 0.07 0.64
0.72
##  [99] 0.73 0.65 0.68 0.65 0.74 0.71 0.59 0.68 0.08 0.72 0.64 0.02 0.74
0.74
## [113] 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74 0.74
0.74
## [127] 0.74 0.74 0.74 0.74 0.74 0.99 0.74 0.74 0.74 0.74 0.74 0.74 0.74
0.74
## [141] 0.74 0.74 0.74 0.74 0.74 0.74 0.01 0.74 0.01 0.74 0.74 1.00 0.04
0.58
## [155] 0.07 1.00 0.00 0.58 0.61 0.61 0.61 0.02 0.67 0.67 0.67 0.58 0.65
0.58
## [169] 0.09 0.08 0.71 0.04 0.03 0.05 0.64 0.64 0.58 0.58 0.81 0.58 0.61
0.62
## [183] 0.59 0.00 0.04 0.63 0.73 0.68 0.09 0.78 0.79 0.09 0.64 0.65 0.65
0.00
## [197] 0.73 0.73 0.64 0.60 0.71 0.72 0.82 0.07 0.58 0.58 1.00 0.66 0.80
0.80
## [211] 1.23 0.59 0.02 0.00 1.00 0.62 0.00 0.71 0.71 0.71 0.61 0.61 0.00
0.60
## [225] 0.58 0.09 0.09 0.72 0.62 0.62 0.79 0.82 0.67 0.01 0.01 0.86 0.61
0.02
## [239] 0.05 0.00 0.69 0.69 0.59 0.01 0.66 0.66 0.78 0.00 0.04 0.91 0.91
0.06
## [253] 0.06 0.04 0.04 0.74 0.09 0.09 0.60 0.62 0.73 0.00 0.09 0.00 0.09
0.67
## [267] 0.01 0.09 0.00 0.02
```

```
boxplot(wine$citric.acid, main="Citric Acid", boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "residual.sugar"
outlier_values <- boxplot.stats(wine$residual.sugar)$out
outlier_values

## [1] 23.50 31.60 31.60 65.80 26.05 26.05 22.60
```

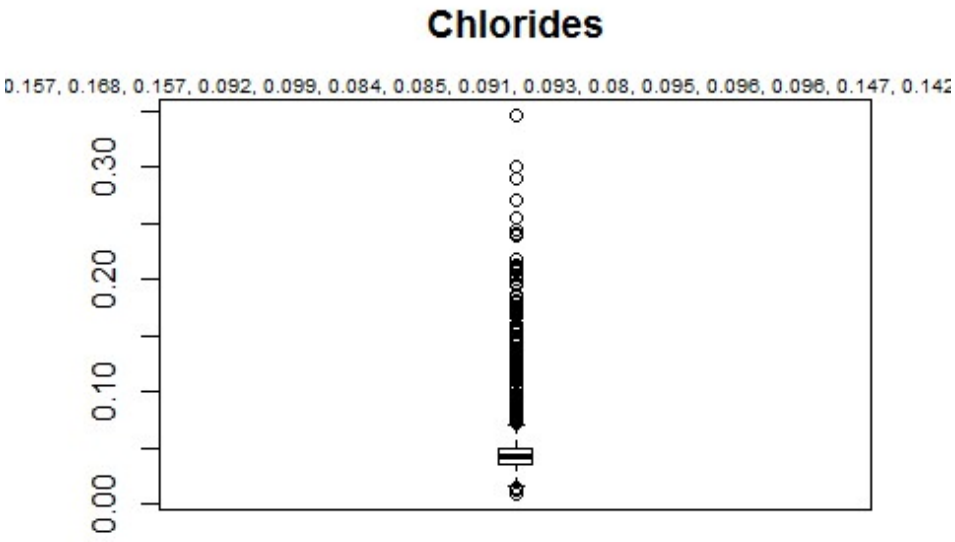
```
boxplot(wine$residual.sugar,     main="Residual     Sugar",     boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "chlorides"
outlier_values <- boxplot.stats(wine$chlorides)$out
outlier_values
```

```
##   [1] 0.074 0.080 0.172 0.173 0.147 0.092 0.082 0.092 0.200 0.197 0.19
7
##  [12] 0.074 0.132 0.089 0.108 0.081 0.073 0.346 0.090 0.114 0.186 0.18
0
##  [23] 0.084 0.083 0.096 0.094 0.240 0.290 0.185 0.110 0.078 0.130 0.13
5
##  [34] 0.115 0.072 0.170 0.080 0.119 0.126 0.150 0.152 0.088 0.244 0.13
7
##  [45] 0.093 0.077 0.079 0.073 0.072 0.076 0.201 0.201 0.074 0.074 0.30
1
##  [56] 0.138 0.169 0.083 0.093 0.168 0.122 0.172 0.167 0.239 0.076 0.13
8
##  [67] 0.137 0.123 0.123 0.133 0.073 0.073 0.211 0.123 0.123 0.255 0.20
4
##  [78] 0.208 0.083 0.080 0.076 0.086 0.084 0.084 0.168 0.160 0.179 0.07
6
##  [89] 0.076 0.087 0.217 0.094 0.157 0.157 0.148 0.158 0.157 0.168 0.15
7
## [100] 0.092 0.099 0.084 0.085 0.091 0.093 0.080 0.095 0.096 0.096 0.14
7
## [111] 0.142 0.079 0.074 0.075 0.074 0.121 0.121 0.079 0.079 0.014 0.15
6
## [122] 0.012 0.119 0.119 0.081 0.170 0.171 0.082 0.074 0.083 0.083 0.15
2
## [133] 0.169 0.073 0.014 0.078 0.112 0.154 0.126 0.126 0.104 0.142 0.10
2
## [144] 0.184 0.184 0.096 0.076 0.146 0.117 0.117 0.118 0.014 0.085 0.08
7
## [155] 0.085 0.087 0.076 0.088 0.160 0.167 0.014 0.009 0.098 0.098 0.08
6
## [166] 0.086 0.194 0.094 0.013 0.144 0.149 0.185 0.084 0.175 0.090 0.09
8
## [177] 0.110 0.110 0.095 0.174 0.097 0.142 0.145 0.208 0.209 0.105 0.08
6
## [188] 0.176 0.176 0.108 0.096 0.271 0.120 0.212 0.094 0.094 0.117 0.17
3
## [199] 0.074 0.076 0.076 0.175 0.174 0.075 0.127 0.127 0.096 0.136
```

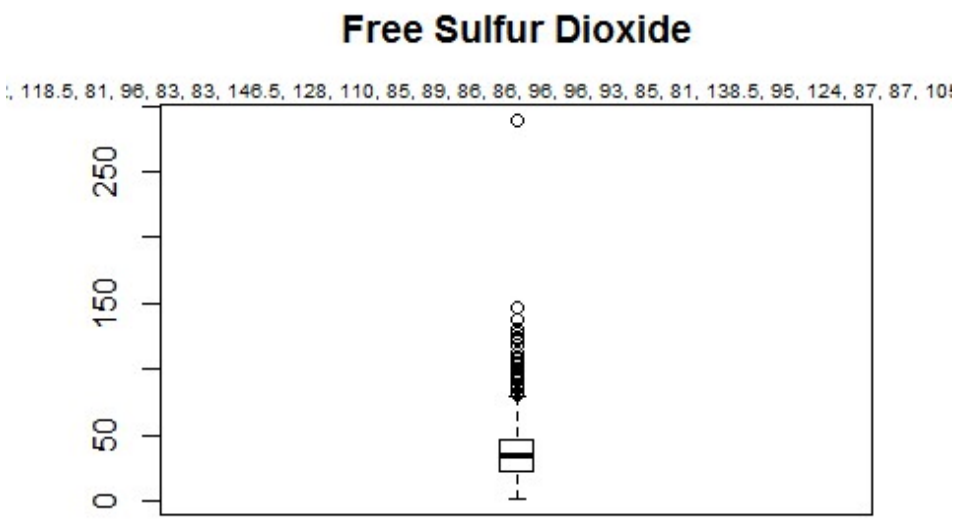
```
boxplot(wine$chlorides,           main="Chlorides",           boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "free.sulfur.dioxide"
outlier_values <- boxplot.stats(wine$free.sulfur.dioxide)$out
outlier_values
```

```
##  [1]  81.0  82.0 131.0  82.5  87.0  87.0  83.0 122.5  83.0  81.0  88.0
## [12]  82.0 118.5  81.0  96.0  83.0  83.0 146.5 128.0 110.0  85.0  89.0
## [23]  86.0  86.0  96.0  96.0  93.0  85.0  81.0 138.5  95.0 124.0  87.0
## [34]  87.0 105.0 105.0 101.0 101.0 108.0 108.0  98.0  98.0 112.0 108.0
## [45]  98.0  81.0  81.0  81.0 289.0  97.0
```

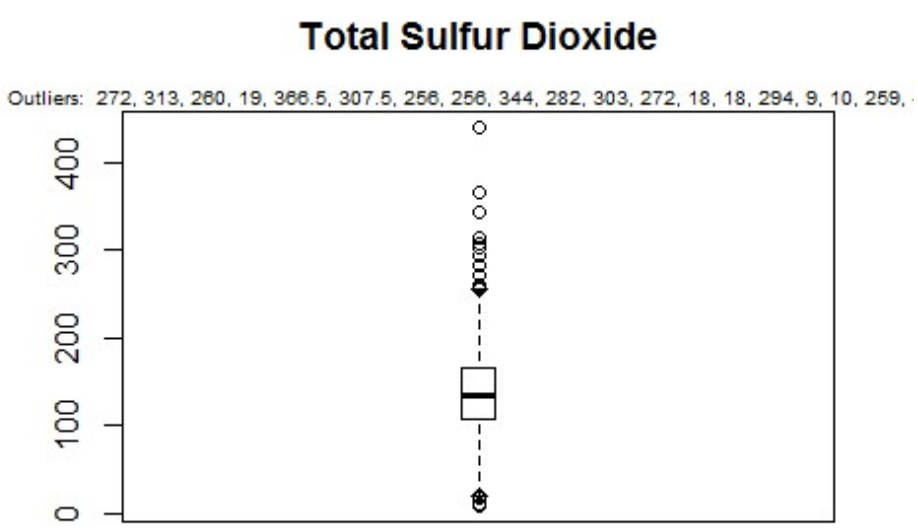
```
boxplot(wine$free.sulfur.dioxide, main="Free Sulfur Dioxide", boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



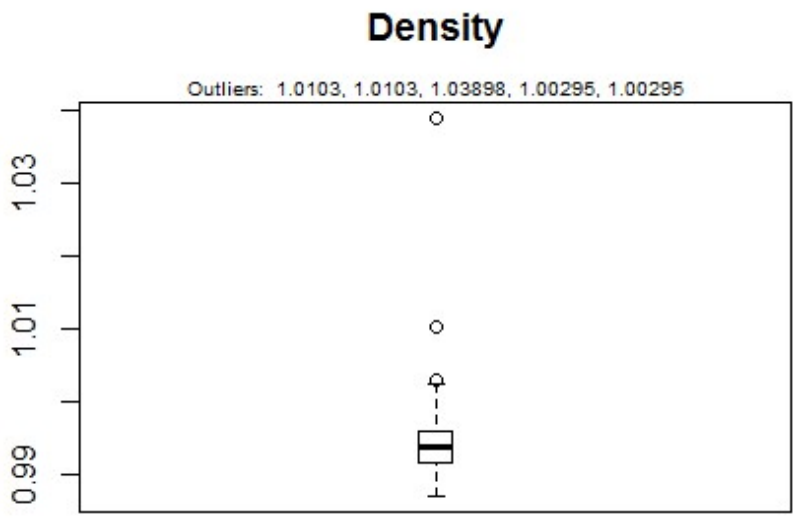
```
# Outliers de "total.sulfur.dioxide"
outlier_values <- boxplot.stats(wine$total.sulfur.dioxide)$out
outlier_values
```

```
##  [1] 272.0 313.0 260.0  19.0 366.5 307.5 256.0 256.0 344.0 282.0 303.0
## [12] 272.0  18.0  18.0 294.0   9.0  10.0 259.0 440.0
```

```
boxplot(wine$total.sulfur.dioxide,    main="Total    Sulfur    Dioxide",
boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



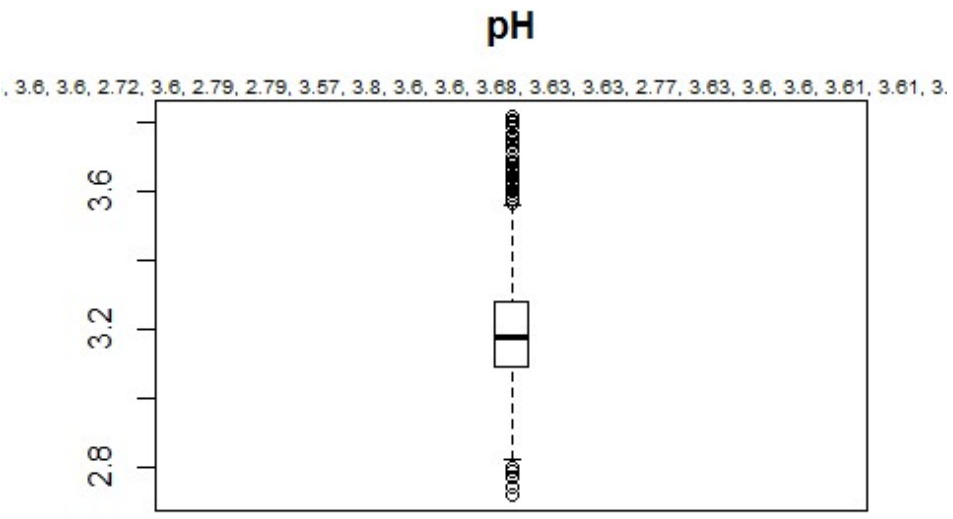
```
# Outliers de "density"
outlier_values <- boxplot.stats(wine$density)$out
outlier_values

## [1] 1.01030 1.01030 1.03898 1.00295 1.00295

boxplot(wine$density,              main="Density",              boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "pH"
outlier_values <- boxplot.stats(wine$pH)$out
outlier_values
```

```
##  [1] 3.69 3.63 3.72 3.61 3.64 3.64 3.72 3.72 3.58 3.58 3.66 3.59 2.74
3.82
## [15] 3.81 3.65 3.65 3.59 3.77 3.62 3.63 3.58 3.58 3.65 3.74 2.80 3.60
3.60
## [29] 2.72 3.60 2.79 2.79 3.57 3.80 3.60 3.60 3.68 3.63 3.63 2.77 3.63
3.60
## [43] 3.60 3.61 3.61 3.59 3.79 3.59 3.68 3.59 3.66 3.70 3.74 3.80 3.57
3.57
## [57] 3.57 3.65 3.58 2.80 3.77 3.76 3.69 3.66 3.59 2.79 3.75 3.63 3.75
3.76
## [71] 3.66 3.66 2.80 3.67 3.57
```

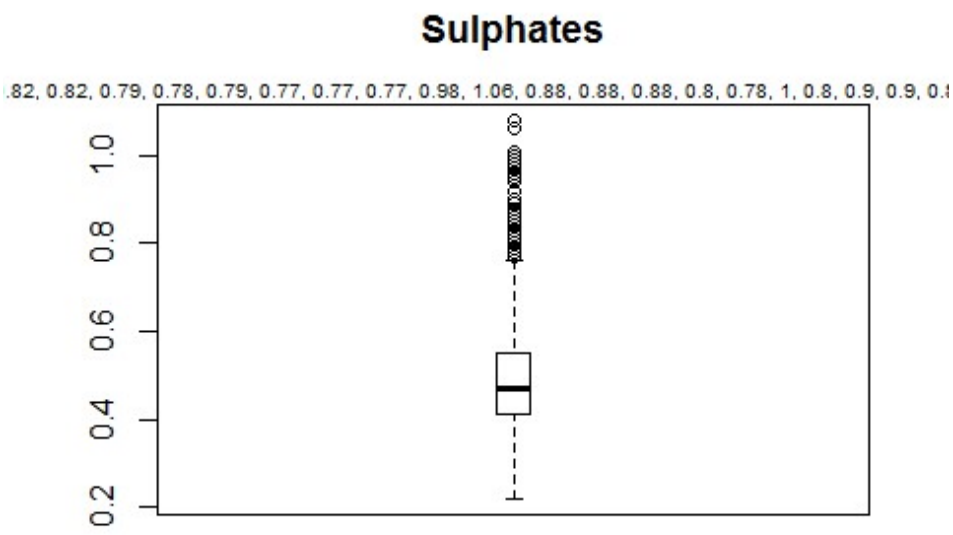
```
boxplot(wine$pH,                    main="pH",                    boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "sulphates"
outlier_values <- boxplot.stats(wine$sulphates)$out
outlier_values
```

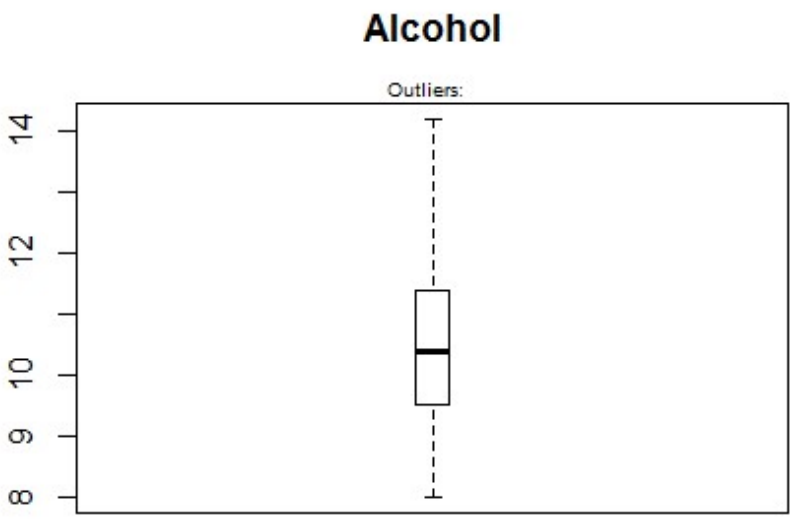
```
##   [1] 0.77 0.84 0.77 0.79 0.85 0.78 0.79 0.79 0.79 0.77 0.78 0.85 0.96
0.97
##  [15] 0.82 0.82 0.77 0.95 0.95 0.77 0.95 0.82 0.82 0.90 0.88 0.88 0.79
0.80
##  [29] 0.80 0.78 0.78 0.87 0.86 0.90 0.90 0.78 0.79 0.81 0.81 0.77 0.82
0.79
##  [43] 0.79 0.77 0.82 0.92 0.79 0.79 0.82 0.82 0.82 0.82 0.82 0.79 0.78
0.79
##  [57] 0.77 0.77 0.77 0.98 1.06 0.88 0.88 0.88 0.80 0.78 1.00 0.80 0.90
0.90
##  [71] 0.89 0.94 0.99 0.86 0.84 0.95 0.84 0.84 0.81 0.80 0.87 0.82 0.78
0.78
##  [85] 0.78 0.78 0.78 0.77 0.85 0.78 0.78 0.88 0.88 0.78 0.78 0.78 0.78
0.79
##  [99] 0.77 0.77 0.83 0.83 0.81 0.81 0.98 0.98 0.98 0.98 0.79 0.79 0.78
0.82
## [113] 0.98 0.77 0.96 1.01 0.77 0.96 0.77 0.92 0.94 0.95 1.08 0.79
```

```
boxplot(wine$sulphates,          main="Sulphates",          boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



```
# Outliers de "alcohol"
outlier_values <- boxplot.stats(wine$alcohol)$out
outlier_values

## numeric(0)
```

```
boxplot(wine$alcohol,              main="Alcohol",              boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```



Se puede apreciar, salvo en alcohol y densidad, que existen bastantes outliers. Pero este conjunto de datos muestra distintas combinaciones que harán que la calidad del vino varíe. Lo que para un atributo aparezca como un valor atípico hará que el vino tome unas características que producirán un vino de un tipo en particular, con una calidad distinta, por lo que contaremos con todos los valores que muestra el conjunto de datos. Los valores indicados como valores extremos, para nuestro caso, son valores posibles que marcarán que el vino adquiera cierta calidad.

Como se indica en la propia descripción del dataset "las entradas incluyen pruebas objetivas", y por ello vamos a mantener todos los outliers como valores posibles, ya que son las medidas normales que se suelen utilizar. Indicamos el texto del conjunto de datos original:

"In the above reference, two datasets were created, using red and white wine samples. The inputs include objective tests (e.g. PH values) and the output is based on sensory data (median of at least 3 evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent)."

2.4.4 Comprobación de la normalidad

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de Anderson Darling. Así, se comprueba que para cada prueba se obtiene un p-valor superior al nivel de significación prefijado alpha = 0,05. Si esto se cumple, entonces se considera que variable en cuestión sigue una distribución normal.

```
alpha = 0.05
col.names = colnames(wine)
for (i in 1:ncol(wine)) {
    if (i == 1) cat("Variables que no siguen una distribución normal:\n")
    if (is.integer(wine[,i]) | is.numeric(wine[,i])) {
        p_val = ad.test(wine[,i])$p.value
        if (p_val < alpha) {
            cat(col.names[i])
            if (i < ncol(wine) - 1) cat(", ")
            if (i %% 3 == 0) cat("\n")
        }
    }
  if (i == 11)
    break()
}
```

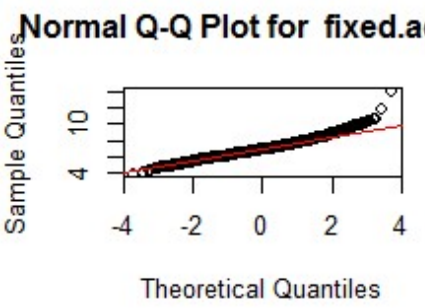
```
## Variables que no siguen una distribución normal:
## fixed.acidity, volatile.acidity, citric.acid,
## residual.sugar, chlorides, free.sulfur.dioxide,
## total.sulfur.dioxide, density, pH,
## sulphates, alcohol
```

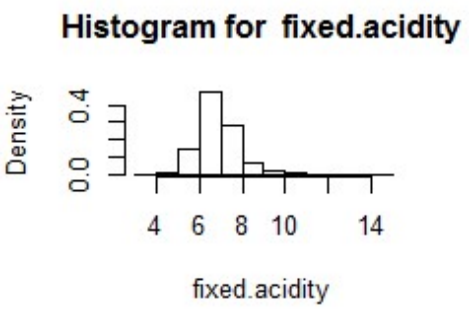
Comprobamos que ninguna de las variables siguen una distribución normal.

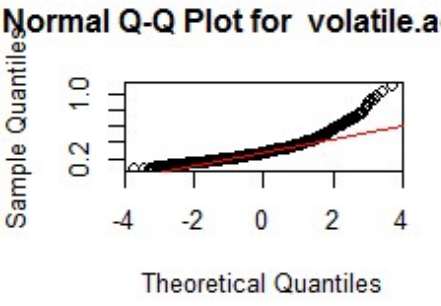
Para verificar lo ya indicado, nos vamos a apoyar en las gráficas Q-Q (Cuantil - Cuantil), las cuales nos ayudarán a comparar gráficamente dos distribuciones. El gráfico Q-Q ayuda a comparar gráficamente dos distribuciones comparando los cuantiles de dos distribuciones. Lo que haremos será comparar nuestros datos con los valores teóricos de una distribución normal y si, nuestros datos siguen una distribución normal, el gráfico será como una línea recta. Para ello utilizaremos dos funciones que tenemos en R, qqnorm y qqline: * Con la función qqnorm podemos generar un gráfico Q-Q que comparará los cuantiles de los datos que tenemos sobre el vino con los cuantiles teóricos de la distribución normal estándar. * Con la función qqline podemos superponer una línea para ayudarnos a evaluar la relación lineal de las dos distribuciones. Dicha línea cruza los puntos del primer y del tercer cuartil.

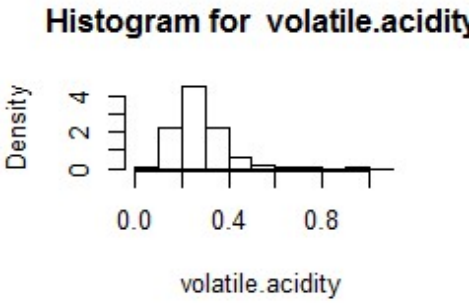
Pasemos a comprobarlo:

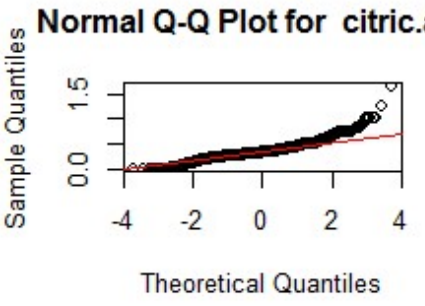
```
par(mfrow=c(2,2))
for(i in 1:ncol(wine)) {
  if (is.numeric(wine[,i])){
    qqnorm(wine[,i],main = paste("Normal Q-Q Plot for
",colnames(wine)[i]))
    qqline(wine[,i],col="red")
    hist(wine[,i],
      main=paste("Histogram for ", colnames(wine)[i]),
      xlab=colnames(wine)[i], freq = FALSE)
  }
  if (i == 11)
```

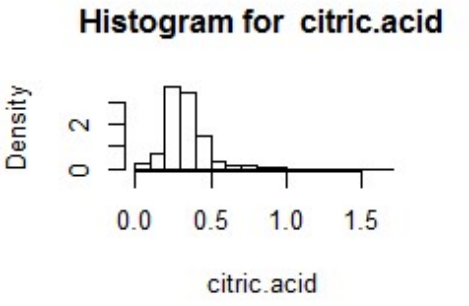

Normal Q-Q Plot for fixed.acid
Sample Quantiles
4 10
-4 -2 0 2 4
Theoretical Quantiles

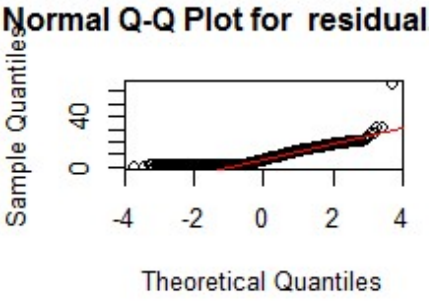

Histogram for fixed.acidity
Density
0.0 0.4
4 6 8 10 14
fixed.acidity

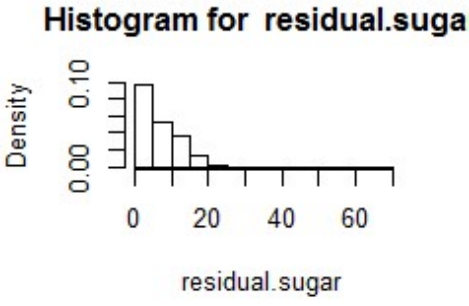

Normal Q-Q Plot for volatile.aci
Sample Quantiles
0.2 1.0
-4 -2 0 2 4
Theoretical Quantiles

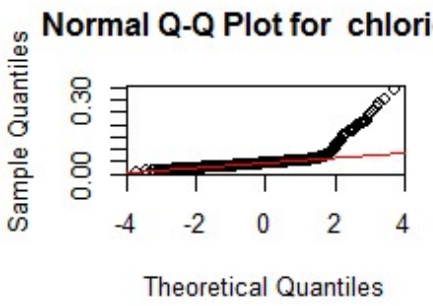

Histogram for volatile.acidity
Density
0 2 4
0.0 0.4 0.8
volatile.acidity

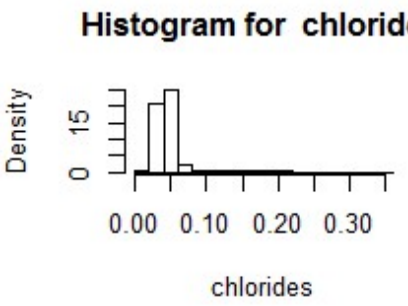

Normal Q-Q Plot for citric.aci
Sample Quantiles
0.0 1.5
-4 -2 0 2 4
Theoretical Quantiles

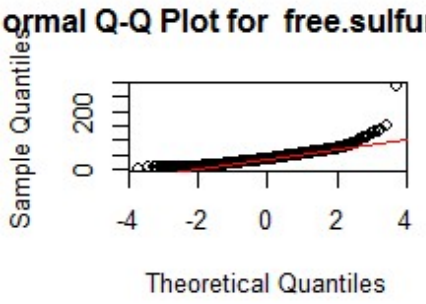

Histogram for citric.acid
Density
0 2
0.0 0.5 1.0 1.5
citric.acid

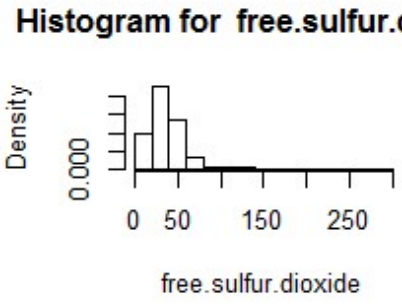

Normal Q-Q Plot for residual.su
Sample Quantiles
0 40
-4 -2 0 2 4
Theoretical Quantiles

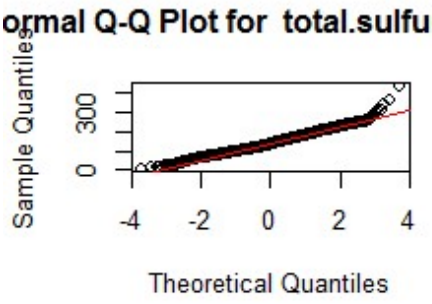

Histogram for residual.suga
Density
0.00 0.10
0 20 40 60
residual.sugar

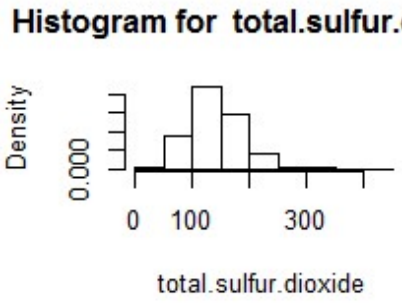

Normal Q-Q Plot for chloride
Sample Quantiles
0.00
0.30
-4
-2
0
2
4
Theoretical Quantiles

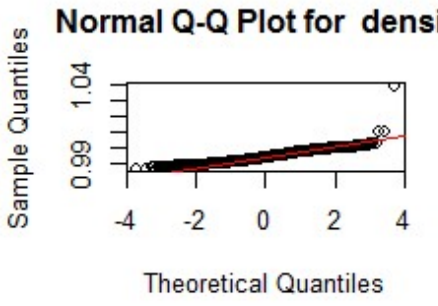

Histogram for chlorides
Density
0
15
0.00
0.10
0.20
0.30
chlorides

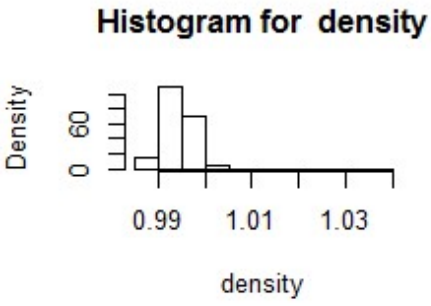

ormal Q-Q Plot for free.sulfur.di
Sample Quantiles
0
200
-4
-2
0
2
4
Theoretical Quantiles

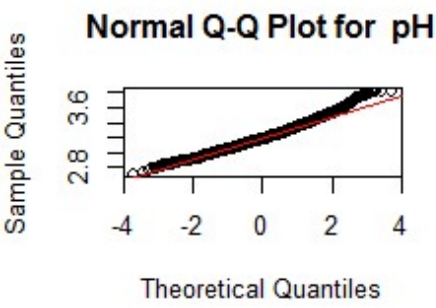

Histogram for free.sulfur.dioxi
Density
0.000
0
50
150
250
free.sulfur.dioxide

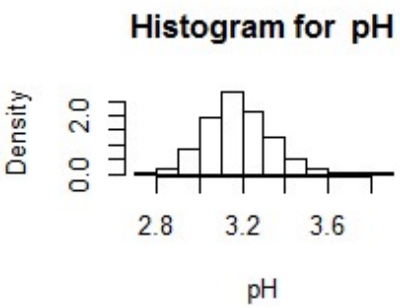

ormal Q-Q Plot for total.sulfur.di
Sample Quantiles
0
300
-4
-2
0
2
4
Theoretical Quantiles

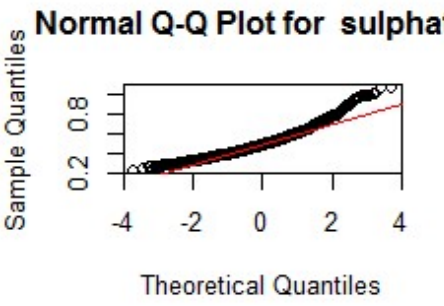

Histogram for total.sulfur.dioxi
Density
0.000
0
100
300
total.sulfur.dioxide

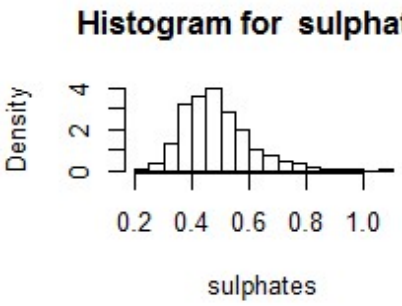

Normal Q-Q Plot for density
Sample Quantiles
0.99
1.04
-4
-2
0
2
4
Theoretical Quantiles

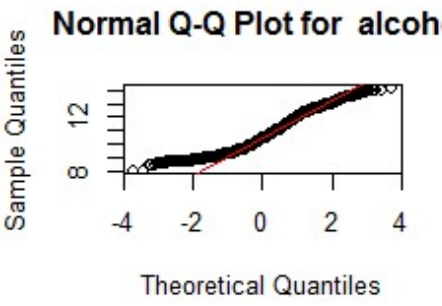

Histogram for density
Density
0
60
0.99
1.01
1.03
density

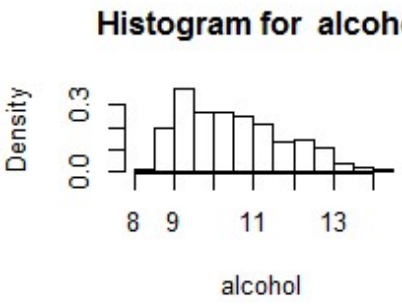

Normal Q-Q Plot for pH
Sample Quantiles
2.8
3.6
-4
-2
0
2
4
Theoretical Quantiles


Histogram for pH
Density
0.0
2.0
2.8
3.2
3.6
pH


Normal Q-Q Plot for sulphate
Sample Quantiles
0.2
0.8
-4
-2
0
2
4
Theoretical Quantiles


Histogram for sulphates
Density
0
2
4
0.2
0.4
0.6
0.8
1.0
sulphates


Normal Q-Q Plot for alcohol
Sample Quantiles
8
12
-4
-2
0
2
4
Theoretical Quantiles


Histogram for alcohol
Density
0.0
0.3
8
9
11
13
alcohol

En las distintas gráficas observamos que todas las variables tienen alguna desviación con respecto a la línea que indica lo que sería la distribución normal y, por ello, podemos resumir diciendo que ninguna de ellas sigue una distribución normal.

Vamos a realizar otra comprobación para verificar este punto. Para ello, utilizaremos el test de Shapiro-Wilk para contrastar la normalidad, ya que dicho test se usa para contrastar si un conjunto de datos siguen una distribución normal o no. Realizaremos el siguiente contraste de hipótesis:

- H0: los datos provienen de una distribución normal
- H1: los datos no provienen de una distribución normal

```
shapiro.test(wine$fixed.acidity)

##
##  Shapiro-Wilk normality test
##
## data:  wine$fixed.acidity
## W = 0.97656, p-value < 2.2e-16

shapiro.test(wine$volatile.acidity)

##
##  Shapiro-Wilk normality test
##
## data:  wine$volatile.acidity
## W = 0.90455, p-value < 2.2e-16

shapiro.test(wine$citric.acid)

##
##  Shapiro-Wilk normality test
##
## data:  wine$citric.acid
## W = 0.92225, p-value < 2.2e-16

shapiro.test(wine$residual.sugar)

##
##  Shapiro-Wilk normality test
##
## data:  wine$residual.sugar
## W = 0.88457, p-value < 2.2e-16

shapiro.test(wine$chlorides)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  wine$chlorides
## W = 0.59081, p-value < 2.2e-16

shapiro.test(wine$free.sulfur.dioxide)

##
##  Shapiro-Wilk normality test
##
## data:  wine$free.sulfur.dioxide
## W = 0.94207, p-value < 2.2e-16

shapiro.test(wine$total.sulfur.dioxide)

##
##  Shapiro-Wilk normality test
##
## data:  wine$total.sulfur.dioxide
## W = 0.98901, p-value < 2.2e-16

shapiro.test(wine$density)

##
##  Shapiro-Wilk normality test
##
## data:  wine$density
## W = 0.9548, p-value < 2.2e-16

shapiro.test(wine$pH)

##
##  Shapiro-Wilk normality test
##
## data:  wine$pH
## W = 0.9881, p-value < 2.2e-16

shapiro.test(wine$sulphates)

##
##  Shapiro-Wilk normality test
##
## data:  wine$sulphates
## W = 0.95161, p-value < 2.2e-16

shapiro.test(wine$alcohol)

##
##  Shapiro-Wilk normality test
##
## data:  wine$alcohol
## W = 0.9553, p-value < 2.2e-16
```

Obtenemos en todos los casos que p-value es menor a 0.05, por lo que rechazamos la hipótesis nula y confirmamos que las variables no siguen una distribución normal.

2.4.5 Análisis de la homogeneidad de la varianza

Para realizar el análisis de la homogeneidad de la varianza, debido a que ya hemos comprobado que las variables no siguen una distribución normal, vamos a utilizar pruebas no paramétricas. Es por ello que utilizaremos el test de Fligner-Killeen, el cual es un test no paramétrico que compara las varianzas basándose en la mediana.

Lo primero que vamos a hacer es agrupar los valores de la calidad del vino. Hemos optamos por definir 3 valores, bueno, regular y malo.

```
winecor$quality <- ifelse(winecor$quality < 6, 'bad', ifelse(winecor$qual
ity == 6, 'normal', 'good'))

winecor$quality <- as.factor(winecor$quality)
```

Vamos a estudiar esta homogeneidad en cuanto a los grupos conformados por la calidad del vino frente a cada una de las variables restantes. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(winecor$fixed.acidity, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$fixed.acidity and winecor$quality
## Fligner-Killeen:med chi-squared = 9.5241, df = 2, p-value =
## 0.008548

fligner.test(winecor$volatile.acidity, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$volatile.acidity and winecor$quality
## Fligner-Killeen:med chi-squared = 35.003, df = 2, p-value =
## 2.507e-08

fligner.test(winecor$citric.acid, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$citric.acid and winecor$quality
## Fligner-Killeen:med chi-squared = 297.83, df = 2, p-value <
## 2.2e-16
```

```
fligner.test(winecor$residual.sugar, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$residual.sugar and winecor$quality
## Fligner-Killeen:med chi-squared = 140.83, df = 2, p-value <
## 2.2e-16

fligner.test(winecor$chlorides, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$chlorides and winecor$quality
## Fligner-Killeen:med chi-squared = 26.364, df = 2, p-value =
## 1.884e-06

fligner.test(winecor$free.sulfur.dioxide, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$free.sulfur.dioxide and winecor$quality
## Fligner-Killeen:med chi-squared = 195.56, df = 2, p-value <
## 2.2e-16

fligner.test(winecor$total.sulfur.dioxide, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$total.sulfur.dioxide and winecor$quality
## Fligner-Killeen:med chi-squared = 137.15, df = 2, p-value <
## 2.2e-16

fligner.test(winecor$density, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$density and winecor$quality
## Fligner-Killeen:med chi-squared = 33.606, df = 2, p-value =
## 5.043e-08

fligner.test(winecor$pH, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
```

```
## data:  winecor$pH and winecor$quality
## Fligner-Killeen:med chi-squared = 33.09, df = 2, p-value =
## 6.527e-08

fligner.test(winecor$sulphates, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$sulphates and winecor$quality
## Fligner-Killeen:med chi-squared = 82.885, df = 2, p-value <
## 2.2e-16

fligner.test(winecor$alcohol, winecor$quality)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  winecor$alcohol and winecor$quality
## Fligner-Killeen:med chi-squared = 229.36, df = 2, p-value <
## 2.2e-16
```

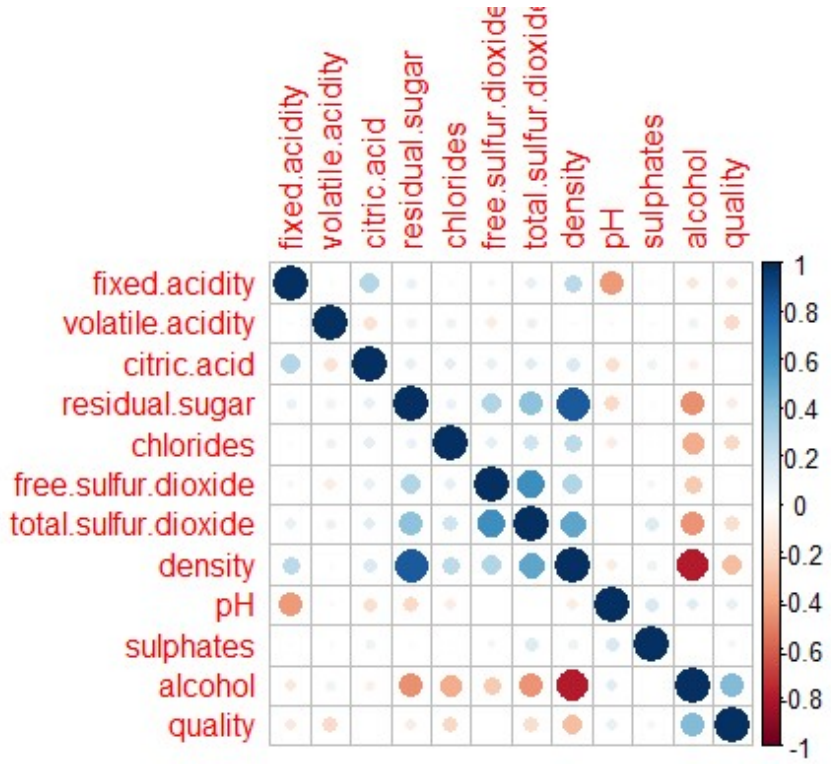
Hemos obtenido en todos los casos un p-valor inferior a 0,05, rechazando la hipótesis de que las varianzas de las muestras sean homogéneas.

2.5 Pruebas estadísticas

2.5.1 Variables que influyen más en la calidad del vino

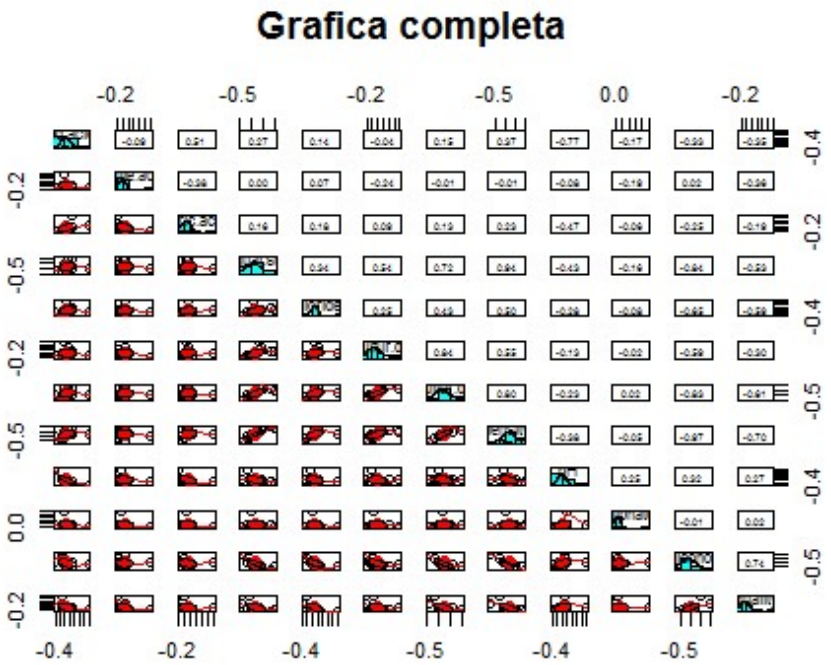
La correlación es una técnica estadística usada para determinar la relación entre dos o más variables. La correlación puede ser de al menos dos variables o de una variable dependiente y dos o más variables independientes, denominada correlación múltiple. El coeficiente de correlación es un valor cuantitativo de la relación entre dos o más variables. El coeficiente de correlación puede variar desde -1.00 hasta 1.00. La correlación de proporcionalidad directa o positiva se establece con los valores +1.00 y de proporcionalidad inversa o negativa, con -1.00. No existe relación entre las variables cuando el coeficiente es de 0.00. Vamos a estudiar dicho coeficiente mostrando una tabla que relacione todas las variables del estudio y así comprobar la relación existente entre todas las variables entre sí.

```
# Tabla de correlación
winecor <- wine
correlation<-cor(winecor[,c("fixed.acidity","volatile.acidity","citric.ac
id","residual.sugar","chlorides","free.sulfur.dioxide","total.sulfur.diox
ide","density","pH","sulphates","alcohol", "quality")])
corrplot(correlation,method = "circle")
```



Veamos una gráfica completa.

```
pairs.panels(correlation , pch=21,main="Grafica completa")
```



Vamos a resumirlo con cifras para que nos quede más claro.

```
rcorr(as.matrix(wine))
```

```
##                       fixed.acidity volatile.acidity citric.acid
## fixed.acidity                  1.00            -0.02        0.29
## volatile.acidity              -0.02             1.00       -0.15
## citric.acid                    0.29            -0.15        1.00
## residual.sugar                 0.09             0.06        0.09
## chlorides                      0.02             0.07        0.11
## free.sulfur.dioxide           -0.05            -0.10        0.09
## total.sulfur.dioxide           0.09             0.09        0.12
## density                        0.27             0.03        0.15
## pH                            -0.43            -0.03       -0.16
## sulphates                     -0.02            -0.04        0.06
## alcohol                       -0.12             0.07       -0.08
## quality                       -0.11            -0.19       -0.01
##                       residual.sugar chlorides free.sulfur.dioxide
## fixed.acidity                   0.09      0.02                -0.05
## volatile.acidity                0.06      0.07                -0.10
## citric.acid                     0.09      0.11                 0.09
```

```
## residual.sugar                 1.00      0.09                0.30
## chlorides                      0.09      1.00                0.10
## free.sulfur.dioxide            0.30      0.10                1.00
## total.sulfur.dioxide           0.40      0.20                0.62
## density                        0.84      0.26                0.29
## pH                            -0.19     -0.09                0.00
## sulphates                     -0.03      0.02                0.06
## alcohol                       -0.45     -0.36               -0.25
## quality                       -0.10     -0.21                0.01
##                       total.sulfur.dioxide density    pH sulphates alco
hol
## fixed.acidity                         0.09    0.27 -0.43     -0.02   -0
.12
## volatile.acidity                      0.09    0.03 -0.03     -0.04    0
.07
## citric.acid                           0.12    0.15 -0.16      0.06   -0
.08
## residual.sugar                        0.40    0.84 -0.19     -0.03   -0
.45
## chlorides                             0.20    0.26 -0.09      0.02   -0
.36
## free.sulfur.dioxide                   0.62    0.29  0.00      0.06   -0
.25
## total.sulfur.dioxide                  1.00    0.53  0.00      0.13   -0
.45
## density                               0.53    1.00 -0.09      0.07   -0
.78
## pH                                    0.00   -0.09  1.00      0.16    0
.12
## sulphates                             0.13    0.07  0.16      1.00   -0
.02
## alcohol                              -0.45   -0.78  0.12     -0.02    1
.00
## quality                              -0.17   -0.31  0.10      0.05    0
.44
##                       quality
## fixed.acidity           -0.11
## volatile.acidity        -0.19
## citric.acid             -0.01
## residual.sugar          -0.10
## chlorides               -0.21
## free.sulfur.dioxide      0.01
## total.sulfur.dioxide    -0.17
## density                 -0.31
## pH                       0.10
## sulphates                0.05
## alcohol                  0.44
## quality                  1.00
```

```
##
## n= 4898
##
##
## P
##                      fixed.acidity volatile.acidity citric.acid
## fixed.acidity                        0.1122           0.0000
## volatile.acidity     0.1122                           0.0000
## citric.acid          0.0000          0.0000
## residual.sugar       0.0000          0.0000           0.0000
## chlorides            0.1062          0.0000           0.0000
## free.sulfur.dioxide  0.0005          0.0000           0.0000
## total.sulfur.dioxide 0.0000          0.0000           0.0000
## density              0.0000          0.0578           0.0000
## pH                   0.0000          0.0255           0.0000
## sulphates            0.2303          0.0124           0.0000
## alcohol              0.0000          0.0000           0.0000
## quality              0.0000          0.0000           0.5193
##                      residual.sugar chlorides free.sulfur.dioxide
## fixed.acidity        0.0000          0.1062    0.0005
## volatile.acidity     0.0000          0.0000    0.0000
## citric.acid          0.0000          0.0000    0.0000
## residual.sugar                       0.0000    0.0000
## chlorides            0.0000                    0.0000
## free.sulfur.dioxide  0.0000          0.0000
## total.sulfur.dioxide 0.0000          0.0000    0.0000
## density              0.0000          0.0000    0.0000
## pH                   0.0000          0.0000    0.9655
## sulphates            0.0620          0.2408    0.0000
## alcohol              0.0000          0.0000    0.0000
## quality              0.0000          0.0000    0.5681
##                      total.sulfur.dioxide density pH     sulphates alc
ohol
## fixed.acidity        0.0000               0.0000  0.0000 0.2303    0.0
000
## volatile.acidity     0.0000               0.0578  0.0255 0.0124    0.0
000
## citric.acid          0.0000               0.0000  0.0000 0.0000    0.0
000
## residual.sugar       0.0000               0.0000  0.0000 0.0620    0.0
000
## chlorides            0.0000               0.0000  0.0000 0.2408    0.0
000
## free.sulfur.dioxide  0.0000               0.0000  0.9655 0.0000    0.0
000
## total.sulfur.dioxide                      0.0000  0.8710 0.0000    0.0
000
## density              0.0000                       0.0000 0.0000    0.0
```

```
000
## pH                   0.8710              0.0000         0.0000    0.0
000
## sulphates            0.0000              0.0000  0.0000           0.2
225
## alcohol              0.0000              0.0000  0.0000 0.2225
## quality              0.0000              0.0000  0.0000 0.0002    0.0
000
##                      quality
## fixed.acidity        0.0000
## volatile.acidity     0.0000
## citric.acid          0.5193
## residual.sugar       0.0000
## chlorides            0.0000
## free.sulfur.dioxide  0.5681
## total.sulfur.dioxide 0.0000
## density              0.0000
## pH                   0.0000
## sulphates            0.0002
## alcohol              0.0000
## quality
```

Vamos a considerar que existe una alta correlación positiva cuando el valor obtenido está entre 0.80 y 1, y una alta correlación negativa cuando dicho valor se encuentra entre -0.80 y -1. En este sentido hemos obtenido:

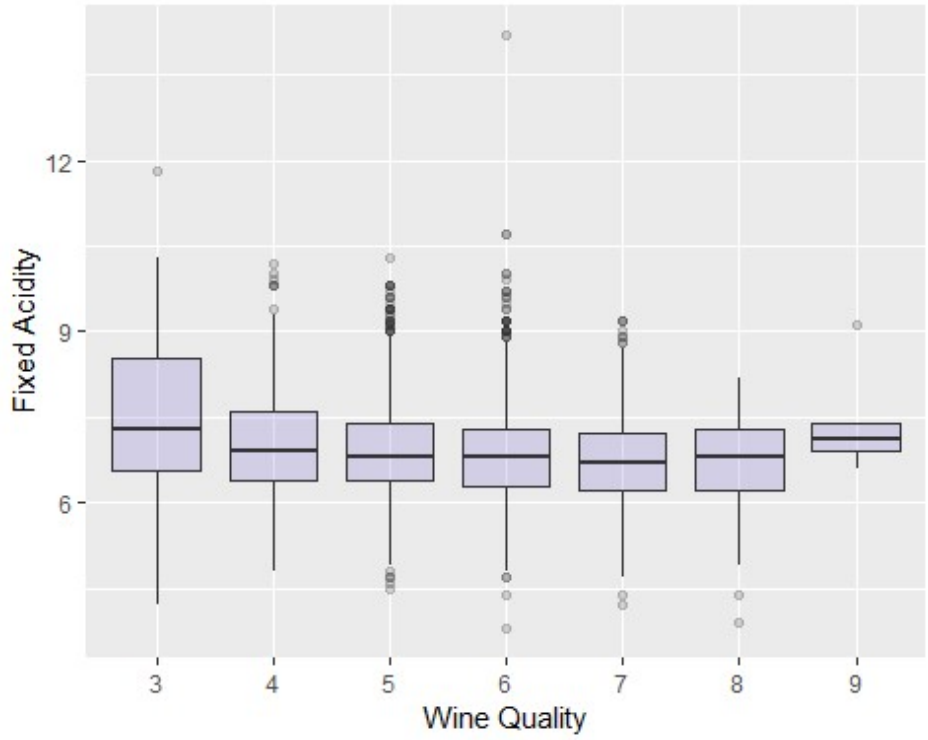
- Alta correlación positiva: density y residual sugar (0.94), total sulfur dioxide y free sulfur dioxide (0.83), y total sulfur dioxide y density (0.80)
- Alta correlación negativa: alcohol y density (-0.96), alcohol y residual sugar (-0.84), y total sulfur dioxide y alcohol (-0.83)

Estos resultados nos indican que podríamos prescindir de uno de los pares en los valores más altos, como density - residual sugar y alcohol - density, ya que ambos pares nos darán información altamente correlacionada.

Si observamos la correlación existente entre nuestras variables con respecto a quality, encontramos una correlación positiva bastante alta con alcohol (0.74), así como negativa con density (-0.69) y total sulfur dioxide (-0.60)

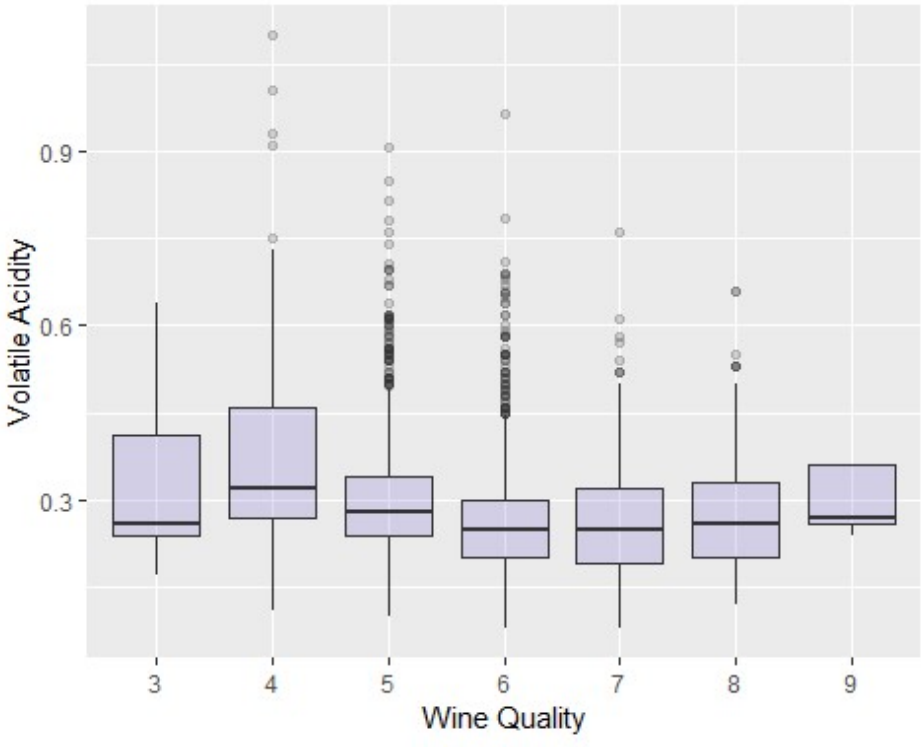
Vamos a centrarnos en la relación de la calidad con el resto de atributos para ver si se cumple lo mencionado anteriormente, así como para obtener información más concluyente.

```
ggplot(winecor, aes(x=as.factor(quality), y=fixed.acidity)) + geom_boxplo
t(fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Fixed Acid
ity")
```



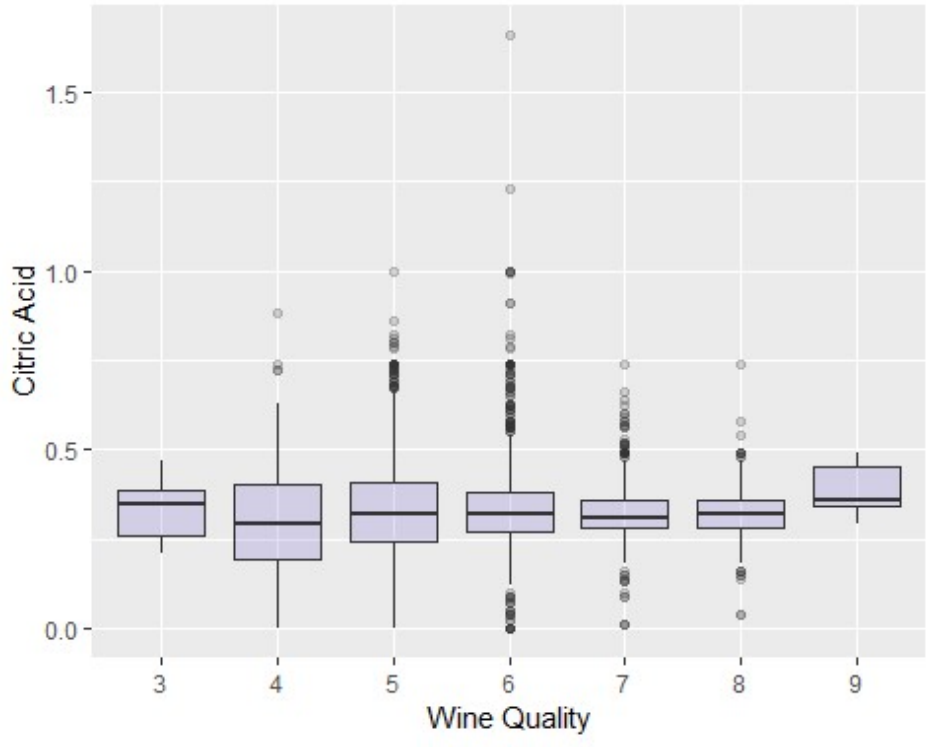
No muestra nada significativo. “Fixed Acidity” no tiene influencia en la calidad.

```
ggplot(winecor, aes(x=as.factor(quality), y=volatile.acidity)) + geom_box
plot(fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Volatil
e Acidity")
```



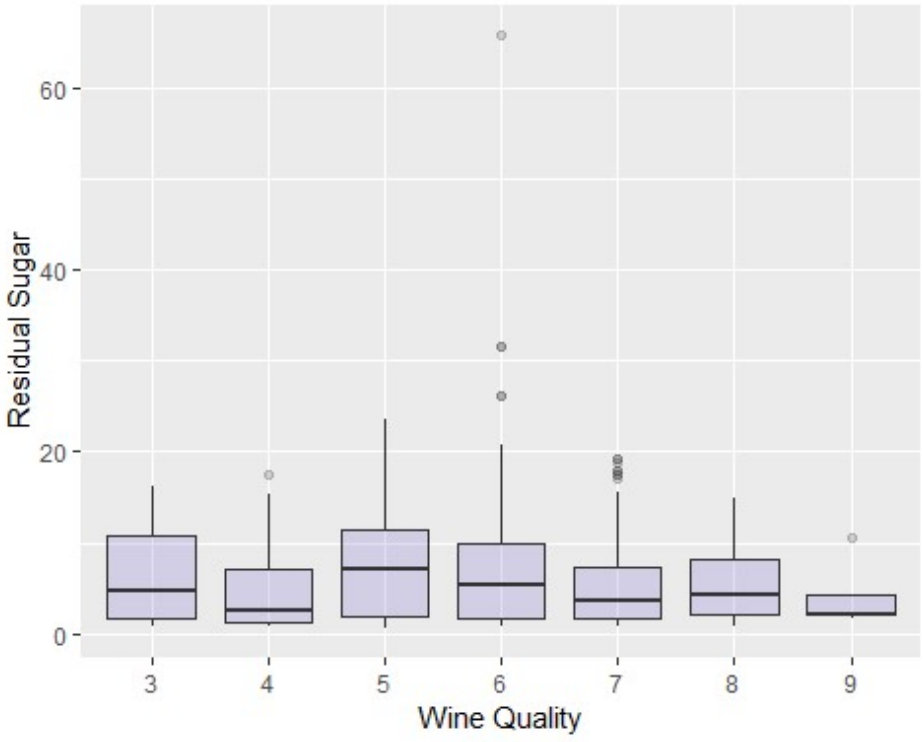
Demostramos que cuanto menor sea “Volatile Acidity”, peor calidad tendrá el vino.

```
ggplot(winecor, aes(x=as.factor(quality), y=citric.acid)) + geom_boxplot(
fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Citric Acid"
)
```



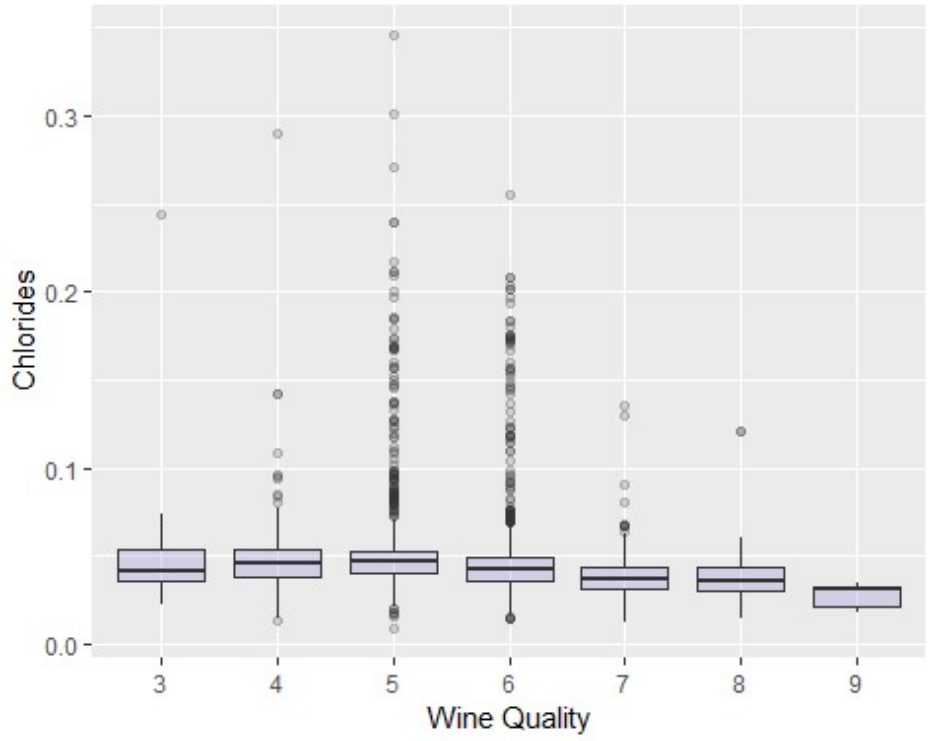
No muestra nada significativo. Apenas influye “Citric Acid” en la calidad.

```
ggplot(winecor, aes(x=as.factor(quality), y=residual.sugar)) + geom_boxplot(fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Residual Sugar")
```



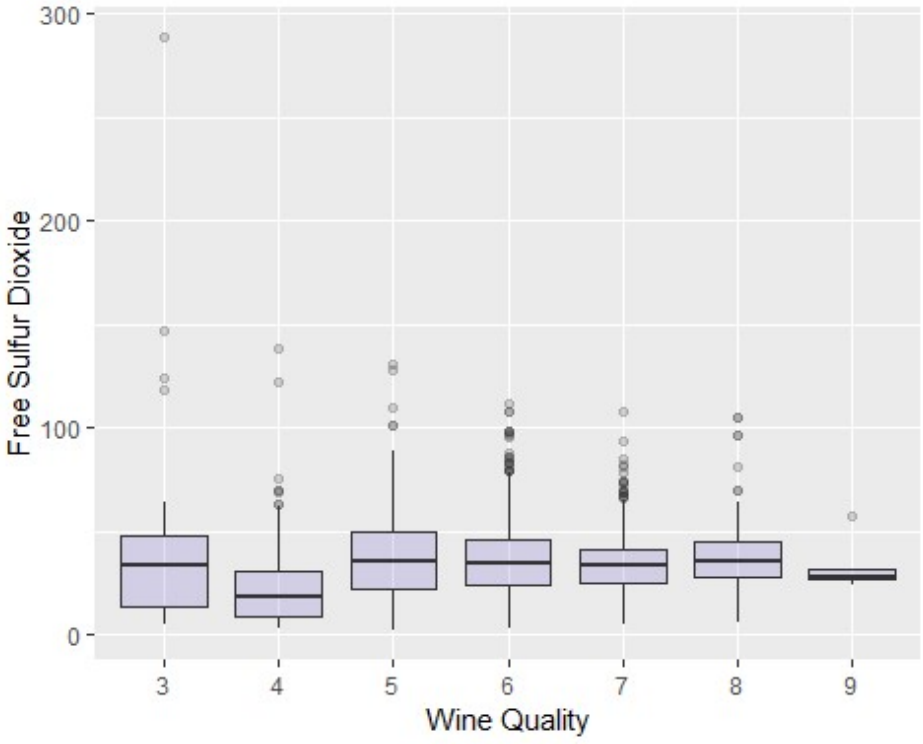
Se observa que prácticamente se mantiene constante la calidad del vino con respecto al "Residual Sugar", aunque la mayor calidad se consigue en vino con mucho menos "Residual Sugar".

```
ggplot(winecor, aes(x=as.factor(quality), y=chlorides)) + geom_boxplot(fi
ll="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Chlorides")
```



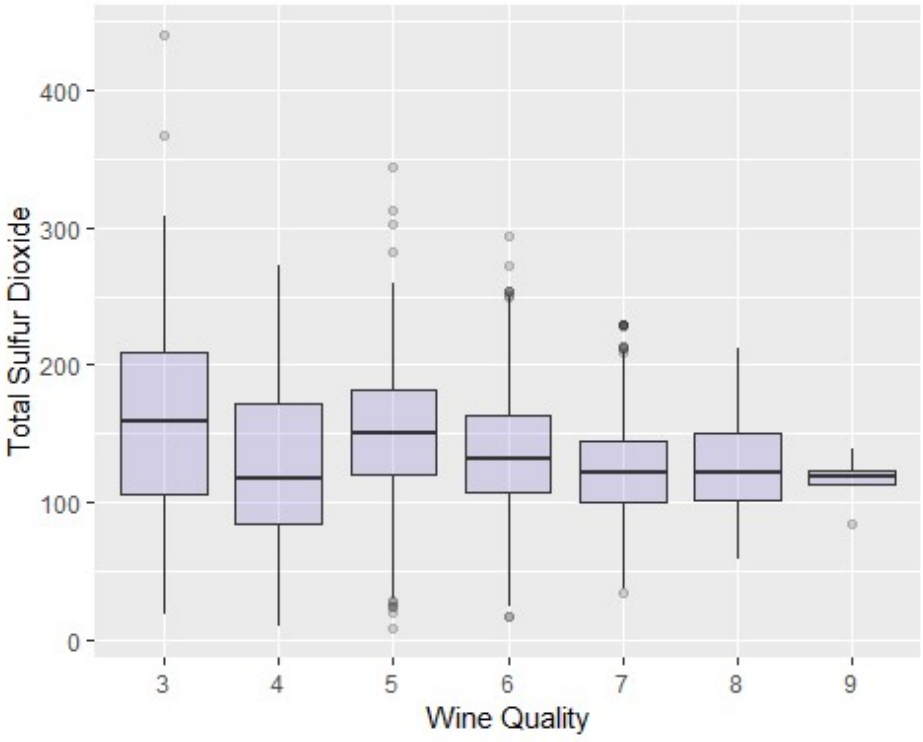
Comprobamos que cuanto menor sea la cantidad de “Chlorides”, mejor será la calidad del vino.

```
ggplot(winecor, aes(x=as.factor(quality), y=free.sulfur.dioxide)) + geom_
boxplot(fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Free
Sulfur Dioxide")
```



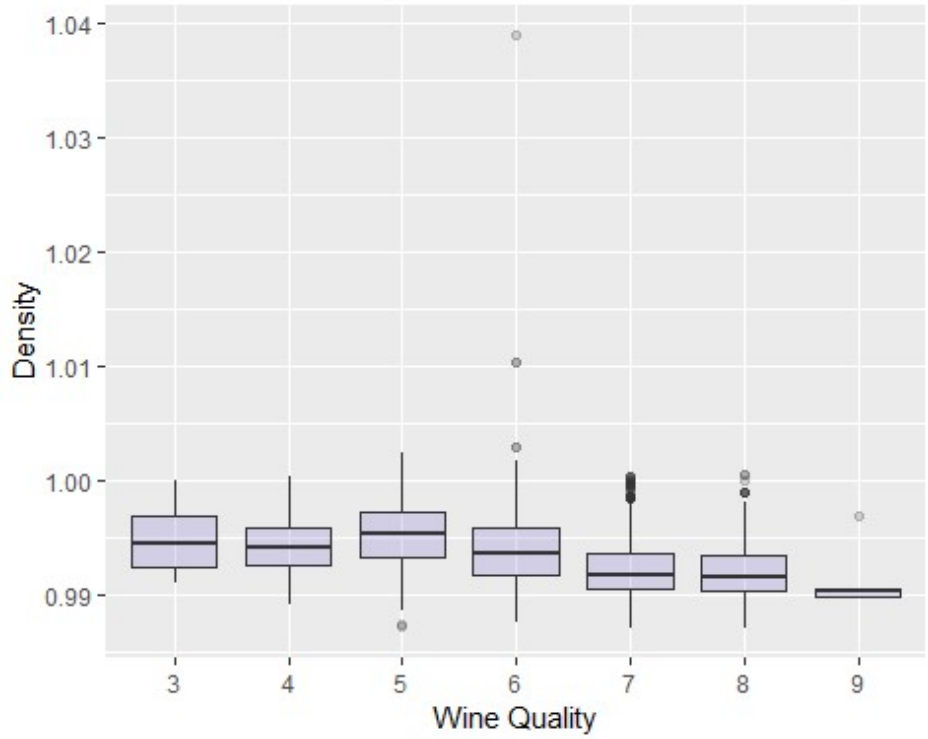
Gráfica un tanto complicada, aunque podemos observar que los mejores y peores vinos tienen menor concentración de "Free Sulfur Dioxide".

```
ggplot(winecor, aes(x=as.factor(quality), y=total.sulfur.dioxide)) + geom
_boxplot(fill="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Tot
al Sulfur Dioxide")
```



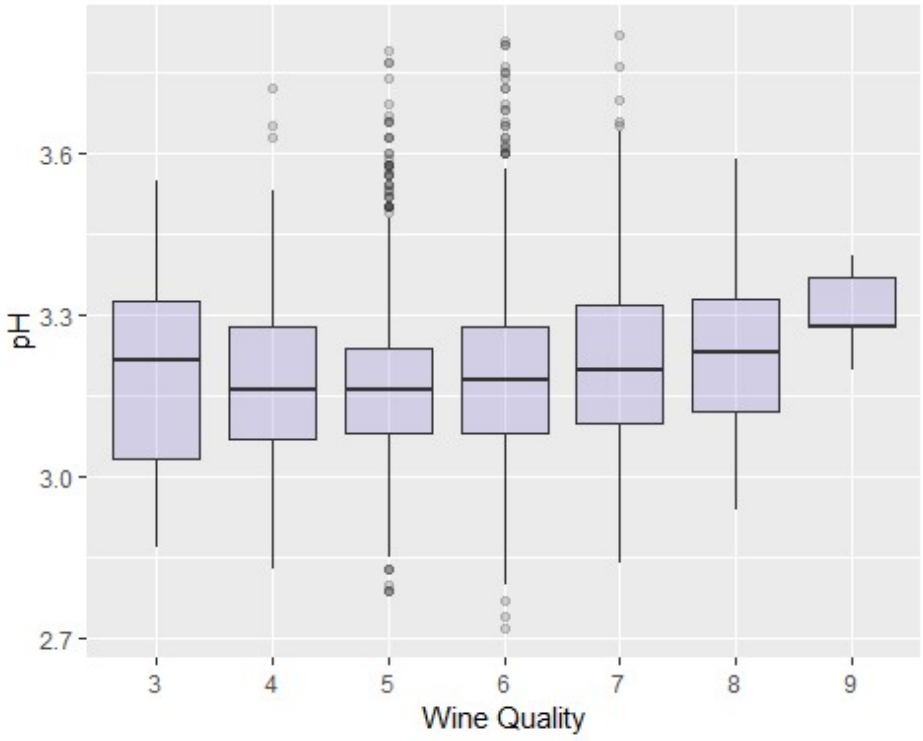
Cuanto mayor sea "Total Sulfur Dioxide", peor será el vino.

```
ggplot(winecor, aes(x=as.factor(quality), y=density)) + geom_boxplot(fill
="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Density")
```



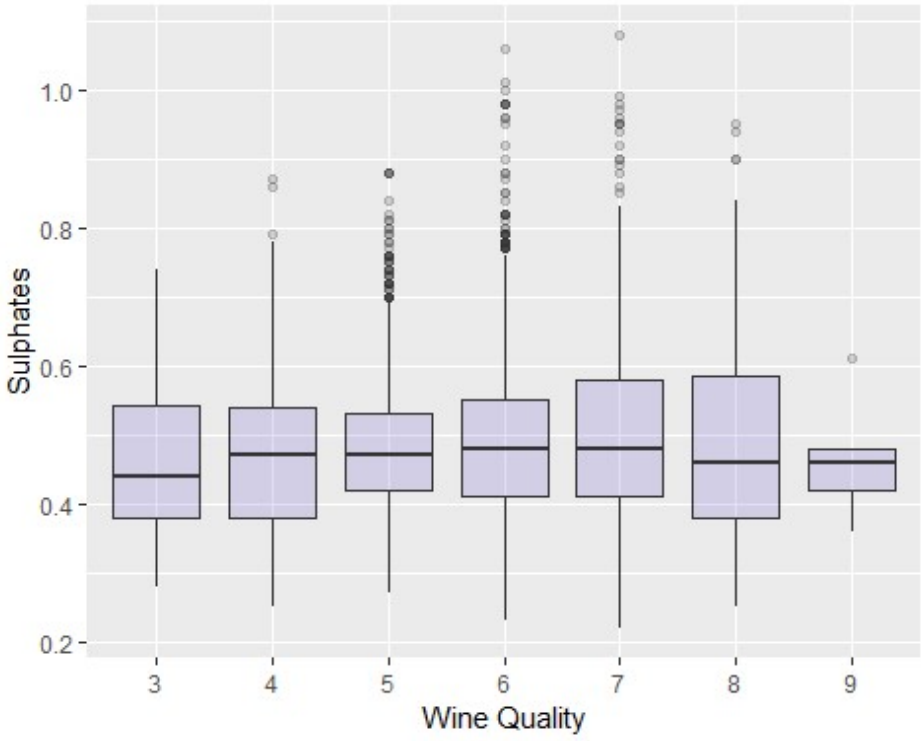
Comprobamos que “Density” influye significativamente en la calidad del vino. Menor densidad, mejor calidad.

```
ggplot(winecor, aes(x=as.factor(quality), y=pH)) + geom_boxplot(fill="sla
teblue", alpha=0.2) +  xlab("Wine Quality") + ylab("pH")
```



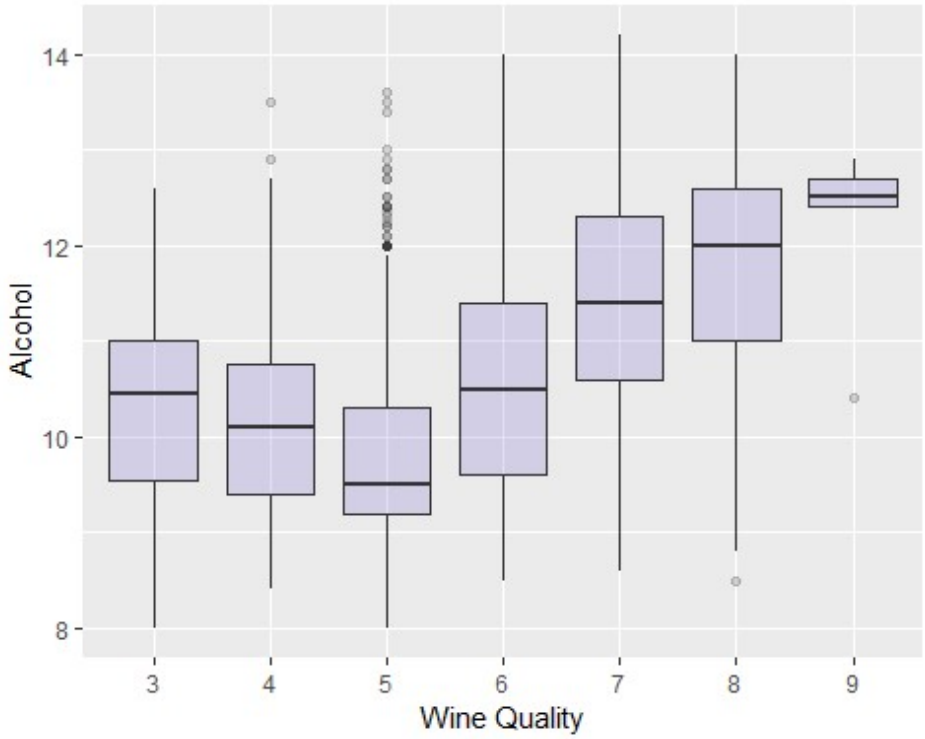
Comprobamos como el pH influye en la calidad del vino, ya que en función de los distintos valores aportados, obtendremos una calidad u otra.

```
ggplot(winecor, aes(x=as.factor(quality), y=sulphates)) + geom_boxplot(fi
ll="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Sulphates")
```



Se observa que se mantiene sin cambios significativos, menos que los vinos con la mejor calidad tienen menos “Sulphates”.

```
ggplot(winecor, aes(x=as.factor(quality), y=alcohol)) + geom_boxplot(fill
="slateblue", alpha=0.2) +  xlab("Wine Quality") + ylab("Alcohol")
```



Comprobamos que a mayor cantidad de “Alcohol”, los vinos serán de mayor calidad.

Para comprobar si las afirmaciones realizadas se cumplen, vamos a realizar la prueba de Kruskal-Wallis. Dicha prueba es un método no paramétrico para probar si un grupo de datos proviene de la misma población. Si tenemos K muestras aleatorias independientes posiblemente de distintos tamaños de k poblaciones distintas.

```
kruskal.test(winecor$fixed.acidity, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$fixed.acidity and winecor$quality
## Kruskal-Wallis chi-squared = 40.868, df = 6, p-value = 3.075e-07

kruskal.test(winecor$volatile.acidity, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$volatile.acidity and winecor$quality
## Kruskal-Wallis chi-squared = 286.13, df = 6, p-value < 2.2e-16

kruskal.test(winecor$citric.acid, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$citric.acid and winecor$quality
## Kruskal-Wallis chi-squared = 13.12, df = 6, p-value = 0.04117

kruskal.test(winecor$residual.sugar, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$residual.sugar and winecor$quality
## Kruskal-Wallis chi-squared = 94.519, df = 6, p-value < 2.2e-16

kruskal.test(winecor$chlorides, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$chlorides and winecor$quality
## Kruskal-Wallis chi-squared = 512.99, df = 6, p-value < 2.2e-16

kruskal.test(winecor$free.sulfur.dioxide, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
```

```
## data:  winecor$free.sulfur.dioxide and winecor$quality
## Kruskal-Wallis chi-squared = 115.07, df = 6, p-value < 2.2e-16

kruskal.test(winecor$total.sulfur.dioxide, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$total.sulfur.dioxide and winecor$quality
## Kruskal-Wallis chi-squared = 266.67, df = 6, p-value < 2.2e-16

kruskal.test(winecor$density, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$density and winecor$quality
## Kruskal-Wallis chi-squared = 652.61, df = 6, p-value < 2.2e-16

kruskal.test(winecor$pH, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$pH and winecor$quality
## Kruskal-Wallis chi-squared = 65.473, df = 6, p-value = 3.454e-12

kruskal.test(winecor$sulphates, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$sulphates and winecor$quality
## Kruskal-Wallis chi-squared = 13.78, df = 6, p-value = 0.03219

kruskal.test(winecor$alcohol, winecor$quality)

##
##  Kruskal-Wallis rank sum test
##
## data:  winecor$alcohol and winecor$quality
## Kruskal-Wallis chi-squared = 1014.1, df = 6, p-value < 2.2e-16
```

Observamos que las variables que nos muestran diferencias más significativas, y que influyen más en la calidad del vino, son aquellas que hemos mencionado en el apartado anterior. Estas son volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density y alcohol, pudiendo mencionar igualmente el pH.

2.5.2 Modelo predictivo

Como objetivo final, nos interesa poder realizar predicciones sobre la calidad del vino, lo que nos permitirá crear mejores vinos en función de ciertos atributos. Para ello utilizaremos Random Forest, que es un algoritmo predictivo el cual combina diferentes arboles, donde cada árbol es construido con observaciones y variables aleatorias.

Nos apoyaremos en los datos extraídos en el apartado anterior, el cual nos indicaba las mejores variables para poder realizar dicha predicción.

Debido a esto, lo primero que haremos será definir la nueva estructura que vamos a utilizar.

```
subset <-

wine[,c("volatile.acidity","residual.sugar","chlorides","free.sulfur.dioxide","total.sulfur.dioxide","pH","density","alcohol","quality")]
```

Para poder extraer la información de forma más precisa, realizaremos una agrupación de los valores de calidad en “bueno”, “malo” y “normal”, discretizando la variable resultante.

```
subset$quality <- ifelse(subset$quality < 6, 'bad', ifelse(subset$quality == 6, 'normal', 'good'))

subset$quality <- as.factor(subset$quality)
```

Comenzamos seleccionando un subconjunto de datos aleatorio dentro de nuestra muestra, e indicamos que nos basaremos en la calidad para realizar nuestra predicción.

```
set.seed(666)
data_random <- subset[sample(nrow(subset)),]

set.seed(666)
y<-data_random[,9] # Calidad
X <- data_random[,1:8] # Resto de atributos
```

Preparamos un grupo de datos para entrenar nuestro modelo, que será de 2/3 de los datos totales, y los datos restantes los utilizaremos para comprobar si nuestro modelo es suficientemente bueno.

```
indexesRF = sample(1:nrow(subset), size=floor((2/3)*nrow(subset)))
trainXRF<-X[indexesRF,]
trainyRF<-y[indexesRF]
testXRF<-X[-indexesRF,]
testyRF<-y[-indexesRF]
```

Creamos el modelo utilizando el modelo Random Forest.

```
modelRF <- randomForest::randomForest(trainXRF,trainyRF)
modelRF

##
## Call:
##  randomForest(x = trainXRF, y = trainyRF)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 30.35%
## Confusion matrix:
##        bad good normal class.error
## bad    781   17    298   0.2874088
## good    12  423    285   0.4125000
## normal 232  147   1070   0.2615597
```

Ahora realizaremos la predicción utilizando el modelo de aprendizaje creado en el pun to anterior, e introducimos los valores de test para ver qué clasificación nos haría.

```
predicted_modelRF <- predict( modelRF, testXRF, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_mode
lRF == testyRF) / length(predicted_modelRF)))

## [1] "La precisión del árbol es: 70.8512 %"
```

Cuando no hay más de veinte clases, el rendimiento en el subconjunto de entrenamiento se analiza mediante una matriz de confusión que identifica los tipos de errores cometidos. La matriz de confusión es la medida típica para expresar la calidad de las clases obtenidas con un modelo.

```
mat_confRF<-table(testyRF,Predicted=predicted_modelRF)
mat_confRF
```

```
##          Predicted
## testyRF  bad good normal
##   bad    372    7    165
##   good     8  225    107
##   normal 116   73    560
```

Por último comprobamos el porcentaje de registros clasificados correctamente.

```
porcentaje_correctRF<-100 * sum(diag(mat_confRF)) / sum(mat_confRF)
print(sprintf("El %% de registros correctamente clasificados es: %.4f %%"
,porcentaje_correctRF))
```

```
## [1] "El % de registros correctamente clasificados es: 70.8512 %"
```

2.6 Conclusiones

Mediante diversas pruebas estadísticas, hemos intentado cumplir con el objetivo que hemos planteado al comienzo del estudio. Se han analizado los datos proporcionados para intentar obtener el conocimiento necesario aportado por ellos y mostrar, mediante distintas herramientas, la información que finalmente nos ayudase a plantear un algoritmo predictivo sobre la calidad de esta variedad de vino.

Nuestro estudio ha comenzado analizando los datos con los que contamos, comprobando si dentro de ellos encontramos valores vacíos y/o valores atípicos. En el primer caso se ha verificado que todos los datos están completos, por lo que en este sentido no se ha tenido que realizar ningún tipo de modificación. En cuanto a los valores atípicos, sí que se ha comprobado que existen diversos datos de los considerados extremos. Debido a que, como se ha mencionado durante el estudio, estos datos provienen de pruebas objetivas que marcarán la calidad final del vino, se ha optado por mantenerlos, ya que consideramos que todos ellos son los que nos aportarán la información necesaria para poder establecer los atributos más influyentes en la elaboración de un vino de una calidad determinada.

Gracias al análisis de la correlación y del contraste de hipótesis hemos podido extraer cuáles son los atributos más determinantes e influyentes para poder elaborar un vino con una calidad específica. Y, mediante la utilización de dichos atributos, hemos podido generar un modelo predictivo utilizando el algoritmo Random Forest que nos va a permitir obtener, previo a la propia elaboración del vino, la calidad que obtendríamos combinando distintas cantidades de cada uno de ellos.

Con todo ello, nos vemos capacitados para poder elaborar vino de más alta calidad centrándonos en los valores de unos atributos específicos que hemos podido localizar gracias al estudio que hemos llevado a cabo.