PRÁCTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS

Marta Martin Llambes

29 de diciembre, 2018

Contents

1. Descripción del dataset

Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

Para la realización de esta práctica se ha decidido utilizar el dataset de kaggle "Red Wine Quality". Los datos originales se han obtenido desde el siguiente enlace: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009.

Los datos que se encuentran en el fichero contienen información sobre un total de 1599 vinos rojos portugueses de la marca "Vinho Verde". Varias cualidades de estos vinos fueron analizadas en un laboratorio y se adjuntaron en este dataset un total de 11 variables que representan propiedades químicas, y una variable que contiene una puntuación que le fué asignada a cada vino como resultado de ser probado por un mínimo de tres expertos.

Las variables que contiene el dataset son las siguientes:

- *fixed.acidity*: cantidad de ácidos no volátiles presente en el vino.
- *volatile.acidity*: cantidad de ácido acético presente en el vino.
- *citric.acid*: cantidad de ácido cítrico presente en el vino.
- *residual.sugar*: cantidad de azúcar presente en el vino después de que acabe la fermentación.
- *chlorides*: cantidad de sal presente en el vino.
- *free.sulfur.dioxide*: cantidad de dióxido de azufre libre presente en el vino.
- *total.sulfur.dioxide*: cantidad total de dióxido de azufre presente en el vino.
- *density*: valor de la medida de densidad del vino.
- *pH*: valor de la escala de pH que tiene el vino.
- *sulphates*: cantidad de sulfatos que contiene el vino, un aditivo.
- *alcohol*: porcentaje de alcohol que contiene el vino.
- *quality*: puntuación otorgada al vino por parte de catadores expertos.

Los datos contenidos en este dataset son importantes porque nos permitirán estudiar qué variables químicas de las anteriores pueden ser más influyentes en la puntuación de un vino. Con esta información se podría predecir para los nuevos vinos, aproximadamente y de manera rápida, qué puntuación se les otorgaría, si serían considerados buenos vinos o no, más información objetiva para estratificar los vinos en gamas, conocer con datos adicionales (como, por ejemplo, de ventas) qué propiedades químicas de los vinos se prefieren en distintas regiones del mundo, con el tiempo estudiar más profundamente qué factores externos afectan y cómo a que las variables más correlacionadas con la calidad del vino obtengan valores más óptimos, etc.

En nuestro caso, en esta práctica nos interesa averiguar qué variables influyen más en la puntuación de un vino, e intentar encontrar un modelo que ayude a predecir la puntuación de un vino a partir de dichas variables. También realizaremos alguna prueba de hipótesis para comprobar que la cantidad presente de alguna de las variables más influyentes en la calidad del vino, realmente hace que un vino se encuentre en el nivel de una gama mejor o no.

2. Integración y selección de los datos a analizar

En primer lugar, procederemos con la importación de los datos del fichero winequality-red.csv mediante la función read.csv.

```
#Carga de los datos del fichero csv.
wine <- read.csv("winequality-red.csv", sep=",", na.strings = "NA")

#Comprobamos que los datos se han cargado correctamente con las funciones head() y tail().
head(wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            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             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

```
tail(wine)
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1594           6.8             0.620        0.08            1.9     0.068
## 1595           6.2             0.600        0.08            2.0     0.090
## 1596           5.9             0.550        0.10            2.2     0.062
## 1597           6.3             0.510        0.13            2.3     0.076
## 1598           5.9             0.645        0.12            2.0     0.075
## 1599           6.0             0.310        0.47            3.6     0.067
```

```
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates
## 1594                  28                   38 0.99651 3.42      0.82
## 1595                  32                   44 0.99490 3.45      0.58
## 1596                  39                   51 0.99512 3.52      0.76
## 1597                  29                   40 0.99574 3.42      0.75
## 1598                  32                   44 0.99547 3.57      0.71
## 1599                  18                   42 0.99549 3.39      0.66
##      alcohol quality
## 1594     9.5       6
## 1595    10.5       5
## 1596    11.2       6
## 1597    11.0       6
## 1598    10.2       5
## 1599    11.0       6
```

```
#Mostramos un pequeño resumen de los datos.
summary(wine)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00
##  Median :0.07900   Median :14.00       Median : 38.00
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.636
##  3rd Qu.:6.000
##  Max.   :8.000
```

Como hemos podido comprobar, los datos se han cargado correctamente, ya que disponemos de información sobre 1599 vinos y de sus 12 variables.

Podemos observar que todas las variables son propiedades químicas de los vinos, a excepción de la última ("quality") que hace referencia a la nota que se le otorga al vino. Por lo tanto, de momento nos interesa conservar todos los datos para nuestros futuros análisis, ya que queremos analizar cuáles de todas las variables son las más influyentes en la puntuación de la calidad.

A continuación, comprobaremos que el tipo de dato asignado a cada variable sea el correcto.

```r
#Comprobación del tipo de dato de cada variable.
sapply(wine, function(x) class(x))
```

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

Tal y como podemos observar, todas las variabes son interpretadas correctamente como tipo numérico o entero, así que no necesitaremos corregir ninguna de ellas.

3. Limpieza de los datos

3.1. Identificación y tratamiento de ceros y valores vacíos

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Vamos a comprobar si los datos contienen valores desconocidos que hayan sido catalogados como "NA" mediante la función "is.na()".

```r
#Recuento de valores NA para cada variable.
sapply(wine, function(x) sum(is.na(x)))
```

```
##        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
```

Observamos que nuestros datos no contienen valores catalogados como NA. A continuación, comprobaremos si los datos contienen ceros que pudieran aparecer debido a valores desconocidos.

```r
#Recuento de ceros para cada variable.
sapply(wine, function(x) sum(x == 0))
```

```
##        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
```

Solo la variable "citric.acid" contiene un total de 132 ceros. Tal y como se menciona en el artículo "Modeling wine preferences by data mining from physicochemical properties" de P. Cortez et al., las muestras de los vinos fueron testeados y examinados por una entidad de certificación oficial llamada CVRVV, así que asumiremos que no todos los vinos contienen ácido cítrico, que los valores igual a cero de la variable ácido cítrico son posibles y correctos.

Por último comprobaremos si nuestros datos contienen valores vacíos:

```r
#Recuento de valores vacíos para cada variable.
sapply(wine, function(x) sum(x == ""))
```

```
##        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
```

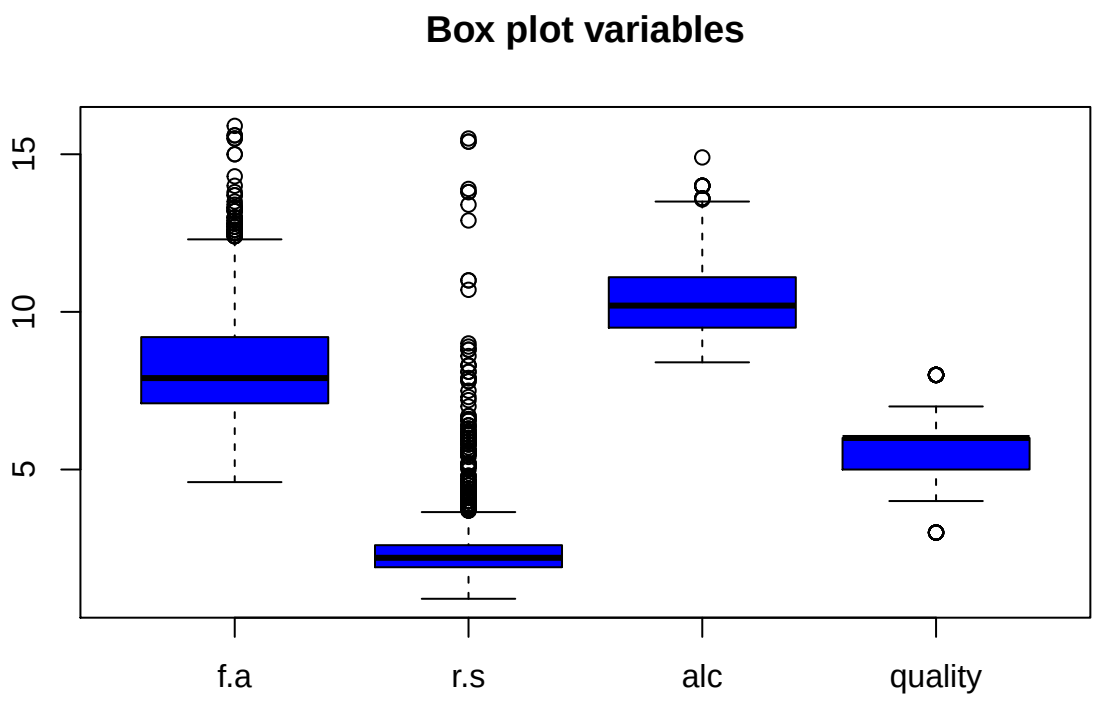
Como hemos observado que nuestros datos no contienen valores vacíos, ni ceros, ni NA's, no tendremos que afrontar la situación donde se debería decidir como lidiar con ellos: eliminando los registros, utilizando técnicas que imputen un valor estimado aproximado (como por ejemplo, el método de los KNN), etc.

Los datos proporcionados se encuentran en un buen estado, parece que anteriormente ya han sido limpiados y preprocesados.

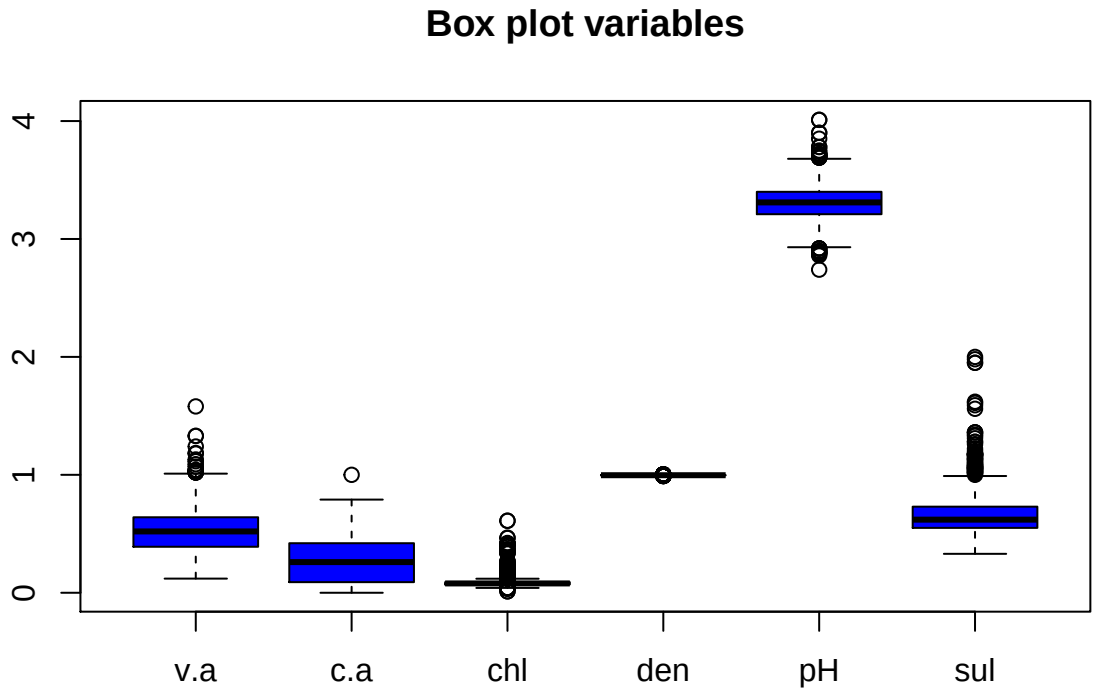
3.2. Identificación y tratamiento de valores extremos

A continuación, analizaremos si nuestros datos contienen valores extremos (outliers). Primero trataremos de identificarlos visualmente mediante diagramas de cajas:

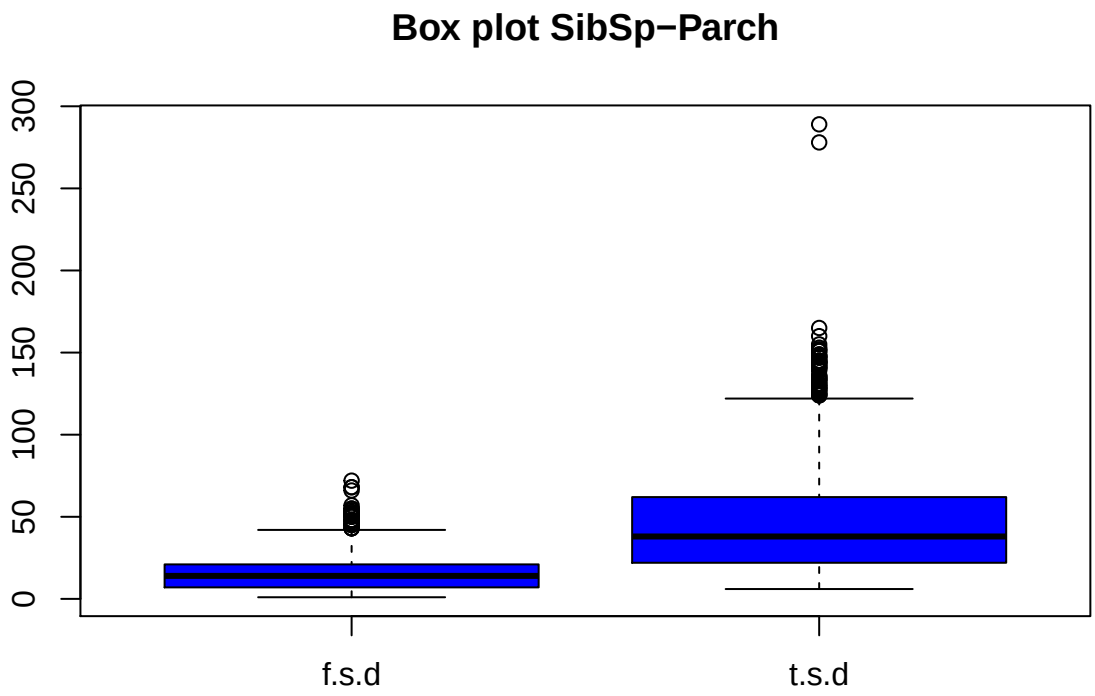
```r
#Creamos el diagrama de cajas de las variables cuantitativas.
boxplot(wine$fixed.acidity, wine$residual.sugar, wine$alcohol, wine$quality ,main="Box plot variables",
```



```
boxplot(wine$volatile.acidity, wine$citric.acid, wine$chlorides, wine$density, wine$pH, wine$sulphates,
```




```
boxplot(wine$free.sulfur.dioxide, wine$total.sulfur.dioxide, names=c("f.s.d","t.s.d"), main="Box plot S
```



Como podemos observar en los diagramas de cajas, todas las variables contienen outliers. Analizando los datos originales, estos parecen haber sido preprocesados, no faltan valores, y tal y como se menciona en el artículo "Modeling wine preferences by data mining from physicochemical properties" de P. Cortez et al., los datos fueron generados a partir de los análisis que llevó a cabo la entidad de certificación oficial llamada CVRVV. Todo esto nos hace pensar que los valores extremos registrados no son erróneos, y decidiremos no eliminarlos ya que se podría perder información valiosa para los futuros análisis que llevaremos a cabo.

4. Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar

Planificación de los análisis a aplicar.

Primero de todo, analizaremos la variable "quality" para hacernos una idea de la cantidad de vinos que hay de cada puntuación.

```
#Número de vinos de cada puntuación.
table(wine$quality)
```

```
##
##   3   4   5   6   7   8
##  10  53 681 638 199  18
```

Para llevar a cabo los análisis de este apartado, crearemos una nueva variable ("classification") a partir de la variable "quality", la cual clasificará los vinos en buenos "B" (para vinos con una puntuación igual a 7 o superior), y en no buenos "NB" (para vinos con una puntuación inferior a 7). Dicha variable la utilizaremos para hacer comparaciones en los siguientes subapartados.

```
#Creación de la variable "classification".
wine$classification <- ifelse(wine$quality < 7, "NB", "B")
#Comprobamos el tipo de la nueva varible "classification".
class(wine$classification)
```

```
## [1] "character"
```

```
#Corregimos el tipo de la variable "classification".
wine$classification <- as.factor(wine$classification)
#Comprobamos que el tipo de la variable "classification" se ha corregido.
class(wine$classification)
```

```
## [1] "factor"
```

Comprobamos que la clasificación se ha realizado correctamente:

```
#Número de vinos buenos y no buenos.
table(wine$classification)
```

```
##
##    B   NB
##  217 1382
```

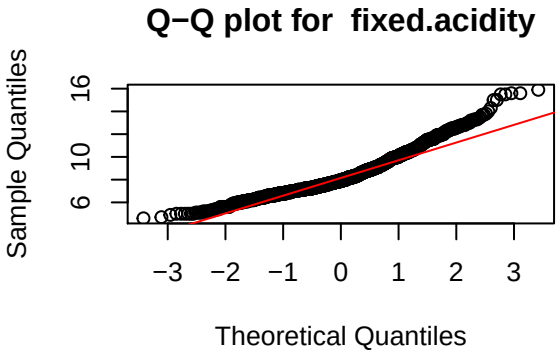
Con los datos ya preparados podemos proceder a realizar los análisis estadísticos que habíamos planteado en el inicio de la práctica. En primer lugar haremos una matriz de correlaciones para examinar qué variables son las más influyentes en la determinación de la puntuación de la variable "quality". Con la información obtenida de la matriz de correlaciones, intentaremos crear un modelo de regresión lineal que explique y permita predecir la puntuación de los vinos en función de las variables más influyentes en dicha puntuación. Ya por último, haremos una prueba de hipótesis para comprobar si los vinos buenos tienen normalmente mayor presencia de una variable química que escojamos, en comparación con los vinos clasificados como no buenos.

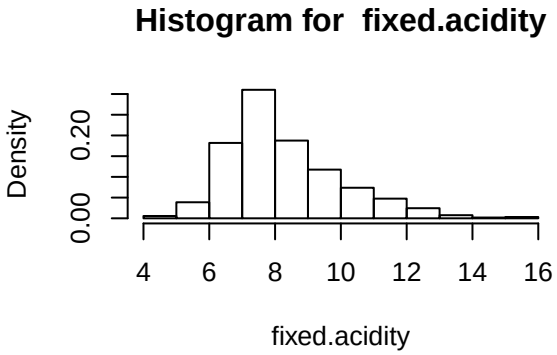
4.2. Comprobación de la normalidad y homogeneidad de la varianza

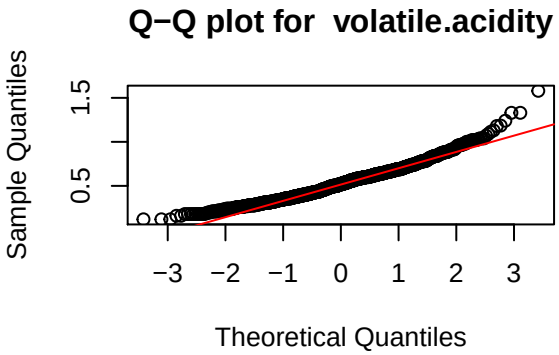
Para comprobar la normalidad o no de las variables cuantitativas de nuestra muestra se pueden llevar a cabo tests de normalidad, o se puede mirar de sacar conclusiones a partir de representaciones gráficas. Por ejemplo, si queremos hacer un análisis de manera visual sobre gráficas, podemos usar histogramas o gráficas de normalidad (Q-Q plot).

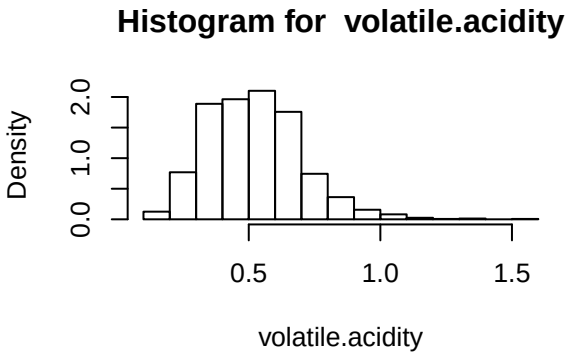
A continuación, por ejemplo, generaremos las gráficas "Q-Q plot" para inspeccionar de manera visual si nuestras variables siguen una distribución normal:

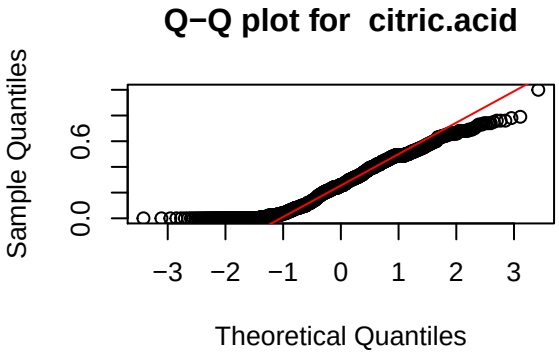
```
par(mfrow=c(2,2))
for(i in 1:ncol(wine)){
  if(is.numeric(wine[,i])){
    qqnorm(wine[,i], main = paste("Q-Q plot for ", colnames(wine)[i]))
    qqline(wine[,i], col="red")
    hist(wine[,i], main=paste("Histogram for ", colnames(wine)[i]), xlab = colnames(wine)[i], freq = FA
  }
}
```

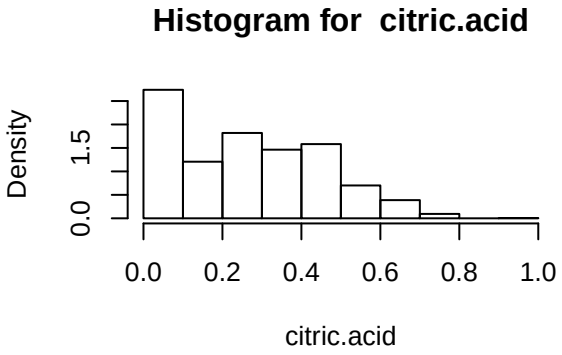

Q−Q plot for fixed.acidity
Sample Quantiles
6
10
16
−3
−2
−1
0
1
2
3
Theoretical Quantiles

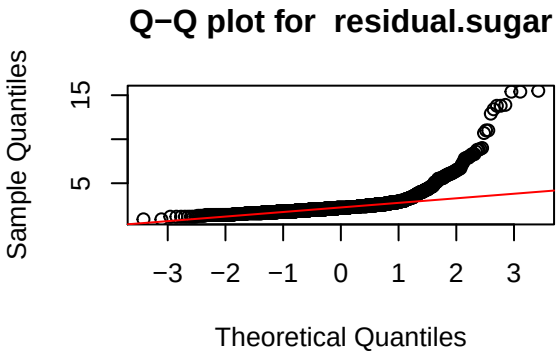

Histogram for fixed.acidity
Density
0.00
0.20
4
6
8
10
12
14
16
fixed.acidity

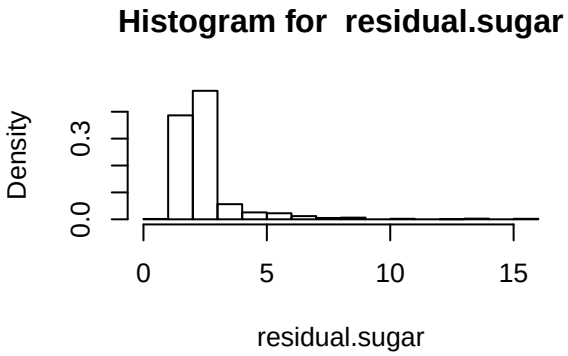

Q−Q plot for volatile.acidity
Sample Quantiles
0.5
1.5
−3
−2
−1
0
1
2
3
Theoretical Quantiles

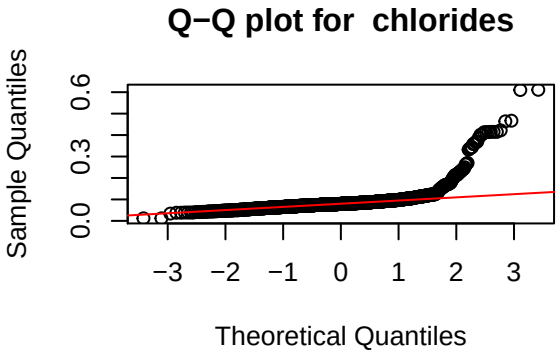

Histogram for volatile.acidity
Density
0.0
1.0
2.0
0.5
1.0
1.5
volatile.acidity

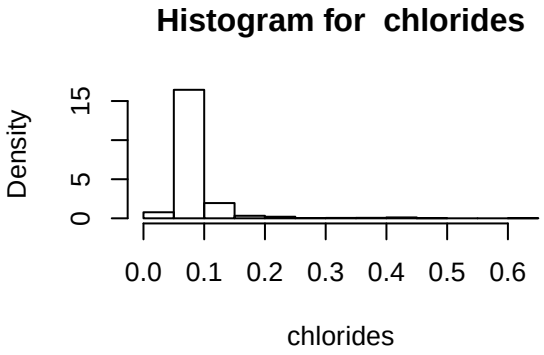

Q−Q plot for citric.acid
Sample Quantiles
0.0
0.6
−3
−2
−1
0
1
2
3
Theoretical Quantiles

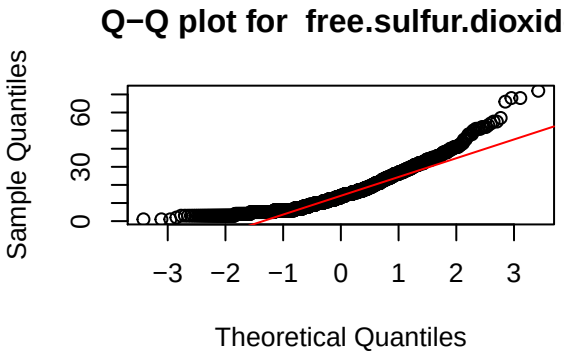

Histogram for citric.acid
Density
0.0
1.5
0.0
0.2
0.4
0.6
0.8
1.0
citric.acid

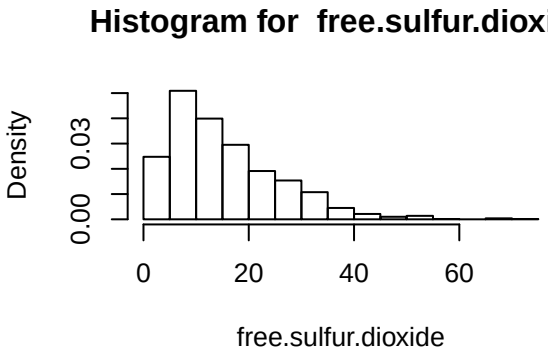

Q−Q plot for residual.sugar
Sample Quantiles
5
15
−3
−2
−1
0
1
2
3
Theoretical Quantiles

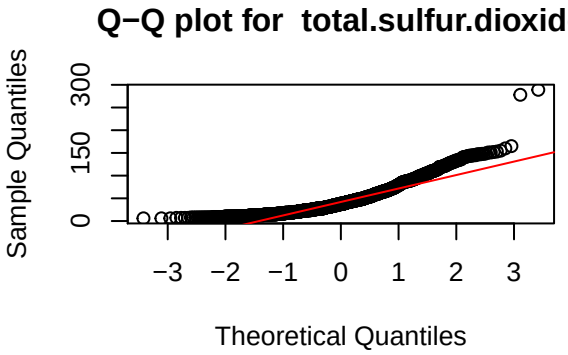

Histogram for residual.sugar
Density
0.0
0.3
0
5
10
15
residual.sugar

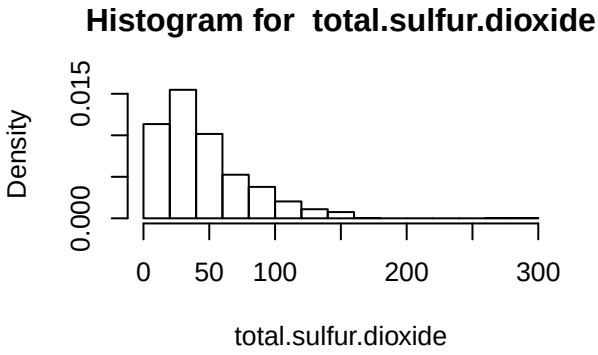

Q−Q plot for chlorides
Sample Quantiles
0.0
0.3
0.6
−3
−2
−1
0
1
2
3
Theoretical Quantiles

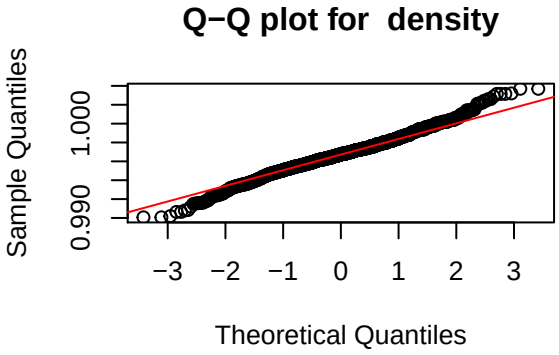

Histogram for chlorides
Density
0
5
15
0.0
0.1
0.2
0.3
0.4
0.5
0.6
chlorides

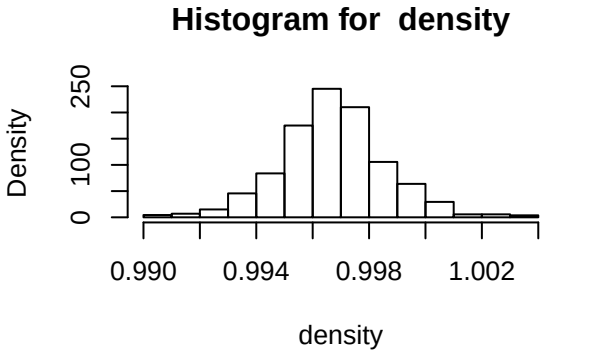

Q−Q plot for free.sulfur.dioxide
Sample Quantiles
0
30
60
−3
−2
−1
0
1
2
3
Theoretical Quantiles

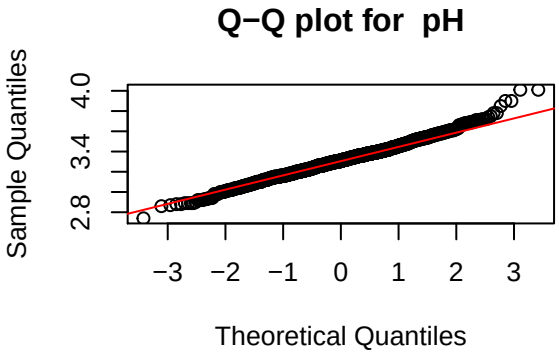

Histogram for free.sulfur.dioxide
Density
0.00
0.03
0
20
40
60
free.sulfur.dioxide

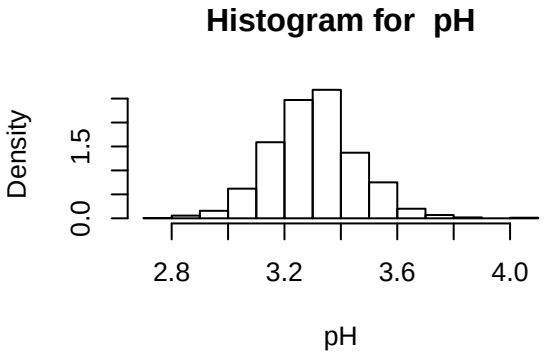

Q−Q plot for total.sulfur.dioxide
Sample Quantiles
0
150
300
−3
−2
−1
0
1
2
3
Theoretical Quantiles

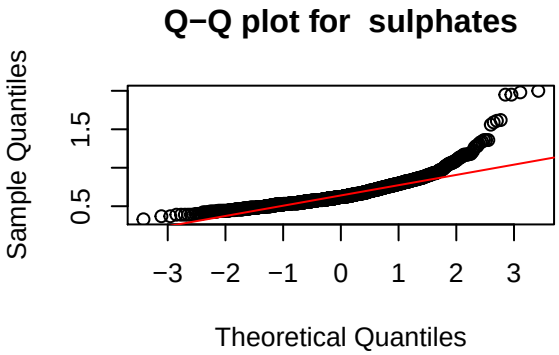

Histogram for total.sulfur.dioxide
Density
0.000
0.015
0
50
100
200
300
total.sulfur.dioxide

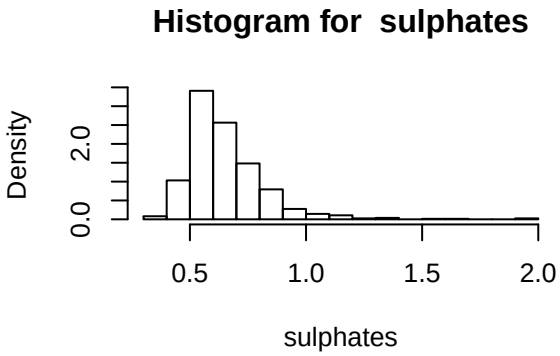

Q−Q plot for density
Sample Quantiles
0.990
1.000
−3
−2
−1
0
1
2
3
Theoretical Quantiles

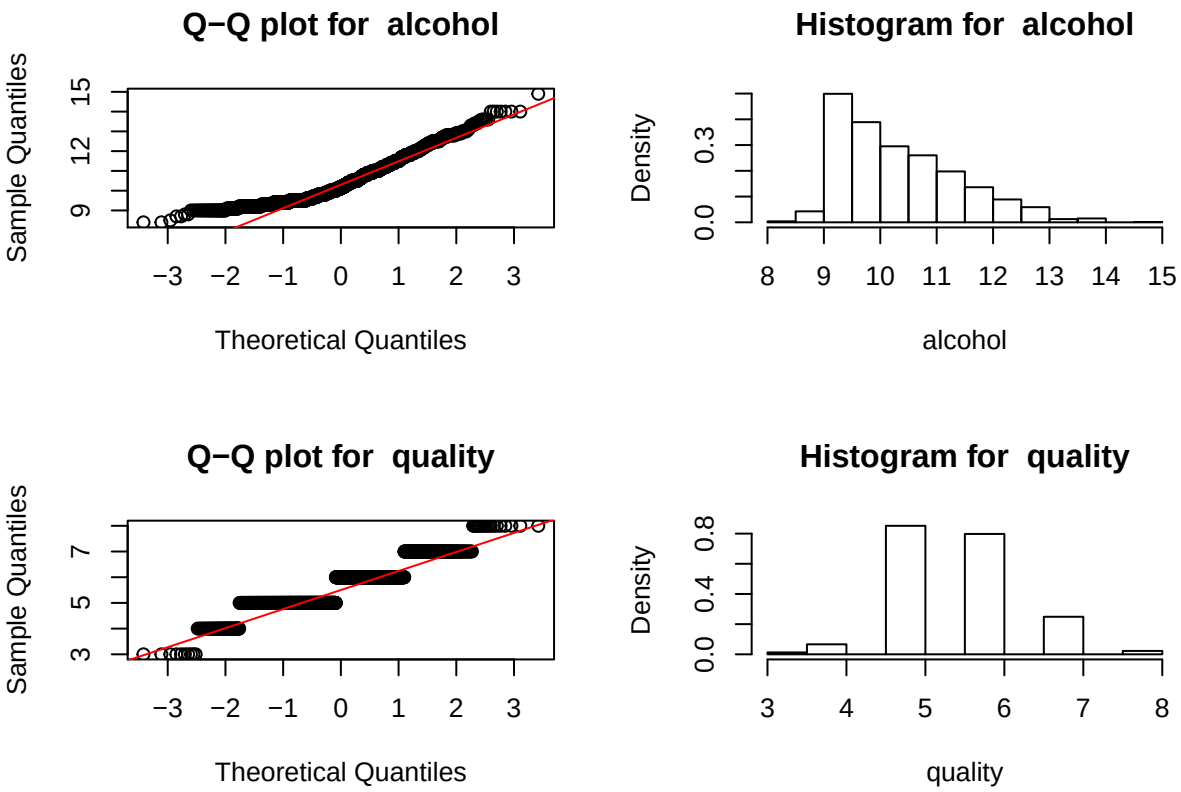

Histogram for density
Density
0
100
250
0.990
0.994
0.998
1.002
density


Q−Q plot for pH
Sample Quantiles
2.8
3.4
4.0
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for pH
Density
0.0
1.5
2.8
3.2
3.6
4.0
pH


Q−Q plot for sulphates
Sample Quantiles
0.5
1.5
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for sulphates
Density
0.0
2.0
0.5
1.0
1.5
2.0
sulphates



De los resultados obtenidos vemos que la mayoría de variables no se ajustan a la línea teória de la gráfica Q-Q, las que más se acercarían quizás son "ph" y "density", pero comprobaremos mediante un test de normalidad si se puede considerar que alguna de las variables sigue una distribución normal.

Como contamos con una muestra grande (n > 50), no se aconseja realizar el test de Shapiro-Wilk, y entonces nos decantaremos por usar el test de Lilliefors (prueba de Kolmogorov-Smirnov con la corrección de Lilliefors).

En primer lugar debemos establecer las dos hipótesis que barajaremos:

- Hipótesis nula H0: La distribución es normal.
- Hipótesis alternativa H1: La distribución no es normal.

A continuación, utilizaremos la función "lillie.test" del paquete "nortest" para llevar a cabo el test de Lilliefors en nuestras variables cuantitativas, considerando para la prueba un nivel de significación del 0,05:

```
#Test Lilliefors para las variables cuantitativas.
lillie.test(x = wine$fixed.acidity)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$fixed.acidity
## D = 0.1105, p-value < 2.2e-16
```

```
lillie.test(x = wine$volatile.acidity)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
```

```
## data:  wine$volatile.acidity
## D = 0.054662, p-value = 4.489e-12
```

```
lillie.test(x = wine$citric.acid)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$citric.acid
## D = 0.083866, p-value < 2.2e-16
```

```
lillie.test(x = wine$residual.sugar)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$residual.sugar
## D = 0.26068, p-value < 2.2e-16
```

```
lillie.test(x = wine$chlorides)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$chlorides
## D = 0.25964, p-value < 2.2e-16
```

```
lillie.test(x = wine$free.sulfur.dioxide)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$free.sulfur.dioxide
## D = 0.11124, p-value < 2.2e-16
```

```
lillie.test(x = wine$total.sulfur.dioxide)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$total.sulfur.dioxide
## D = 0.12098, p-value < 2.2e-16
```

```
lillie.test(x = wine$density)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$density
## D = 0.044787, p-value = 6.252e-08
```

```
lillie.test(x = wine$pH)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$pH
## D = 0.040368, p-value = 2.244e-06
```

```
lillie.test(x = wine$sulphates)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$sulphates
## D = 0.12479, p-value < 2.2e-16
```

```
lillie.test(x = wine$alcohol)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$alcohol
## D = 0.12145, p-value < 2.2e-16
```

```
lillie.test(x = wine$quality)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  wine$quality
## D = 0.24982, p-value < 2.2e-16
```

Como el p-valor obtenido para cada test es inferior al nivel de significación de 0,05 tenemos que rechazar la hipótesis nula para todos los casos, por lo tanto ninguna variable seguiría una distribución normal. Pero por el teorema del límite central, al tener una muestra grande (n > 30), podemos aproximar que las variables de la muestra siguen una distribución normal.

En la segunada parte de este subapartado debemos comprobar la homogeneidad de varianzas. Para llevar a cabo esta comprobación, consideraremos los dos grupos de datos que hemos creado en el subapartado anterior (los que agrupan los vinos en buenos y no buenos).

Las hipótesis que barajamos en este caso son las siguientes:

- Hipótesis nula H0: Las varianzas de los dos grupos son homogéneas.
- Hipótesis alternativa H1: Las varianzas de los dos grupos son distintas.

A continuación, aplicaremos el test de Levene con un nivel de significación del 0,05 para comprobar qué hipótesis aceptamos/rechazamos. Utilizaremos la función "leveneTest" del paquete "car".

```
#Test de Levene para comprobar la homogeneidad de varianzas.
leveneTest(fixed.acidity ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  13.119 0.0003014 ***
##       1597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(volatile.acidity ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value   Pr(>F)
## group    1  12.954 0.000329 ***
##       1597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(citric.acid ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.0566 0.3042
##       1597
```

```r
leveneTest(residual.sugar ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  2.2933 0.1301
##       1597
```

```r
leveneTest(chlorides ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.8136 0.1783
##       1597
```

```r
leveneTest(free.sulfur.dioxide ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.7548 0.1855
##       1597
```

```r
leveneTest(total.sulfur.dioxide ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  15.202 0.0001006 ***
##       1597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(density ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  17.027 3.874e-05 ***
##       1597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(pH ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  0.0554 0.8139
##       1597
```

```r
leveneTest(sulphates ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  0.7069 0.4006
##       1597
```

```
leveneTest(alcohol ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.4506 0.2286
##       1597
```

```
leveneTest(quality ~ classification, wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  146.11 < 2.2e-16 ***
##       1597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analizando los resultados, en las pruebas donde la Pr(F) >= 0,05, significa que se acepta la hipótesis nula y que las varianzas de los dos grupos son homogéneas. Esto se cumple con los datos de los grupos de vinos buenos y no buenos cuando consideramos las variables citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, pH, sulphates y alcohol. Para el resto de variables, las varianzas de los dos grupos de vinos no se pueden considerar homogéneas.

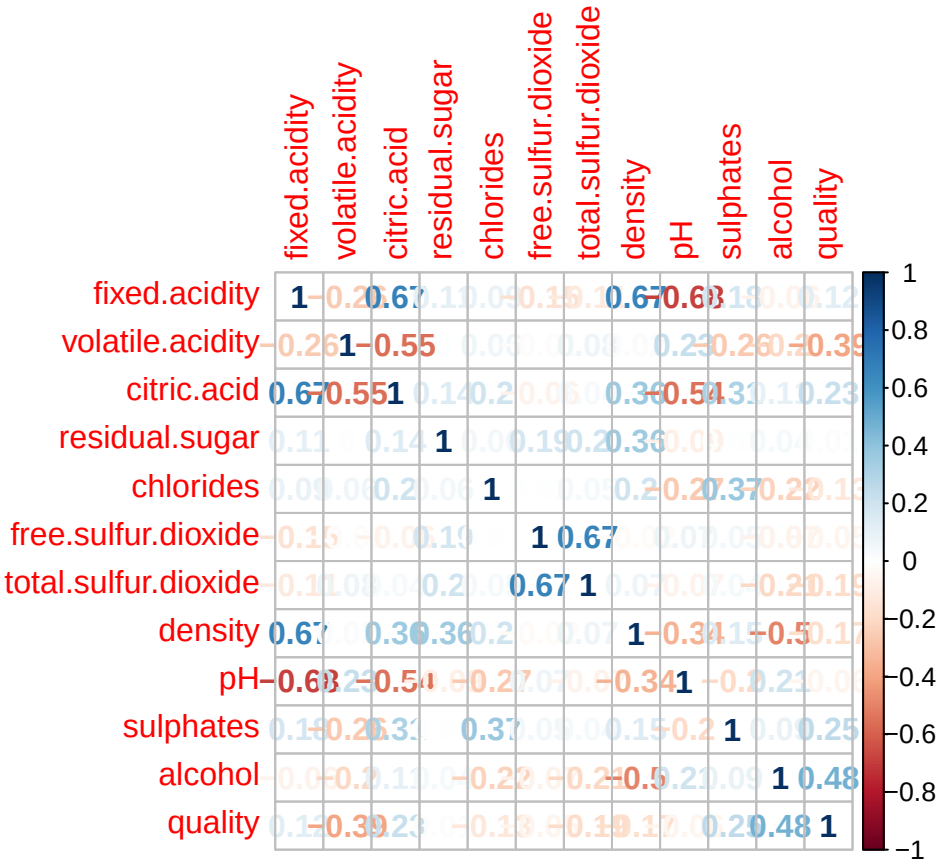
4.3. Aplicación de pruebas estadísticas para comprobar grupos de datos

En función de los datos y el objetivo de estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

4.3.1. Matriz de correlación

Con la función "corrplot" del paquete "corrplot" vamos a crear una representación gráfica de una matriz de correlación de todas las variables cuantitativas de nuestros datos. Los valores representados en azul corresponden a los coeficientes de correlación positivos, mientras que los valores representados en rojo corresponden a los coeficientes de correlación negativos. Los resultados nos permitirán ver qué variables influyen más/menos en otras variables, como, por ejemplo, qué variables determinan más el valor de la variable "quality".

```
par(mfrow = c(1,1))
cor.wine <- cor(wine[,1:12])
corrplot(cor.wine, method = 'number')
```



Del resultado podemos extraer las siguientes conclusiones:

- Las variables que se encuentran más correlacionadas con "quality" son por orden (aunque no presentan valores muy elevados): "alcohol", "volatile.acidity", "sulphates", "citric.acid", "total.sulfur.dioxide", "density", "chlorides", y "fixed.acidity".
- La variable "fixed.acidity" se encuentra fuertemente correlacionada con "citric.acid", "density" y "pH".
- La variable "total.sulfur.dioxide" se encuentra fuertemente correlacionada (como ya era de esperar) con la variable "free.sulfur.dioxide".
- La variable "density", además de estar fuertemente correlacionada con "fixed.acidity", también lo está con "alcohol".
- La variable "pH", además de estar fuertemente correlacionada con "fixed.acidity", también lo está con "citric.acid" y "density".

4.3.2. Modelo de regresión lineal múltiple (regresores cuantitativos)

En este subapartado, vamos a intentar encontrar un modelo de regresión que explique como viene determinada la variable "quality".

Para llevar a cabo la estimación de un modelo lineal por mínimos cuadrados, utilizaremos la fnción "lm()". Definiremos la variable "quality" como variable dependiente, y las otras como independientes. En primer lugar crearemos un modelo con las dos variables que en el subapartado anterior hemos visto que tenían una mayor correlación con nuestra variable dependiente: "alcohol" y "volatile.acidity". A continuación, iremos creando más modelos incorporando más variables para comprobar si el modelo se ajusta mejor a los datos.

```
#Estimamos el modelo lineal por mínimos cuadrados con la función lm().
model1 <- lm(quality ~ alcohol + volatile.acidity, data=wine)
```

```r
#Mostramos por pantalla las propiedades del modelo obtenido.
summary(model1)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59342 -0.40416 -0.07426  0.46539  2.25809
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.09547    0.18450   16.78   <2e-16 ***
## alcohol           0.31381    0.01601   19.60   <2e-16 ***
## volatile.acidity -1.38364    0.09527  -14.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6678 on 1596 degrees of freedom
## Multiple R-squared:  0.317,  Adjusted R-squared:  0.3161
## F-statistic: 370.4 on 2 and 1596 DF,  p-value: < 2.2e-16
```

El modelo de regresión lineal obtenido es el siguiente:

$$quality = 3.09547 + 0.31381 alcohol_i - 1.38364 volatile.acidity_i$$

Como podemos observar en los resultados del modelo, el coeficiente de determinación o de correlación múltiple de este modelo tiene un valor de 0.317. Este valor nos indica que la bondad del ajuste no es muy buena, ya que el valor de este coeficiente siempre se encuentra entre 0 y 1, y cuánto mayor es, mayor es la bondad del ajuste.

Vamos a añadir más variables al modelo a ver si aumenta la bondad del ajuste, y representaremos todos los resultados en una tabla con la función "mtable" del paquete "memisc":

```r
#Creación de modelos lineales incluyendo más variables.
model2 <- update(model1, ~ . + sulphates)
model3 <- update(model2, ~ . + citric.acid)
model4 <- update(model3, ~ . + total.sulfur.dioxide)
model5 <- update(model4, ~ . + density)
model6 <- update(model5, ~ . + chlorides)
model7 <- update(model6, ~ . + fixed.acidity)

#Mostrar los resultados de todos los modelos en una tabla.
mtable(model1,model2,model3,model4,model5,model6,model7)
```

```
##
## Calls:
## model1: lm(formula = quality ~ alcohol + volatile.acidity, data = wine)
## model2: lm(formula = quality ~ alcohol + volatile.acidity + sulphates,
##     data = wine)
## model3: lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     citric.acid, data = wine)
## model4: lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     citric.acid + total.sulfur.dioxide, data = wine)
```

```
## model5: lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     citric.acid + total.sulfur.dioxide + density, data = wine)
## model6: lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     citric.acid + total.sulfur.dioxide + density + chlorides,
##     data = wine)
## model7: lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     citric.acid + total.sulfur.dioxide + density + chlorides +
##     fixed.acidity, data = wine)
##
## ============================================================================================
##                            model1        model2        model3        model4        model5        mod
## --------------------------------------------------------------------------------------------
##   (Intercept)               3.095***      2.611***      2.646***      2.843***     -7.009        -0.
##                            (0.184)       (0.196)       (0.201)       (0.205)      (11.972)      (11.
##   alcohol                   0.314***      0.309***      0.309***      0.295***      0.305***      0.
##                            (0.016)       (0.016)       (0.016)       (0.016)       (0.020)       (0.
##   volatile.acidity         -1.384***     -1.221***     -1.265***     -1.222***     -1.247***     -1.
##                            (0.095)       (0.097)       (0.113)       (0.112)       (0.116)       (0.
##   sulphates                               0.679***      0.696***      0.721***      0.710***      0.
##                                          (0.101)       (0.103)       (0.103)       (0.104)       (0.
##   citric.acid                                           -0.079        -0.043        -0.093        0.
##                                                         (0.104)       (0.104)       (0.120)       (0.
##   total.sulfur.dioxide                                                -0.002***     -0.002***     -0.
##                                                                       (0.001)       (0.001)       (0.
##   density                                                                            9.820         3.
##                                                                                     (11.931)      (11.
##   chlorides                                                                                        -1.
##                                                                                                    (0.
##   fixed.acidity
##
## --------------------------------------------------------------------------------------------
##   R-squared                 0.317         0.336         0.336         0.344         0.344         0.
##   adj. R-squared            0.316         0.335         0.334         0.342         0.342         0.
##   sigma                     0.668         0.659         0.659         0.655         0.655         0.
##   F                       370.379       268.912       201.777       166.962       139.219       123.
##   p                         0.000         0.000         0.000         0.000         0.000         0.
##   Log-likelihood        -1621.814     -1599.384     -1599.093     -1589.749     -1589.409     -1580.
##   Deviance                711.796       692.105       691.852       683.814       683.523       675.
##   AIC                    3251.628      3208.768      3210.186      3193.499      3194.818      3178.
##   BIC                    3273.136      3235.654      3242.448      3231.138      3237.835      3226.
##   N                      1599          1599          1599          1599          1599          1599
## ============================================================================================
```

Como resultado, observamos que añadiendo más variables al primer modelo que habíamos calculado tampoco se consigue mejorar mucho la bondad del ajuste, ya que el coeficiente de determinación o correlación prácticamente no aumenta. Parece ser que un modelo de regresión lineal no es la mejor opción para explicar los datos de nuestro dataset, por lo tanto, si lo usáramos para predecir el valor de la variable "quality" de nuevos vinos, nos daría un resultado muy poco preciso.

4.3.3. Contraste de hipótesis

En este subapartado vamos a comparar dos muestras: vinos buenos vs. vinos no buenos. Nos vamos a plantear como objetivo resolver la siguiente pregunta: ¿Podemos afirmar que el contenido de alcohol de los vinos no buenos ("NB") es inferior al de los vinos buenos ("B")?

Para este contraste de hipótesis vamos a considerar como hipótesis nula que la media del contenido de alcohol de ambos vinos es igual, y como hipótesis alternativa que la media del contenido de alcohol de los vinos no buenos es inferior a la de los buenos:

$$H_0 : \mu_{NB} = \mu_B H_1 : \mu_{NB} < \mu_B$$

Para llevar a cabo este contraste, asumiremos un nivel de significación del 0,05, que se trata de dos muestras independientes, que tienen distribución normal (por el teorema del límite central), y que se trata de un caso de varianzas poblacionales desconocidas iguales. Aplicaremos un contraste unilateral mediante la función "t.test":

```
#Prueba de hipótesis unilateral.
t.test( wine[wine$classification=="B",]$alcohol, wine[wine$classification=="NB",]$alcohol, alternative=
```

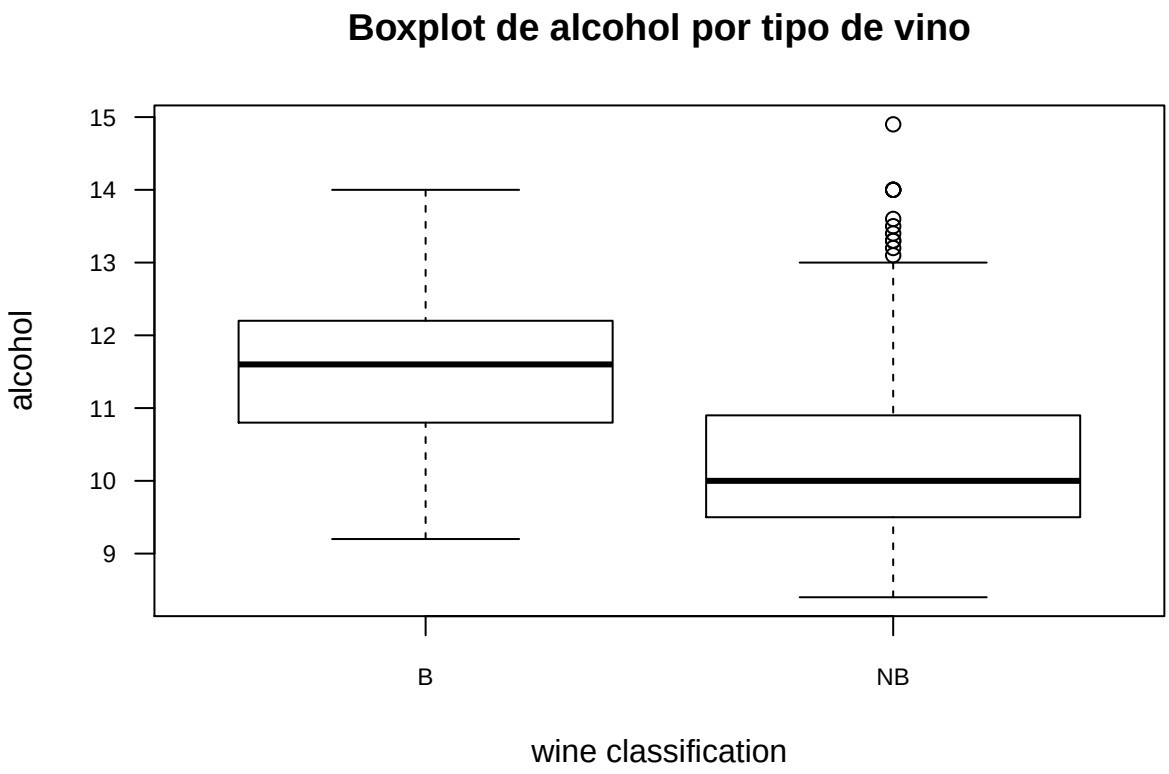
```
##
##  Two Sample t-test
##
## data:  wine[wine$classification == "B", ]$alcohol and wine[wine$classification == "NB", ]$alcohol
## t = 17.823, df = 1597, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.150012      Inf
## sample estimates:
## mean of x mean of y
##  11.51805  10.25104
```

Como el p-valor obtenido es menor que el nivel de significación (0,05) rechazamos la hipótesis nula y aceptamos la hipótesis alternativa, según la cual la media de alcohol de los vinos no buenos ("NB") es inferior a la media de alcohol de los vinos buenos ("B").

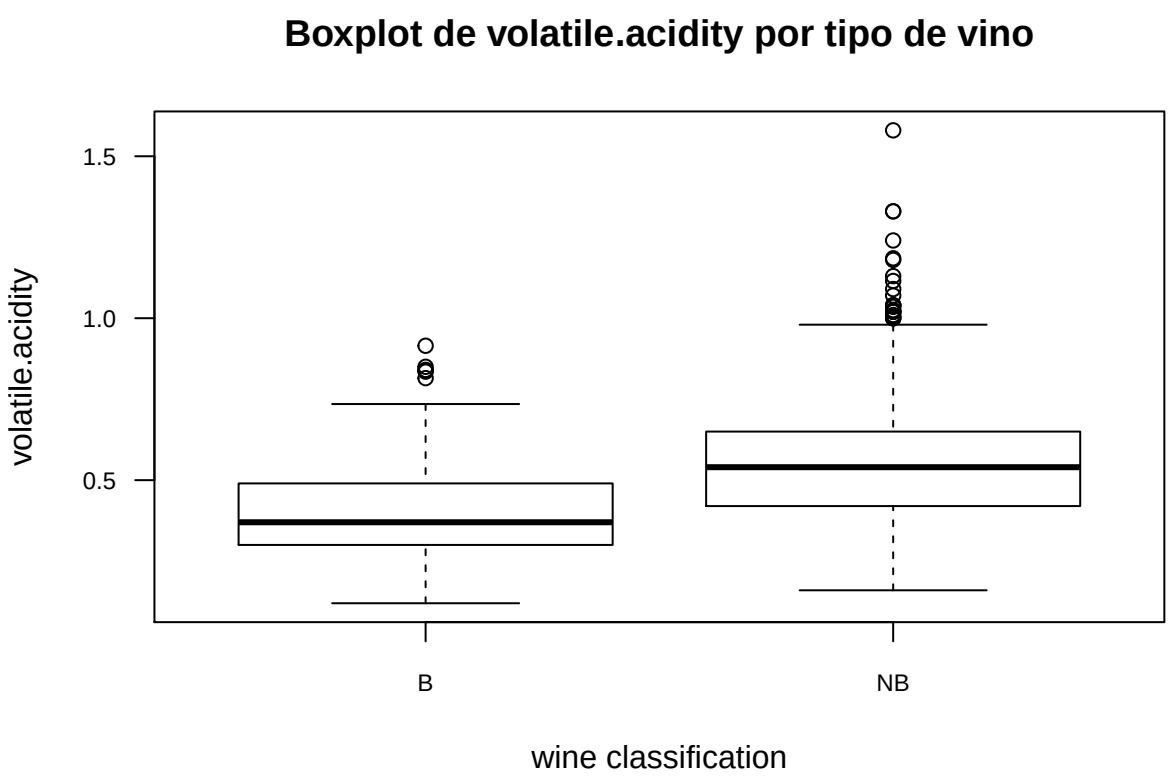
5. Representación de los resultados a partir de tablas y gráficas

Aparte de la representación gráfica de la matriz de correlaciones, y de la tabla de resultados de los modelos de regresión lineal adjuntados en el punto número 4 de esta práctica, para acabar incluiremos un último diagrama. Se trata de los diagramas de cajas de las dos variables más influyentes en la puntuación de la calidad del vino. Estos diagramas separan los rangos de valores que presentan los vinos buenos y no buenos en cuanto a "alcohol" y "volatile.acidity".

```
#Diagramas de caja de los valores de "alcohol" por tipo de vino.
boxplot(wine$alcohol~wine$classification, main="Boxplot de alcohol por tipo de vino",ylab="alcohol", xla
```




```
#Diagramas de caja de los valores de "volatile.acidity" por tipo de vino.
boxplot(wine$volatile.acidity~wine$classification, main="Boxplot de volatile.acidity por tipo de vino",
```



En los diagramas se puede observar claramente como la media de alcohol en vinos buenos es superior a la de los vino no buenos (salvo en pocas ocasiones), y como la media de acidez volátil de los vinos buenos suele ser inferior a la de los vinos no buenos.

6. Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Como conclusión, hemos aprendido que las variables que tienen mayor influencia positiva en la puntuación de la calidad del vino, por orden, son: "alcohol", "sulphates", "citric.acid", y "fixed.acidity". Por otro lado, las variables que tienen una mayor influencia negativa en la puntuación de la calidad del vino, por orden, son: "volatile.acidity", "total.sulfur.dioxide", "density", y "chlorides". Por lo tanto, los vinos con mayor cantidad de alcohol y menor cantidad de acidez volátil suelen ser clasificados como vinos buenos.

En cuanto a los resultados de los modelos de regresión lineal, no podemos aceptar ningún modelo como válido ya que la bondad de los ajustes no era muy buena, y se producirían resultados muy poco precisos si utilizaramos dichos modelos para classificar nuevos vinos en buenos y no buenos. Se deberían de usar otras técnicas para tratar de encontrar un modelo más preciso.

7. Código

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código R y las respuestas a las preguntas de la práctica se entregaran en un único archivo en versión pdf, html y Rmd. Dichos ficheros se pueden encontrar en https://github.com/mmartinlla/wine-cleaning/tree/

master/Code_and_Answers.

El dataset modificado lo generaremos de la siguiente manera:

```
#Generamos el nuevo archivo csv.
write.csv(wine, file = "../data/modified_wine.csv", row.names=F)
```

Dicho dataset se puede encontrar en el siguiente enlace: https://github.com/mmartinlla/wine-cleaning/tree/master/Data/Final.

8. Referencias

- P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. "Modeling wine preferences by data mining from physicochemical properties". In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
- López-Roldán, P.; Fachelli, S. (2016). "Análisis de varianza". En P. López-Roldán y S.Fachelli, "Metodología de la Investigación Social Cuantitativa". Bellaterra (Cerdanyola del Vallès): Dipòsit Digital de Documents, Universitat Autònoma de Barcelona. 1ª edición. Edición digital: http://ddd.uab.cat/record/163568
- Rovira Escofet, C., "Contraste de hipótesis", Apuntes de la UOC, P08/75057/02308
- Gibergans Bàguena, J., "Contraste de dos muestras", Apuntes de la UOC, P08/75057/02309
- Gibergans Bàguena, J., "Regresión lineal múltiple", Apuntes de la UOC, P08/75057/02312
- https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html
- http://vivaelsoftwarelibre.com/test-de-kolmogorov-smirnov-en-r/
- https://github.com/Bengis/nba-gap-cleaning
- Ejemplo de práctica proporcionado en la asignatura, de Teguayco Gutiérrez González.