Práctica 2: Limpieza y validación de los datos

ANTONIO SÁNCHEZ NAVARRO

Índice

1 Detalles de la actividad

1.1. Descripción

En esta práctica se elabora un caso práctico 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. Esta práctica se ha llevado a cabo en su totalidad de forma individual. Se ha hecho entrega de un único archivo con el enlace a Github (https://github.com) donde se encuentran las soluciones, así como el nombre del autor de esta práctica. Se ha utilizado la Wiki de Github para describir y estructurar los diferentes archivos que corresponden a esta entrega.

1.2. Objetivos

Los objetivos concretos de esta práctica son:

- 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 les permitan 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

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

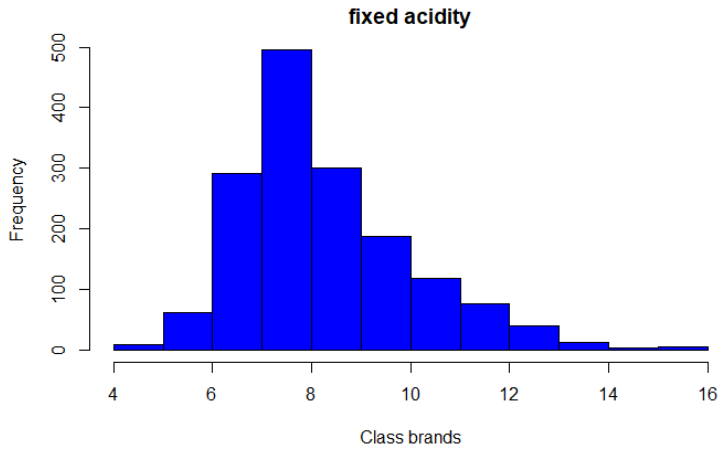
- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para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

2 Resolución

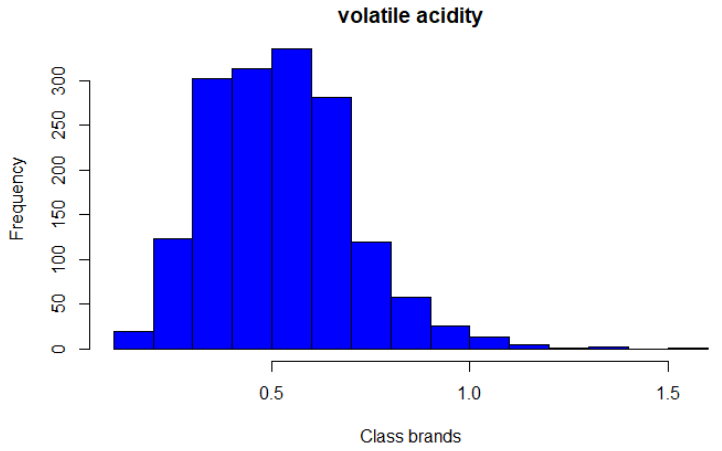
2.1. Descripción del dataset

El conjunto de datos de análisis se obtiene a partir del enlace https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 y está formado por 1599 registros y 12 variables fisicoquímicas. Estas variables son: *fixed.acidity*, *volatile.acidity*, *citric.acid*, *residual.sugar*, *chlorides*, *free.sulfur.dioxide*, *total.sulfur.dioxide*, *density*, *pH*, *sulphates*, *alcohol*, *quality*. Haremos una breve descripción de cada una:

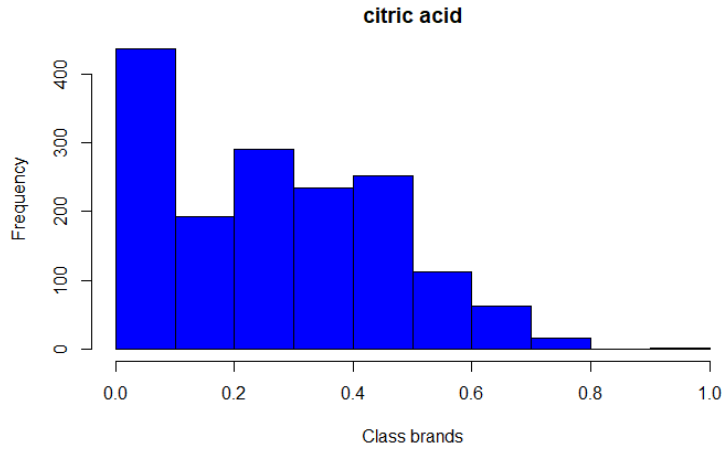
• ***fixed acidity***. La mayoría de los ácidos involucrados con el vino son fijos o no volátiles (no se evaporan fácilmente).



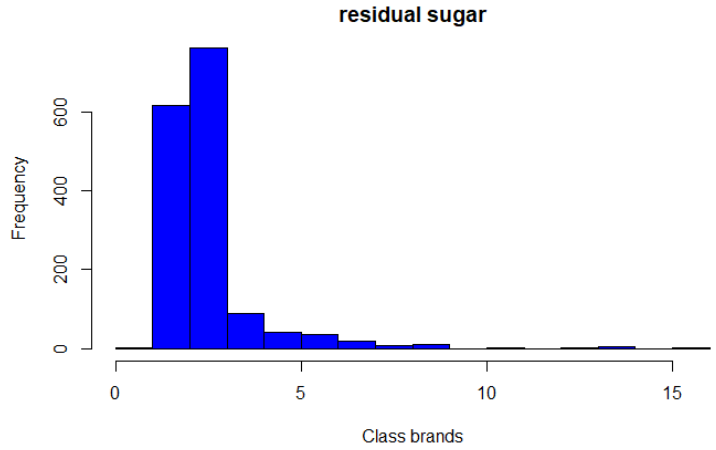
• ***volatile acidity***. Cantidad de ácido acético en el vino, que en niveles demasiado altos puede llevar a un sabor desagradable a vinagre.



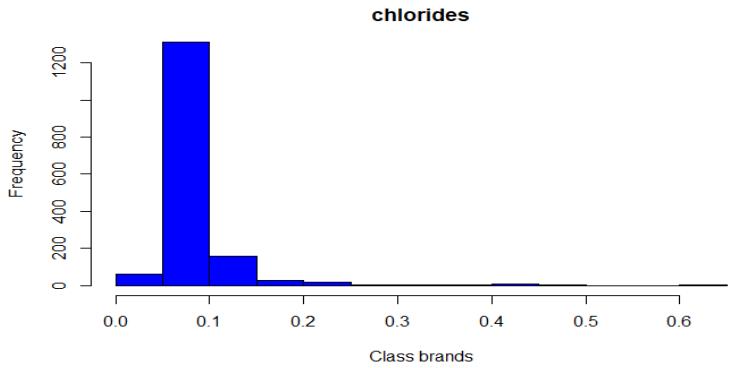
• ***citric acid***. En pequeñas cantidades, el ácido cítrico puede agregar “frescura” y sabor a los vinos.



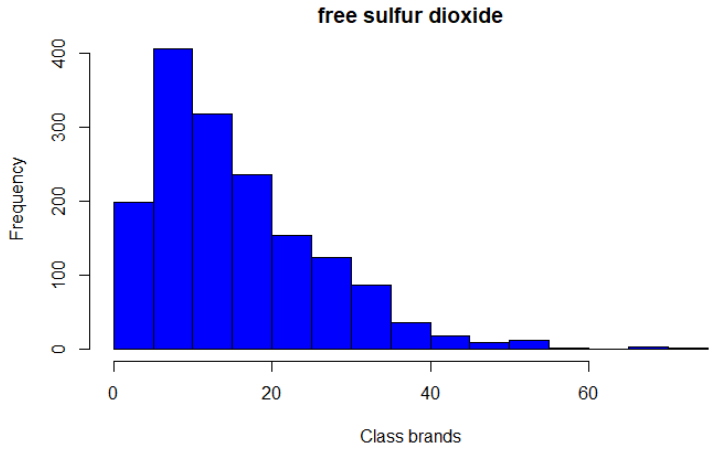
• ***residual sugar***. Cantidad de azúcar residual después de la fermentación. Es raro encontrar vinos con menos de 1 g/l y los vinos con más de 45 g/l se consideran dulces.



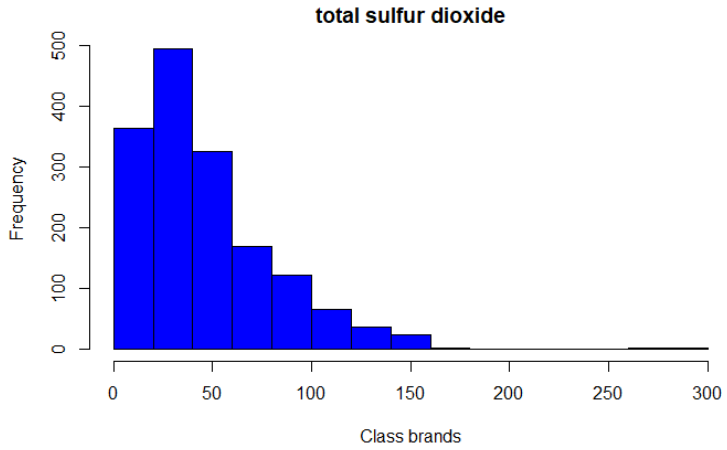
• ***chlorides***. Cantidad de sal en el vino.



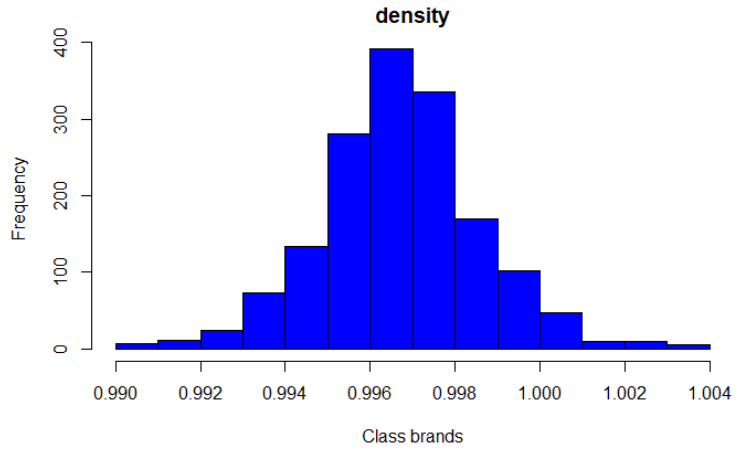
• ***free sulfur dioxide***. En estado natural, el SO_2 presenta un equilibrio entre el SO_2 molecular (como un gas disuelto) y el ion bisulfito. Previene el crecimiento microbiano y la oxidación del vino.



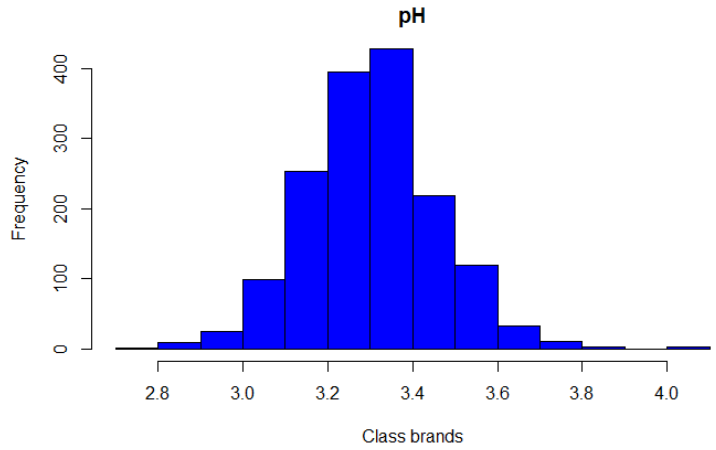
• ***total sulfur dioxide***. Cantidad de formas libres y ligadas de SO_2. En bajas concentraciones, el SO_2 es mayormente indetectable en el vino, pero a concentraciones de SO_2 libres superiores a 50 ppm, el SO_2 se hace evidente en el olfato y también en el sabor del vino.



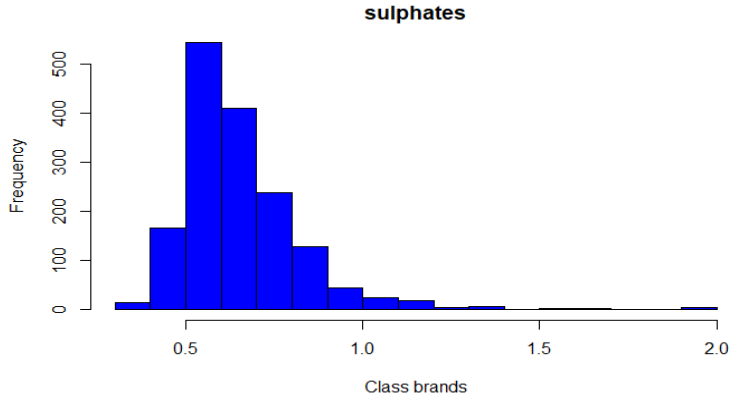
• ***density***. Densidad del agua según el porcentaje de alcohol y contenido en azúcar.



• ***pH***. Describe el grado de acidez o basicidad del vino en una escala de 0 (muy ácido) a 14 (muy básico). La mayoría de los vinos están entre 3 y 4 en la escala de pH.

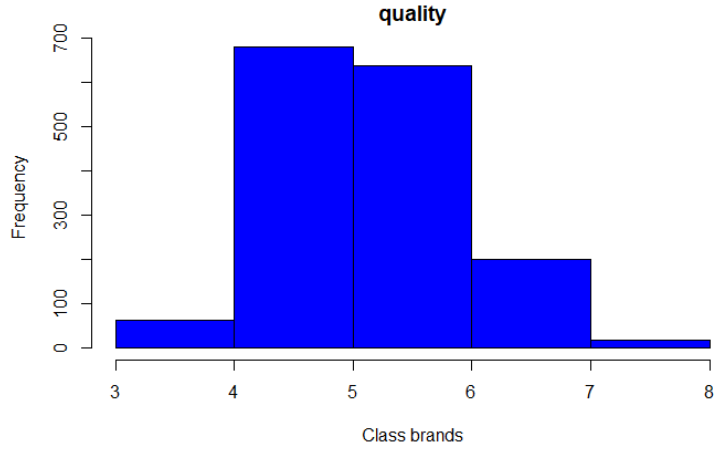


• ***sulphates***. Aditivo para vinos que puede contribuir a los niveles de gas de SO_2, que actúa como antimicrobiano y antioxidante.



• ***alcohol***. Porcentaje de alcohol en el vino.

• ***quality***. Variable de salida, basada en datos sensoriales, con una puntuación entre 0 y 10.



2.2. Importancia y objetivos de los análisis

A partir de este conjunto de datos , me planteo la problemática de determinar qué variables influyen más sobre la calidad de los vinos. Procederé además a crear modelos de regresión multilineales que permitan predecir la calidad de un vino en función de las características fisicoquímicas más influyentes, así como contrastes de hipótesis que contribuyan a identificar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

Estos análisis toman una especial relevancia en todo sector relacionado con la calidad y la enología. Son ejemplos de ello los servicios de peritaje internos en bodegas y laboratorios de investigación llevados a cabo por enólogos o químicos. En este sentido, el perito o responsable

de calidad podrá valerse de los análisis planteados en esta actividad y utilizarlos como soporte para sus tasaciones, validaciones y categorías o reconocimientos.

2.3. Limpieza de los datos

En primer lugar, se procede a la lectura del fichero CSV separado por comas *winequality-red.csv*. La llamada a la función *read.csv()* devuelve el siguiente conjunto de datos:

```
# Cargo el archivo de datos "winequality-red.csv" y valido que los tipos
# de datos se interpretan correctamente
winequality.red <- read.csv("C:/Users/tonis/Desktop/UOC/Tipologýa y ciclo de vida de los datos/PRAC2/winequality-red.csv", stringsAsFactors = FALSE, header = TRUE)
head(winequality.red[,1:5])
```

```
##   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
```

A continuación, examinamos el tipo de dato asignado a cada campo:

```
# Tipo de dato asignado a cada campo
sapply(winequality.red, 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"
```

Notamos que los tipos de datos asignados por R a las variables se corresponden con el dominio de estas. Salvo “quality”, que es de tipo *integer*, todas las variables son de tipo *numeric*.

2.3.1. Integración y selección de los datos de interés a analizar

Si bien todos los atributos presentes en el conjunto de datos se corresponden con características fisicoquímicas que reúnen los distintos vinos recogidas en forma de registros, considero que se puede prescindir en nuestro análisis de la columna ***free sulfur dioxide*** por encontrarse contenida en la columna ***total sulfur dioxide***. De esta manera, se eliminan redundancias y posibles problemas de multicolinealidad, debido a variables estrechamente correlacionadas, en análisis posteriores. Por otra parte, he considerado oportuno unir los dos tipos de acidez que pueden presentar los vinos (fijo y volátil) en uno solo, creando una nueva variable que sea la suma de ambos tipos y cuyo nombre sea ***acidity***. Este proceso se ha conseguido transformando y renombrando la columna ***fixed acidity*** y eliminando después la columna ***volatile acidity***.

```
# Eliminación de datos de columnas redundantes
winequality.red <- winequality.red[, -(6:6)]
# Unimos las dos columnas de acidez (fija y volátil) en una sola columna
winequality.red$fixed.acidity<-winequality.red$fixed.acidity + winequality.red$volatile.acidity
winequality.red$fixed.acidity<-round(winequality.red$fixed.acidity,2)
colnames(winequality.red)[colnames(winequality.red)=="fixed.acidity"] <- "acidity"
# Ahora que ya disponemos de la acidez total, eliminamos la columna "volatile.acidity":
winequality.red <- winequality.red[, -(2:2)]
head(winequality.red)
```

```
##   acidity citric.acid residual.sugar chlorides total.sulfur.dioxide
## 1    8.10        0.00            1.9     0.076                   34
## 2    8.68        0.00            2.6     0.098                   67
## 3    8.56        0.04            2.3     0.092                   54
## 4   11.48        0.56            1.9     0.075                   60
## 5    8.10        0.00            1.9     0.076                   34
## 6    8.06        0.00            1.8     0.075                   40
##   density   pH sulphates alcohol quality
## 1  0.9978 3.51      0.56     9.4       5
## 2  0.9968 3.20      0.68     9.8       5
## 3  0.9970 3.26      0.65     9.8       5
## 4  0.9980 3.16      0.58     9.8       6
## 5  0.9978 3.51      0.56     9.4       5
## 6  0.9978 3.51      0.56     9.4       5
```

Habiendo advertido que algunos valores de las columnas ***chlorides*** y ***density*** exceden el número de cifras decimales, establecemos en tres para la primera y cuatro para la segunda sendos números de cifras decimales. Además, los valores de las columnas ***acidity*** y ***citric.acid*** se fijan a dos cifras decimales. Para terminar esta sección, el tipo *integer* de la variable ***quality*** se ha transformado a *numeric* para ser del todo compatible con el resto de las variables y no provocar conflictos en los valores de regresión y futuras predicciones.

```
# Establecemos el número de cifras decimales en las columnas "acidity", "citric.acid" "chlorides"
y "density"
winequality.red$acidity<-round(winequality.red$acidity, 2)
winequality.red$citric.acid<-round(winequality.red$citric.acid, 2)
winequality.red$chlorides<-round(winequality.red$chlorides, 3)
winequality.red$density<-round(winequality.red$density, 4)
# Convertimos la columna "quality" a tipo "numeric":
winequality.red$quality<-as.numeric(winequality.red$quality)
class(winequality.red$quality)
```

```
## [1] "numeric"
```

2.3.2. Ceros y elementos vacíos

A pesar de que habitualmente se usan ceros como valores centinela para alertar del desconocimiento de ciertos valores, no es el caso de este conjunto de datos, ya que los ceros que aparecen en alguna columna tienen significado numérico y se corresponden con valor nulo de una característica fisicoquímica que se puede medir en los vinos. Por ejemplo, puede ser habitual que un vino carezca de ácido cítrico. La ausencia de este valor se corresponde con un cero y es perfectamente inteligible. Procedemos ahora a detectar los campos que contienen valores vacíos:

```
# Número de valores desconocidos por campo
sapply(winequality.red, function(x) sum(is.na(x)))
```

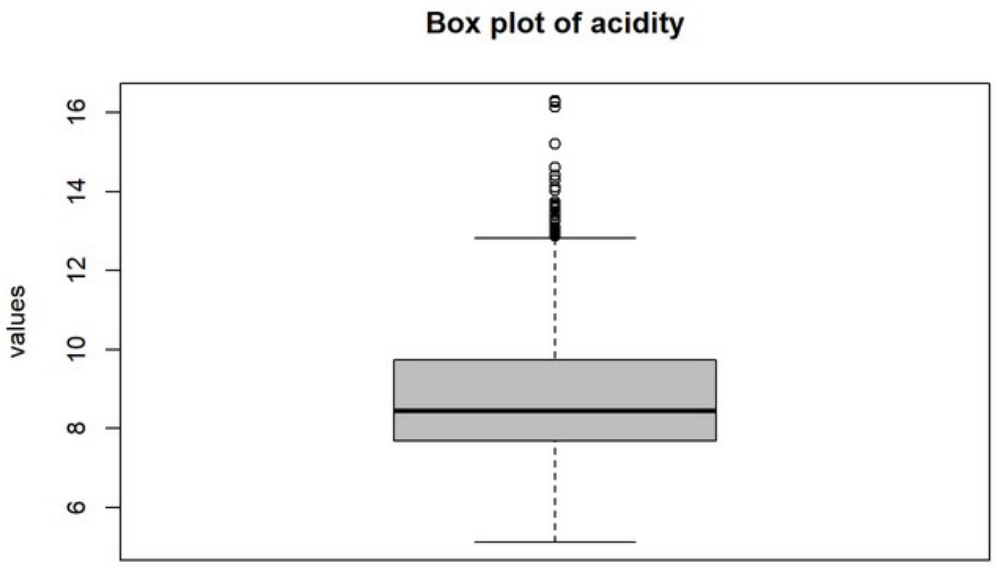
```
##              acidity          citric.acid       residual.sugar
##                    0                    0                    0
##            chlorides total.sulfur.dioxide              density
##                    0                    0                    0
##                   pH         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

No se observa ningún dato vacío en ningún campo, por lo que no hay que plantearse eliminar registros que contengan este tipo de valores ni tampoco aplicar métodos de imputación basados en similitudes o diferencias de los valores más próximos.

2.3.3. Valores extremos (outliers)

Para identificar los valores atípicos (*outliers*), nos valemos de los diagramas de cajas (*boxplots*) para cada variable y de la función **boxplots.stats()** de R. Con esta última opción, se detallan los valores atípicos para cada una de las variables que los contienen.

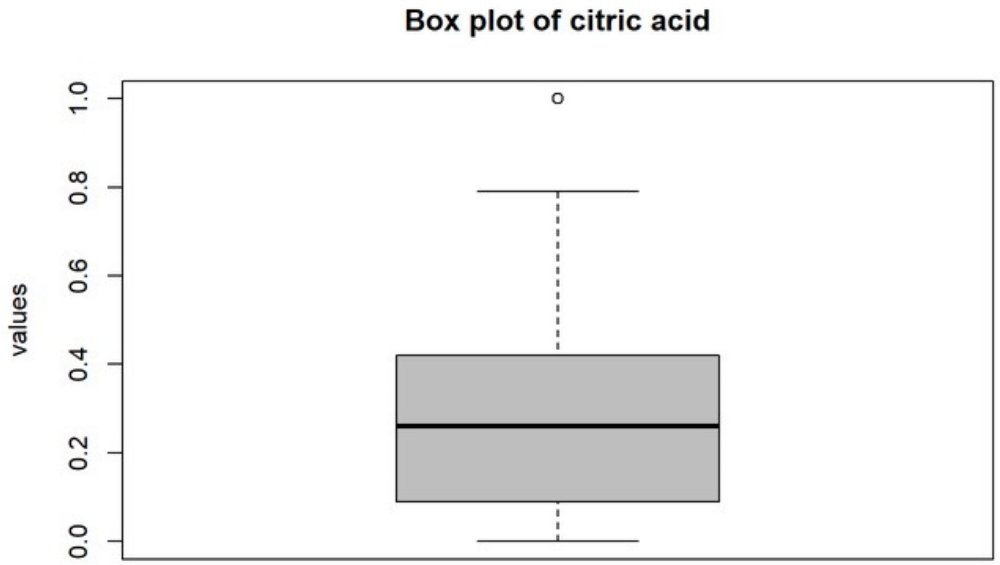
```
boxplot(winequality.red$acidity,main="Box plot of acidity", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$acidity)$out
```

```
##  [1] 13.10 13.10 15.21 15.21 13.06 13.64 13.67 12.89 14.29 14.03 12.98
## [12] 12.96 13.42 13.42 14.41 14.11 14.11 13.30 12.96 13.64 12.91 16.29
## [23] 12.88 13.32 12.87 13.59 13.10 13.25 14.61 16.14 16.14 16.25 13.47
## [34] 13.30 13.47 13.30 13.29 13.66 13.66 13.58 16.26 13.73 13.40 13.01
## [45] 12.99
```

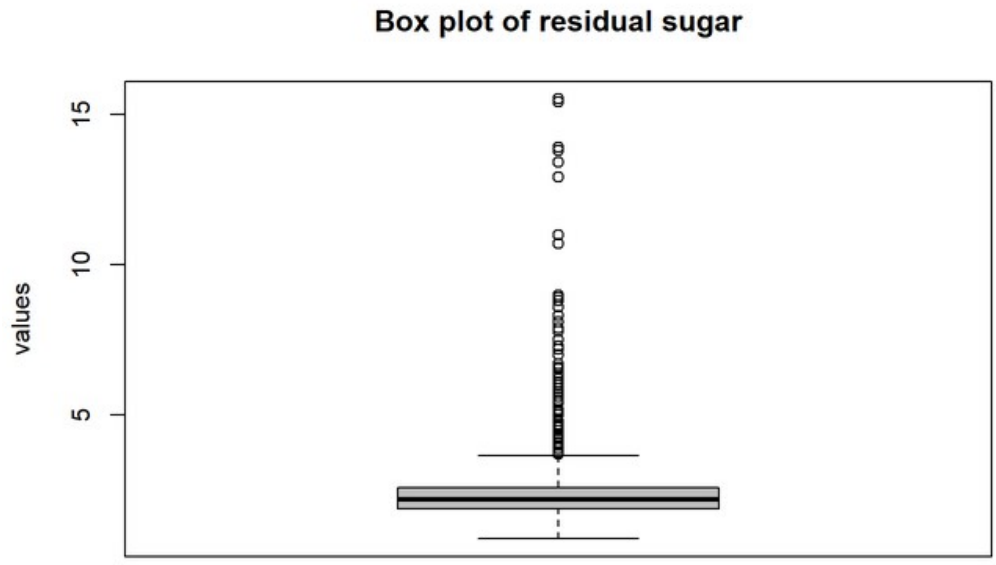
```
boxplot(winequality.red$citric.acid,main="Box plot of citric acid", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$citric.acid)$out
```

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

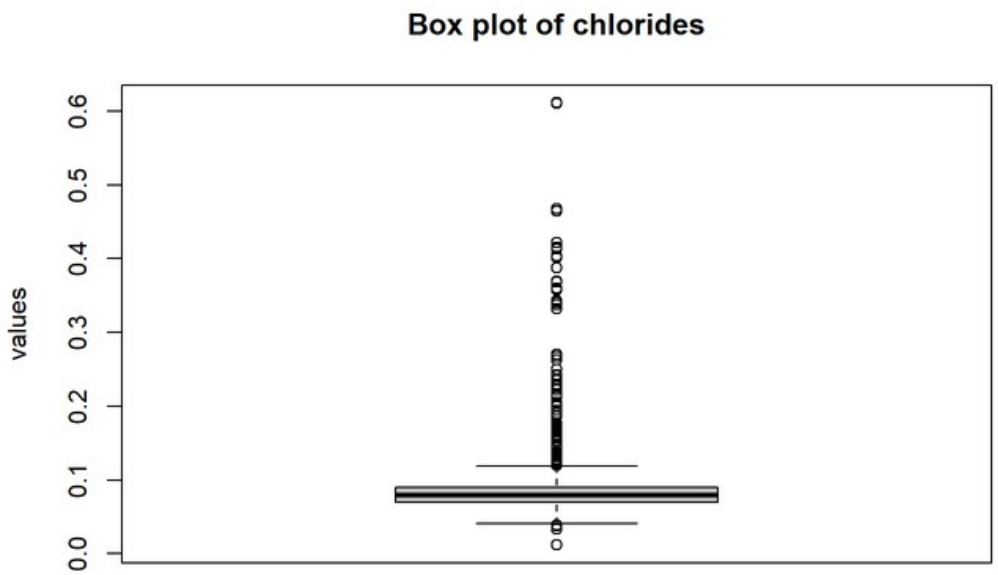
```
boxplot(winequality.red$residual.sugar,main="Box plot of residual sugar", col="gray",ylab="values"
)
```



```
boxplot.stats(winequality.red$residual.sugar)$out
```

```
##   [1]  6.10  6.10  3.80  3.90  4.40 10.70  5.50  5.90  5.90  3.80  5.10
##  [12]  4.65  4.65  5.50  5.50  5.50  5.50  7.30  7.20  3.80  5.60  4.00
##  [23]  4.00  4.00  4.00  7.00  4.00  4.00  6.40  5.60  5.60 11.00 11.00
##  [34]  4.50  4.80  5.80  5.80  3.80  4.40  6.20  4.20  7.90  7.90  3.70
##  [45]  4.50  6.70  6.60  3.70  5.20 15.50  4.10  8.30  6.55  6.55  4.60
##  [56]  6.10  4.30  5.80  5.15  6.30  4.20  4.20  4.60  4.20  4.60  4.30
##  [67]  4.30  7.90  4.60  5.10  5.60  5.60  6.00  8.60  7.50  4.40  4.25
##  [78]  6.00  3.90  4.20  4.00  4.00  4.00  6.60  6.00  6.00  3.80  9.00
##  [89]  4.60  8.80  8.80  5.00  3.80  4.10  5.90  4.10  6.20  8.90  4.00
## [100]  3.90  4.00  8.10  8.10  6.40  6.40  8.30  8.30  4.70  5.50  5.50
## [111]  4.30  5.50  3.70  6.20  5.60  7.80  4.60  5.80  4.10 12.90  4.30
## [122] 13.40  4.80  6.30  4.50  4.50  4.30  4.30  3.90  3.80  5.40  3.80
## [133]  6.10  3.90  5.10  5.10  3.90 15.40 15.40  4.80  5.20  5.20  3.75
## [144] 13.80 13.80  5.70  4.30  4.10  4.10  4.40  3.70  6.70 13.90  5.10
## [155]  7.80
```

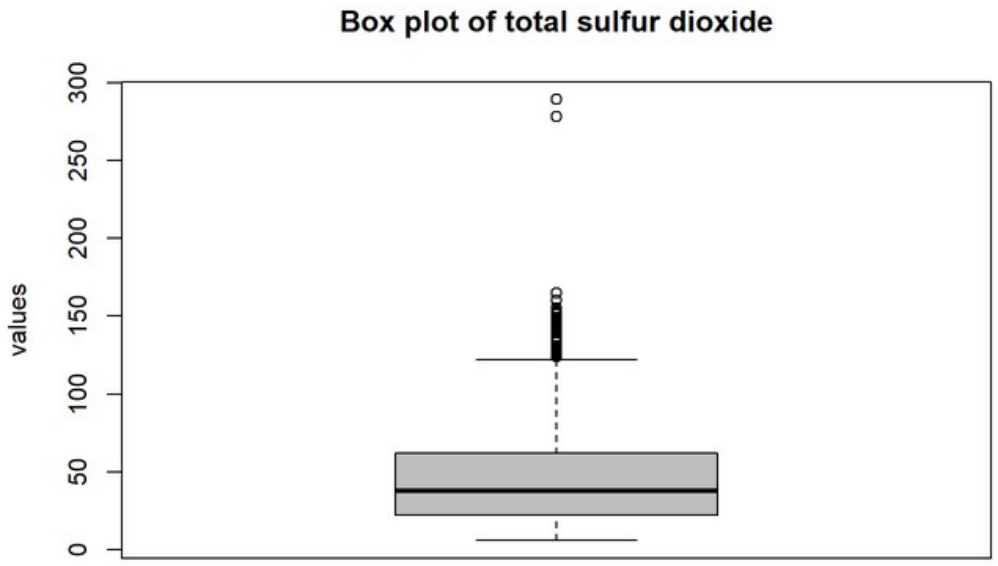
```
boxplot(winequality.red$chlorides,main="Box plot of chlorides", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$chlorides)$out
```

```
##   [1] 0.176 0.170 0.368 0.341 0.172 0.332 0.464 0.401 0.467 0.122 0.178
##  [12] 0.146 0.236 0.610 0.360 0.270 0.039 0.337 0.263 0.611 0.358 0.343
##  [23] 0.186 0.213 0.214 0.121 0.122 0.122 0.128 0.120 0.159 0.124 0.122
##  [34] 0.122 0.174 0.121 0.127 0.413 0.152 0.152 0.125 0.122 0.200 0.171
##  [45] 0.226 0.226 0.250 0.148 0.122 0.124 0.124 0.143 0.222 0.039 0.157
##  [56] 0.422 0.034 0.387 0.415 0.157 0.157 0.243 0.241 0.190 0.132 0.126
##  [67] 0.038 0.165 0.145 0.147 0.012 0.012 0.039 0.194 0.132 0.161 0.120
##  [78] 0.120 0.123 0.123 0.414 0.216 0.171 0.178 0.369 0.166 0.166 0.136
##  [89] 0.132 0.132 0.123 0.123 0.123 0.403 0.137 0.414 0.166 0.168 0.415
## [100] 0.153 0.415 0.267 0.123 0.214 0.214 0.169 0.205 0.205 0.039 0.235
## [111] 0.230 0.038
```

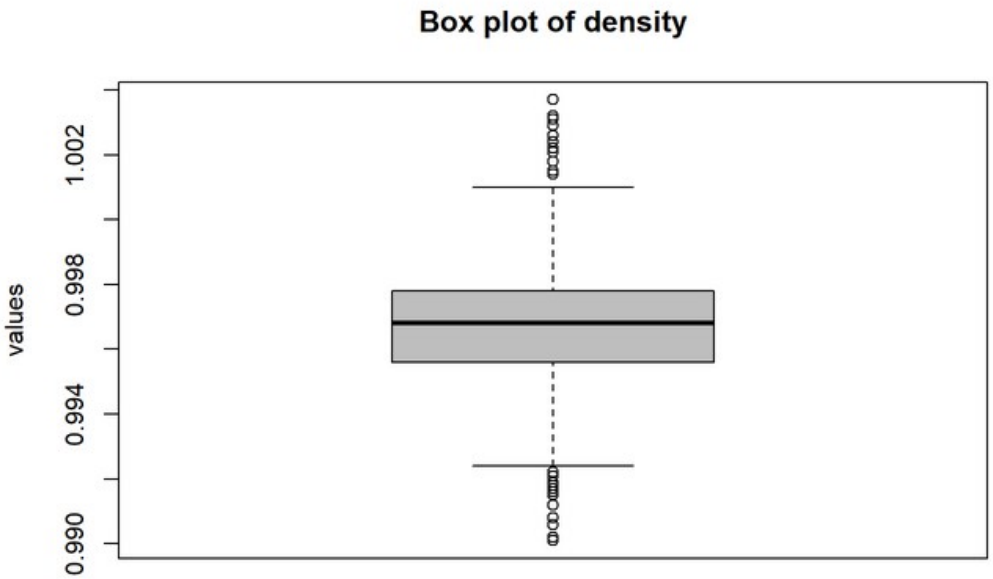
```
boxplot(winequality.red$total.sulfur.dioxide,main="Box plot of total sulfur dioxide", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$total.sulfur.dioxide)$out
```

```
##  [1] 145 148 136 125 140 136 133 153 134 141 129 128 129 128 143 144 127
## [18] 126 145 144 135 165 124 124 134 124 129 151 133 142 149 147 145 148
## [35] 155 151 152 125 127 139 143 144 130 278 289 135 160 141 141 133 147
## [52] 147 131 131 131
```

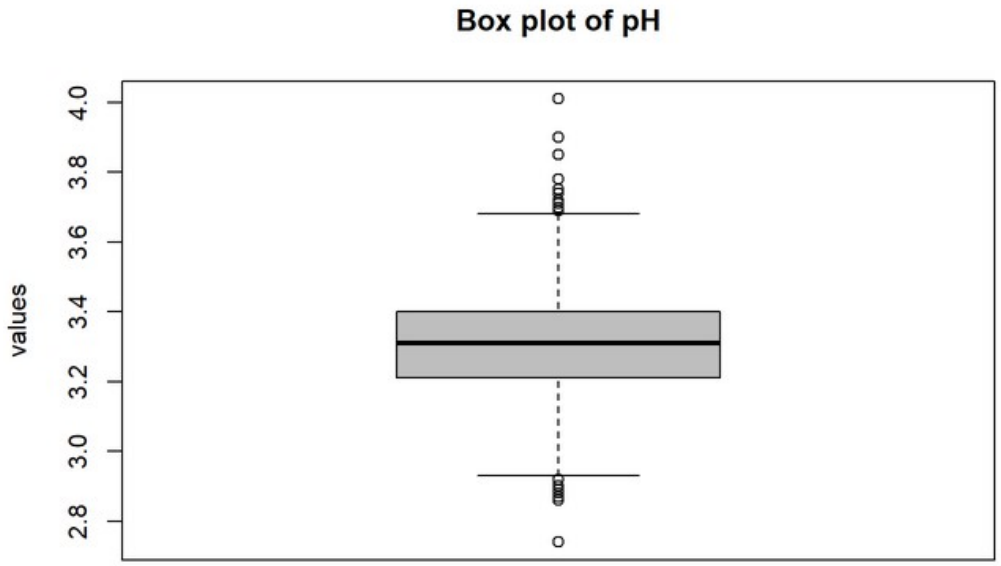
```
boxplot(winequality.red$density,main="Box plot of density", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$density)$out
```

```
##  [1] 0.9916 0.9916 1.0014 1.0015 1.0015 1.0018 0.9912 1.0022 1.0022 1.0014
## [11] 1.0014 1.0014 1.0014 1.0032 1.0026 1.0014 1.0031 1.0031 1.0031 1.0021
## [21] 1.0021 0.9917 0.9922 1.0026 0.9921 0.9915 0.9906 0.9906 1.0029 0.9916
## [31] 0.9901 0.9901 0.9902 0.9922 0.9915 0.9916 0.9908 0.9908 0.9919 1.0037
## [41] 1.0037 1.0024 0.9918 1.0024 0.9918
```

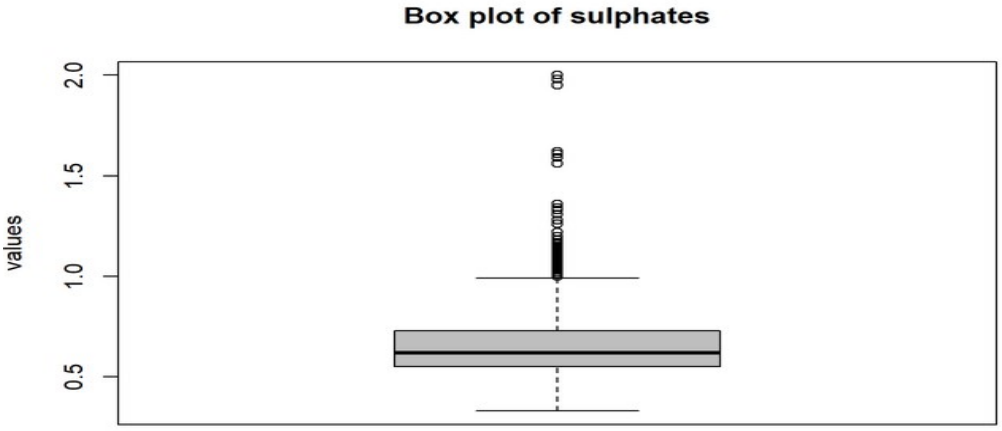
```
boxplot(winequality.red$pH,main="Box plot of pH", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$pH)$out
```

```
##  [1] 3.90 3.75 3.85 2.74 3.69 3.69 2.88 2.86 3.74 2.92 2.92 2.92 3.72 2.87
## [15] 2.89 2.89 2.92 3.90 3.71 3.69 3.69 3.71 3.71 2.89 2.89 3.78 3.70 3.78
## [29] 4.01 2.90 4.01 3.71 2.88 3.72 3.72
```

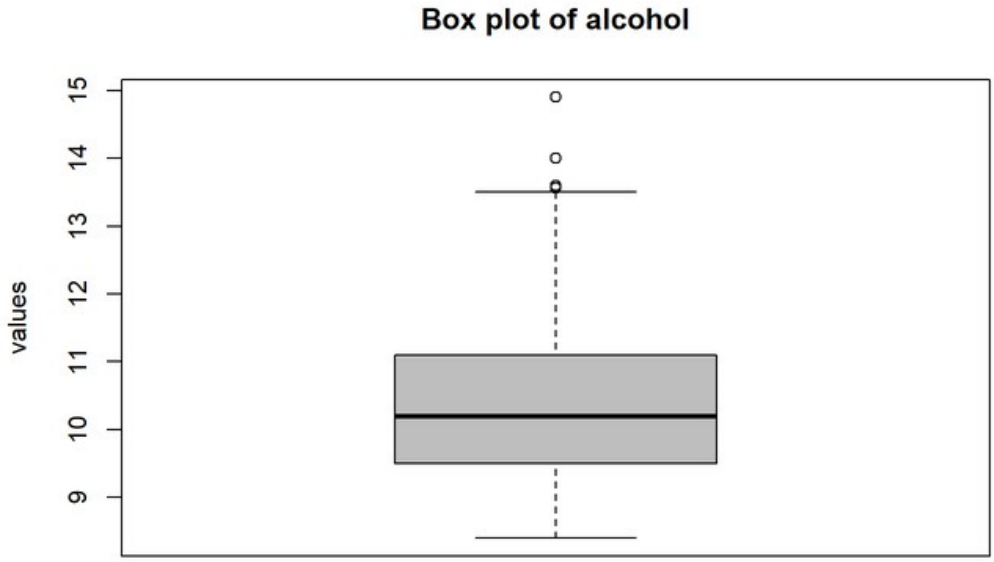
```
boxplot(winequality.red$sulphates,main="Box plot of sulphates", col="gray",ylab="values")
```



```
boxplot.stats(winequality.red$sulphates)$out
```

```
##  [1] 1.56 1.28 1.08 1.20 1.12 1.28 1.14 1.95 1.22 1.95 1.98 1.31 2.00 1.08
## [15] 1.59 1.02 1.03 1.61 1.09 1.26 1.08 1.00 1.36 1.18 1.13 1.04 1.11 1.13
## [29] 1.07 1.06 1.06 1.05 1.06 1.04 1.05 1.02 1.14 1.02 1.36 1.36 1.05 1.17
## [43] 1.62 1.06 1.18 1.07 1.34 1.16 1.10 1.15 1.17 1.17 1.33 1.18 1.17 1.03
## [57] 1.17 1.10 1.01
```

```
boxplot(winequality.red$alcohol,main="Box plot of alcohol", col="gray",ylab="values")
```




```
boxplot.stats(winequality.red$alcohol)$out
```

```
##  [1] 14.00000 14.00000 14.00000 14.00000 14.90000 14.00000 13.60000
##  [8] 13.60000 13.60000 14.00000 14.00000 13.56667 13.60000
```

A pesar de que los valores detectados anteriormente son valores que pueden darse (hay vinos con alto grado de alcohol, baja densidad o extrema acidez), dada la gran cantidad de registros que posee el dataset, el tratamiento que van a recibir, en lugar de dejarlos como se han recogido, va a ser eliminarlos. Es un hecho contrastado que los valores extremos ocasionan serios problemas en análisis estadísticos, ya que se alejan mucho de la población de interés,

provocando incrementos en los errores de varianza y en la fiabilidad de los test estadísticos. Esta es la principal razón que me ha llevado a eliminar los puntos extremos.

```
# Eliminamos valores outliers de cada una de las variables fisicoquímicas.

outliers.acidity <- boxplot(winequality.red$acidity, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$acidity %in% outliers.acidity),]

outliers.citric.acid <- boxplot(winequality.red$citric.acid, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$citric.acid %in% outliers.citric.acid),]

outliers.residual.sugar <- boxplot(winequality.red$residual.sugar, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$residual.sugar %in% outliers.residual.sugar),]

outliers.chlorides <- boxplot(winequality.red$chlorides, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$chlorides %in% outliers.chlorides),]

outliers.total.sulfur.dioxide <- boxplot(winequality.red$total.sulfur.dioxide, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$total.sulfur.dioxide %in% outliers.total.sulfur.di
oxide),]

outliers.density <- boxplot(winequality.red$density, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$density %in% outliers.density),]

outliers.pH <- boxplot(winequality.red$pH, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$pH %in% outliers.pH),]

outliers.sulphates <- boxplot(winequality.red$sulphates, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$sulphates %in% outliers.sulphates),]

outliers.alcohol <- boxplot(winequality.red$alcohol, plot=FALSE)$out
winequality.red <- winequality.red[-which(winequality.red$alcohol %in% outliers.alcohol),]
```

Con todos estos cambios, el nuevo conjunto de datos resultante pasa a tener 10 columnas y 1.182 registros.

```
# Número de columnas y registros o filas del nuevo dataset
ncol(winequality.red)
```

```
## [1] 10
```

```
nrow(winequality.red)
```

```
## [1] 1182
```

2.3.4. Exportación de los datos preprocesados

Una vez sometido el fichero inicial *winequality-red.csv* a los procesos de limpieza, integración y validación descritos anteriormente, se procede a guardar el nuevo conjunto de datos resultante en un nuevo fichero cuyo nombre es ***winequality-red_data_clean.csv***:

```
write.csv(winequality.red, "C:/Users/tonis/Desktop/UOC/Tipologýa y ciclo de vida de los datos/PRAC2/winequality-red_data_clean.csv")
```

2.4. Análisis de los datos

2.4.1. Selección de los grupos de datos que se quieren analizar

En este apartado, se seleccionan grupos de nuestro conjunto de datos para posteriores análisis, comparaciones y realización de pruebas estadísticas. Se establecen subgrupos de densidad y sal presente en los vinos por debajo y por encima de sus respectivos valores medios, así como subgrupos del porcentaje de alcohol en los vinos por debajo y por encima del 11,5 %.

```
# Agrupación por valores de densidad
low.density <- winequality.red[winequality.red$density <= mean(winequality.red$density),]
high.density <- winequality.red[winequality.red$density > mean(winequality.red$density),]

# Agrupación por porcentaje de alcohol en vino
low.alcohol.percentage<- winequality.red[winequality.red$alcohol <= 11.5,]
high.alcohol.percentage <- winequality.red[winequality.red$alcohol > 11.5,]

# Agrupación por cantidadf de sal presente en el vino
low.clhorides <- winequality.red[winequality.red$chlorides <= mean(winequality.red$chlorides),]
high.clhorides <- winequality.red[winequality.red$chlorides > mean(winequality.red$chlorides),]
```

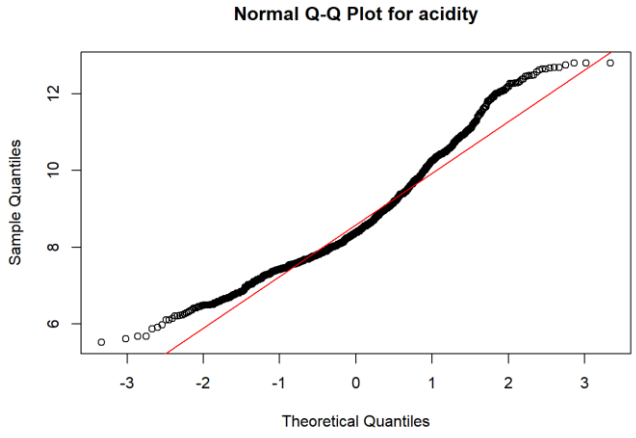
2.4.2. Comprobación de la normalidad y homogeneidad de la varianza

Una primera comprobación de si los valores que toman las variables cuantitativas provienen de una población que sigue una distribución normal se realiza mediante la prueba de normalidad de *Anderson-Darling*. Se trata de un estadístico que mide qué tan bien siguen los datos una distribución específica. Para cada variable analizada, se examina el p-valor obtenido. Si se obtiene un p-valor superior al nivel de significación prefijado (α = 0,05), se puede considerar que la variable en cuestión sigue una distribución normal.

```
library(nortest)
alpha = 0.05
col.names = colnames(winequality.red)
for (i in 1:ncol(winequality.red)) {
if (i == 1) cat("Listado de variables fisicoquímicas que no siguen una distribución normal:\n")
if (is.integer(winequality.red[,i]) | is.numeric(winequality.red[,i])) {
p_val = ad.test(winequality.red[,i])$p.value
if (p_val < alpha) {
cat(col.names[i])
# Format output
if (i < ncol(winequality.red) - 1) cat(", ")
if (i %% 3 == 0) cat("\n")
}
}
}
```

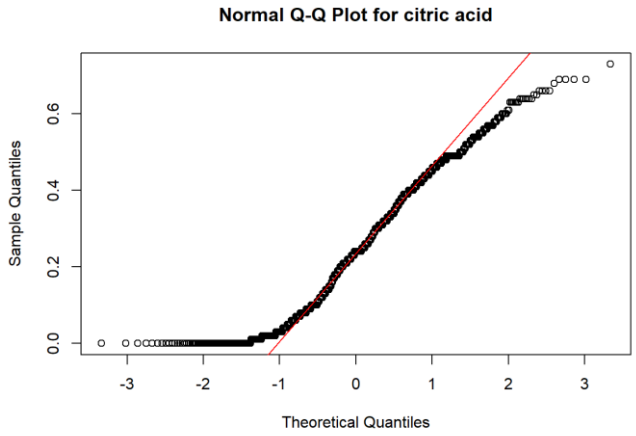
```
## Listado de variables fisicoquímicas que no siguen una distribución normal:
## acidity, citric.acid, residual.sugar,
## chlorides, total.sulfur.dioxide, density,
## pH, sulphates, alcohol
## quality
```

Una forma alternativa de analizar la normalidad de las variables es utilizar pruebas de normalidad de *Shapiro-Wilk*. Estas pruebas utilizan también contraste de hipótesis para rechazar la normalidad de la muestra, asumiendo como hipótesis nula que la muestra proviene de una población distribuida normalmente. Si el p-valor es menor que el nivel de significación ($\alpha = 0{,}05$), se rechaza la hipótesis nula y se considera que hay evidencias como para admitir que la muestra no proviene de una distribución normal. Si es mayor, no se acepta la hipótesis alternativa, simplemente no se rechaza la hipótesis nula (no se demuestra nada). Adicionalmente, se puede contrastar la normalidad visualmente con gráficos Q-Q.



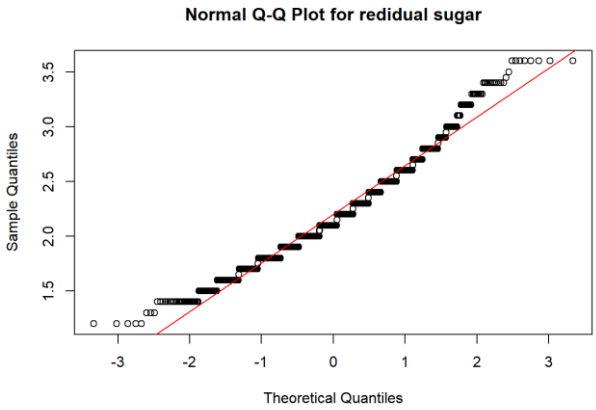
```
shapiro.test(winequality.red$acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$acidity
## W = 0.95654, p-value < 2.2e-16
```



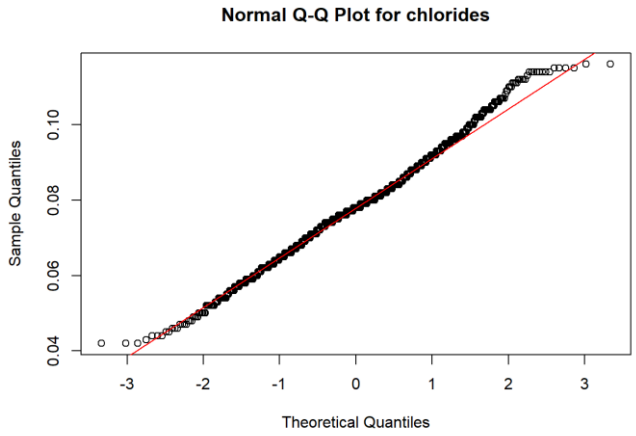

```
shapiro.test(winequality.red$citric.acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$citric.acid
## W = 0.94951, p-value < 2.2e-16
```



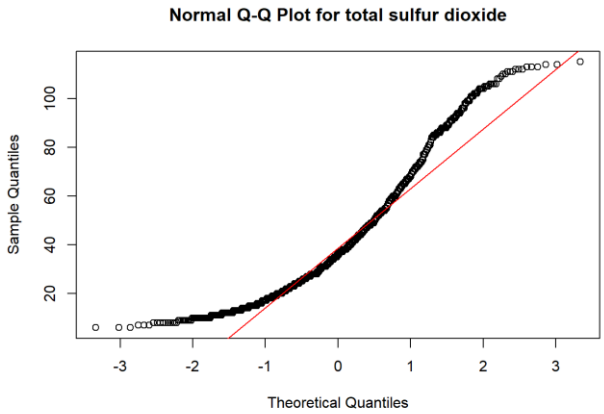
```
shapiro.test(winequality.red$residual.sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$residual.sugar
## W = 0.97058, p-value = 9.184e-15
```



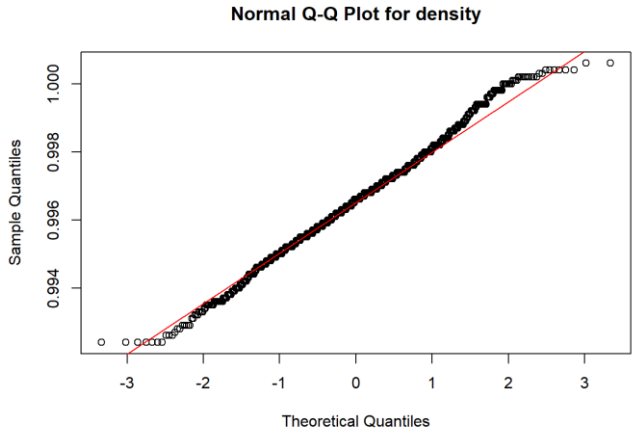

```
shapiro.test(winequality.red$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$chlorides
## W = 0.99382, p-value = 8.049e-05
```



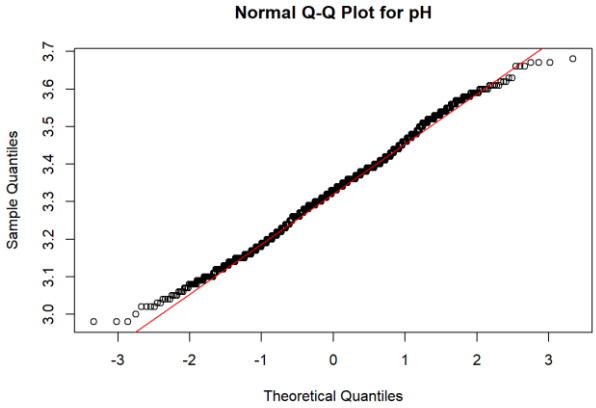
```
shapiro.test(winequality.red$total.sulfur.dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$total.sulfur.dioxide
## W = 0.92227, p-value < 2.2e-16
```



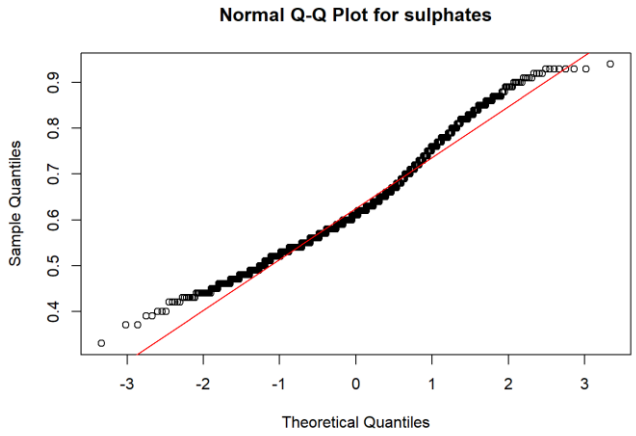

```
shapiro.test(winequality.red$density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$density
## W = 0.99502, p-value = 0.0006067
```



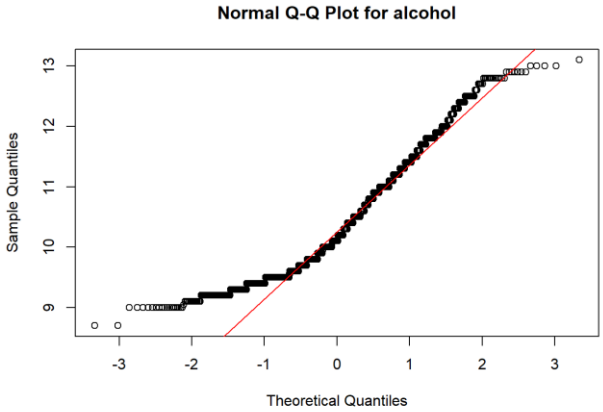
```
shapiro.test(winequality.red$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$pH
## W = 0.99516, p-value = 0.0007893
```



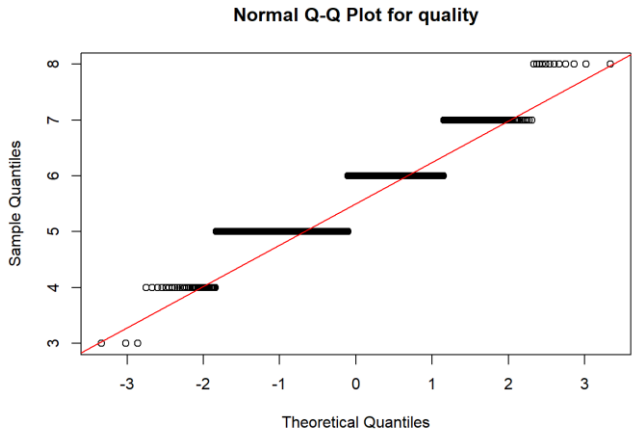

```
shapiro.test(winequality.red$sulphates)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$sulphates
## W = 0.9721, p-value = 2.543e-14
```



```
shapiro.test(winequality.red$alcohol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$alcohol
## W = 0.93279, p-value < 2.2e-16
```




```
shapiro.test(winequality.red$quality)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  winequality.red$quality
## W = 0.84742, p-value < 2.2e-16
```

Los resultados de los gráficos Q-Q sugieren que las variables son susceptibles de normalización, si se precisa hacerlo. Por otra parte, las pruebas delatan que ninguna variable está normalizada, ya que los p-valores son muy inferiores al valor $\alpha = 0{,}05$ y entendemos que podemos rechazar la hipótesis nula y considerar que no se sigue una distribución normal. No obstante, todas las variables se pueden normalizar, ya que, en virtud del teorema del límite central, la muestra es lo bastante grande (más de 30 registros) como para considerar que sigue una distribución normal de media 0 y desviación estándar 1.

A continuación, estudiamos la homogeneidad de varianzas mediante un test de **Fligner-Kileen** sobre los grupos de datos creados anteriormente. En este test, la hipótesis nula consiste en considerar que ambas varianzas son iguales.

```
fligner.test(quality ~ density, data = winequality.red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by density
## Fligner-Killeen:med chi-squared = 84.586, df = 77, p-value =
## 0.2593
```

```
fligner.test(quality ~ alcohol, data = winequality.red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by alcohol
## Fligner-Killeen:med chi-squared = 72.047, df = 50, p-value =
## 0.02225
```

```
fligner.test(quality ~ chlorides, data = winequality.red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by chlorides
## Fligner-Killeen:med chi-squared = 82.893, df = 73, p-value =
## 0.2007
```

Puesto que, para el primer y tercer conjunto de datos (densidad y cantidad de sal en los vinos), obtenemos p-valores mayores que el nivel de significación α = 0,05, aceptamos la hipótesis conforme las varianzas son homogéneas en cada uno de ellos. No obtenemos la misma conclusión en el segundo conjunto de datos (porcentaje de alcohol), ya que p-value = 0,02225 está por debajo del nivel de significación. Se concluye para este último grupo que las varianzas no son iguales.

2.5. Pruebas estadísticas

2.5.1. Variables cuantitativas con más influencia en la calidad de los vinos

Realizamos un análisis de la correlación entre las distintas variables fisicoquímicas con la calidad de los vinos. Se utiliza para tal propósito el coeficiente de correlación de Spearman, ya que no existe una distribución normal en los datos que utilizamos.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable fisicoquímica
# con respecto al campo "quality"
for (i in 1:(ncol(winequality.red) - 1)) {
if (is.integer(winequality.red[,i]) | is.numeric(winequality.red[,i])) {
spearman_test = cor.test(winequality.red[,i],
winequality.red[,length(winequality.red)],
method = "spearman")
corr_coef = spearman_test$estimate
p_val = spearman_test$p.value
# Add row to matrix
pair = matrix(ncol = 2, nrow = 1)
pair[1][1] = corr_coef
pair[2][1] = p_val
corr_matrix <- rbind(corr_matrix, pair)
rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(winequality.red)[i]
}
}
```

```
print(corr_matrix)
```

```
##                        estimate      p-value
## acidity               0.06334928 2.941751e-02
## citric.acid           0.22587602 3.864772e-15
## residual.sugar        0.02399919 4.097446e-01
## chlorides            -0.20202975 2.367987e-12
## total.sulfur.dioxide -0.14232988 8.960679e-07
## density              -0.21470668 8.584723e-14
## pH                   -0.06304193 3.021513e-02
## sulphates             0.43853552 9.900814e-57
## alcohol               0.48457880 1.261876e-70
```

A partir de su proximidad con los valores -1 y +1, identificamos las variables más correlacionadas con la calidad de los vinos. A pesar de no tratarse de sólidas correlaciones, las variables más relevantes en la definición de la calidad son ***alcohol***, ***sulphates***, ***citric.acid***, ***density*** y ***chlorides*** por este orden. Se ha decidido añadir también el p-valor asociado a cada coeficiente de correlación, ya que nos informa acerca del peso estadístico de la correlación obtenida. Estas variables serán utilizadas en modelos de regresión multilineal para predecir la calidad de los vinos.

2.5.2. Correlación entre variables

El siguiente código aporta resultados en forma de tablas y gráficos para representar la correlación que existe entre variables. La información visual que aportan las matrices de correlación facilita

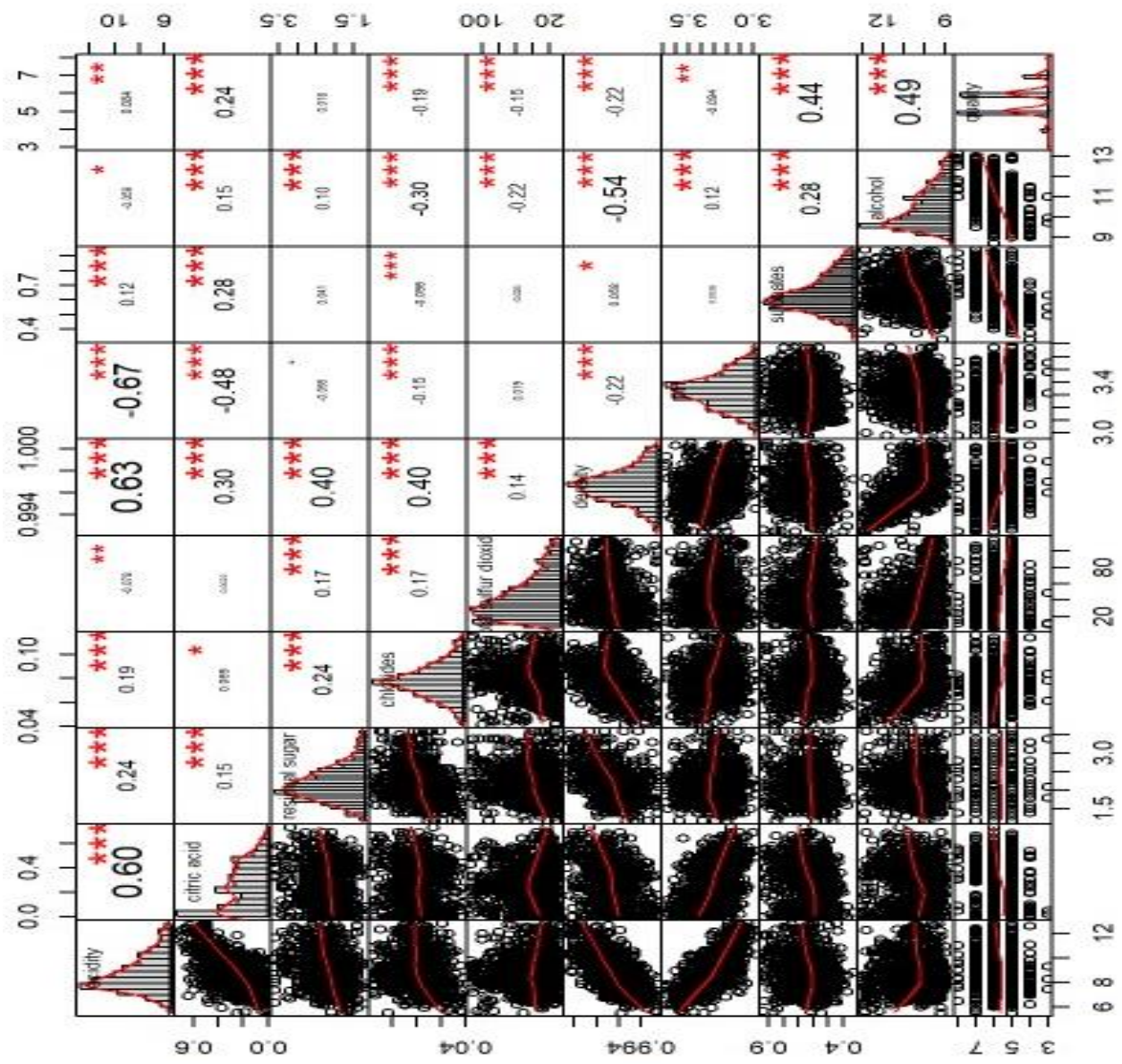
la comprensión de las relaciones que existen entre las propiedades fisicoquímicas de los vinos.

```
library(PerformanceAnalytics)
```

```
# Guardamos datos en un data.frame
acidity<-winequality.red$acidity
citric.acid<-winequality.red$citric.acid
residual.sugar<-winequality.red$residual.sugar
chlorides<-winequality.red$chlorides
total.sulfur.dioxide<-winequality.red$total.sulfur.dioxide
density<-winequality.red$density
pH<-winequality.red$pH
sulphates<-winequality.red$sulphates
alcohol<-winequality.red$alcohol
quality<-winequality.red$quality
data <- data.frame(acidity, citric.acid, residual.sugar, chlorides, total.sulfur.dioxide, density,
 pH, sulphates, alcohol, quality)
colnames(data) <- c("acidity","citric acid","residual sugar","chlorides","total sulfur dioxide","d
ensity", "pH", "sulphates", "alcohol", "quality")
cor(data)
```

```
##                         acidity  citric acid residual sugar   chlorides
## acidity              1.00000000  0.603353709     0.24484897  0.19254182
## citric acid          0.60335371  1.000000000     0.15025882  0.06902013
## residual sugar       0.24484897  0.150258817     1.00000000  0.24054793
## chlorides            0.19254182  0.069020125     0.24054793  1.00000000
## total sulfur dioxide -0.07812217  0.002177295     0.17088644  0.16974256
## density              0.62718180  0.300976996     0.39804848  0.40341417
## pH                  -0.67289768 -0.482954798    -0.05565157 -0.15070000
## sulphates            0.12300305  0.275666484     0.04069258 -0.09583778
## alcohol             -0.05935530  0.146719282     0.10393533 -0.29691699
## quality              0.08447757  0.244631954     0.01784264 -0.18985018
##                      total sulfur dioxide     density           pH
## acidity                       -0.078122174  0.62718180 -0.672897678
## citric acid                    0.002177295  0.30097700 -0.482954798
## residual sugar                 0.170886444  0.39804848 -0.055651570
## chlorides                      0.169742556  0.40341417 -0.150699998
## total sulfur dioxide           1.000000000  0.14109698  0.019098399
## density                        0.141096985  1.00000000 -0.223299857
## pH                             0.019098399 -0.22329986  1.000000000
## sulphates                     -0.024142466  0.05816787  0.003591845
## alcohol                       -0.223753777 -0.54100848  0.117723679
## quality                       -0.151868436 -0.21581441 -0.093901964
##                        sulphates    alcohol     quality
## acidity               0.123003050 -0.0593553  0.08447757
## citric acid           0.275666484  0.1467193  0.24463195
## residual sugar        0.040692585  0.1039353  0.01784264
## chlorides            -0.095837782 -0.2969170 -0.18985018
## total sulfur dioxide -0.024142466 -0.2237538 -0.15186844
## density               0.058167875 -0.5410085 -0.21581441
## pH                    0.003591845  0.1177237 -0.09390196
## sulphates             1.000000000  0.2787311  0.43968490
## alcohol               0.278731106  1.0000000  0.49172395
## quality               0.439684896  0.4917240  1.00000000
```

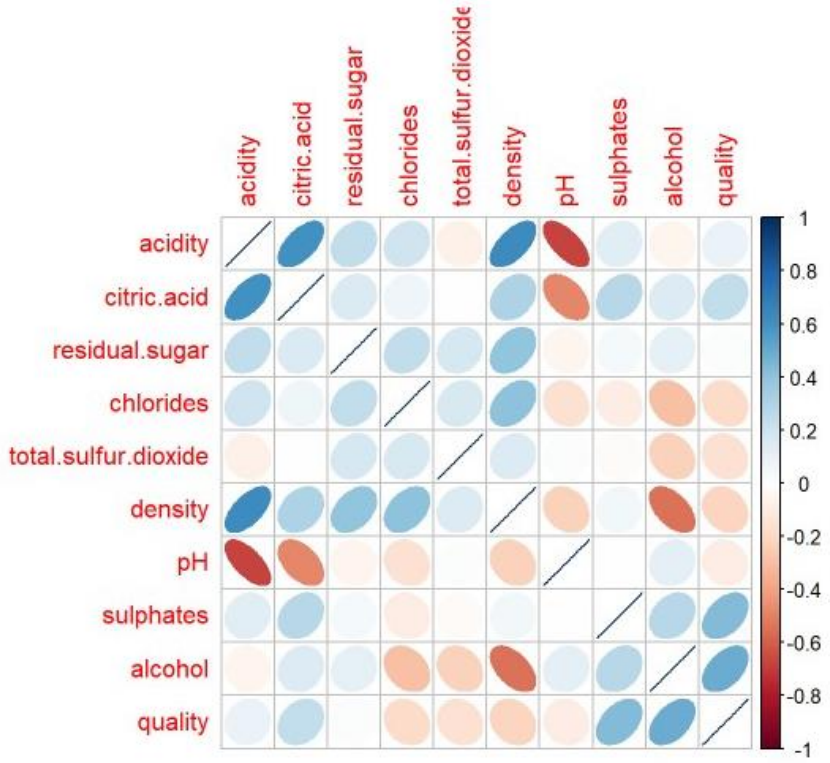
```
chart.Correlation(data)
```



```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
M<-cor(winequality.red)
corrplot(M, method = "ellipse")
```



2.5.3. Pruebas de contraste de hipótesis

Se va a realizar un primer contraste de hipótesis sobre dos muestras para determinar si la calidad de los vinos es superior dependiendo del valor de la densidad (por debajo o por encima de la media). Consideramos para ello dos muestras, la primera se corresponde con los vinos con densidad por debajo de la media del conjunto de datos y la segunda, con aquellos vinos con densidad por encima de dicha media. El tamaño de las muestras valida el test, ya que n > 30.

```
low.density.quality <- winequality.red[winequality.red$density <= mean(winequality.red$density),]$
quality
high.density.quality <- winequality.red[winequality.red$density > mean(winequality.red$density),]$
quality
```

Planteamos el siguiente **contraste de hipótesis de dos muestras sobre la diferencia de medias**:

H_0: $\mu_1 - \mu_2 = 0$
H_1: $\mu_1 - \mu_2 < 0$ (hipótesis alternativa unilateral)

Siendo μ_1 la media de calidad de la población de la que se extrae la primera muestra (vinos menos densos) y μ_2 la media de calidad de la población de la segunda (vinos más densos que la media de densidad). Tomamos como nivel de significación α = 0,05.

```
t.test(low.density.quality, high.density.quality, alternative = "less", conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  low.density.quality and high.density.quality
## t = 5.4571, df = 1155.5, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.3162226
## sample estimates:
## mean of x mean of y
##  5.764605  5.521667
```

Como el p-valor obtenido supera el valor del nivel de significación, no podemos rechazar la hipótesis nula. Por consiguiente, no podemos afirmar que los vinos con densidad por debajo de la media de densidad sean de menor calidad que los vinos con densidad superior a la media de densidad.

Procederemos ahora de forma análoga para determinar, mediante contraste de hipótesis sobre la diferencia de medias, si los vinos con menos presencia de sal tienen la misma calidad o no que los vinos con menos presencia de sal en su composición. En primer lugar, definimos los grupos de datos:

```
low.chlorides.quality <- winequality.red[winequality.red$chlorides <= mean(winequality.red$chlorid
es),]$quality
high.chlorides.quality <- winequality.red[winequality.red$chlorides > mean(winequality.red$chlorid
es),]$quality
```

Planteamos el nuevo contraste de hipótesis:

H_0: $\mu_1 - \mu_2 = 0$
H_1: $\mu_1 - \mu_2 \neq 0$ (hipótesis alternativa bilateral)

Siendo μ_1 la media de calidad de la población de la que se extrae la primera muestra (vinos menos salados) y μ_2 la media de calidad de la población de la segunda (vinos más salados). Tomamos de nuevo, como nivel de significación, α = 0,05.

```
t.test(low.chlorides.quality, high.chlorides.quality, alternative = "two.sided", conf.level = 0.95
)
```

```
##
##  Welch Two Sample t-test
##
## data:  low.chlorides.quality and high.chlorides.quality
## t = 5.5282, df = 1178.3, p-value = 3.981e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1576308 0.3310735
## sample estimates:
## mean of x mean of y
##  5.757674  5.513321
```

El p-valor obtenido está muy por debajo del nivel de significación, por lo que se debe rechazar la hipótesis nula. Se puede concluir, por consiguiente, que los vinos menos salados tienen una calidad distinta que los del grupo más salados (de hecho, los vinos con sal por debajo del nivel medio tienen un poco más de calidad que los que tienen un nivel de sal superior al valor medio).

2.5.4. Modelos de regresión lineal y logística

Nos proponemos en esta sección poder realizar predicciones sobre la calidad de los vinos a partir de sus características fisicoquímicas más influyentes en la calidad. En una primera instancia, se calculará un modelo de regresión lineal con regresores cuantitativos para predecir la calidad. Nos propondremos también predecir la acidez de un vino a partir de las tres variables más influyentes en la acidez (ácido cítrico, acidez y pH) ; en una segunda instancia, habiendo creado una variable binaria que defina un nivel de densidad que supere al del agua (1 g/cm^3), se calculará un modelo de regresión logístico para predecir si un vino pertenece a este nivel de densidad, dadas unas características fisicoquímicas estrechamente relacionadas con esta propiedad del vino. Para garantizar la eficiencia del modelo de regresión lineal, se propondrán varios modelos de regresión usando las variables más influyentes en la calidad, a partir de la tabla obtenida en el apartado 2.5.1. El modelo elegido será el que presente un mayor coeficiente de determinación R^2.

```
# Regresores cuantitativos más influyentes en la calidad de los vinos
alcohol<-winequality.red$alcohol
sulphates<-winequality.red$sulphates
citric.acid<-winequality.red$citric.acid
density<-winequality.red$density
chlorides<-winequality.red$chlorides
total.sulfur.dioxide<-winequality.red$total.sulfur.dioxide
# Variable que se quiere predecir
quality<-winequality.red$quality
# Modelos de regresión lineal
modelo1 <- lm(quality ~ alcohol + sulphates + citric.acid, data = winequality.red)
modelo2 <- lm(quality ~ alcohol + sulphates + citric.acid + chlorides, data = winequality.red)
modelo3 <- lm(quality ~ alcohol + sulphates + citric.acid + chlorides + density, data = winequalit
y.red)
modelo4 <- lm(quality ~ alcohol + sulphates + citric.acid + density + total.sulfur.dioxide, data =
 winequality.red)
```

La siguiente tabla muestra el modelo de regresión con mejor coeficiente:

```
# Tabla con los coeficientes de determinación de cada modelo
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
2, summary(modelo2)$r.squared,
3, summary(modelo3)$r.squared,
4, summary(modelo4)$r.squared),
ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes
```

```
##      Modelo       R^2
## [1,]      1 0.3510071
## [2,]      2 0.3539755
## [3,]      3 0.3572902
## [4,]      4 0.3594511
```

Asumimos que el cuarto modelo es el óptimo, ya que tiene mejor coeficiente de determinación. El modelo resultante es:

```
summary(modelo4)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + sulphates + citric.acid + density +
##     total.sulfur.dioxide, data = winequality.red)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4797 -0.3792 -0.0628  0.4683  1.9680
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.805e+01  1.545e+01   3.109  0.00192 **
## alcohol               2.543e-01  2.564e-02   9.919  < 2e-16 ***
## sulphates             2.154e+00  1.731e-01  12.443  < 2e-16 ***
## citric.acid           5.990e-01  1.150e-01   5.207 2.26e-07 ***
## density              -4.664e+01  1.537e+01  -3.034  0.00247 **
## total.sulfur.dioxide -1.831e-03  7.327e-04  -2.499  0.01258 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6201 on 1176 degrees of freedom
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
## F-statistic:   132 on 5 and 1176 DF,  p-value: < 2.2e-16
```

quality = 48.05 – 0.2543 * alcohol + 2.154 * sulphates – 0.599 * citric.acid – 46.64 * density – 0.001831 * total.sulfur.dioxide

Vamos a realizar ahora un ejemplo de predicción de la calidad del vino, tomando como valores de cada variable influyente sus correspondientes valores medios:

```
newdata <- data.frame(
alcohol = mean(winequality.red$alcohol),
sulphates = mean(winequality.red$sulphates),
citric.acid = mean(winequality.red$citric.acid),
density = mean(winequality.red$density),
total.sulfur.dioxide = mean(winequality.red$total.sulfur.dioxide)
)
# Predecir el precio
predict(modelo4, newdata)
```

```
##        1
## 5.641286
```

El valor predicho de la calidad para un vino de las características especificadas es **5.641286**

Vamos a realizar ahora un modelo de regresión lineal para predecir la acidez de un vino a partir de las características fisicoquímicas más influyentes. Según la matriz de la sección 2.5.2., estas características son la cantidad de ácido cítrico, la densidad del vino y su pH.

```
# Regresores cuantitativos más influyentes en la calidad de los vinos
citric.acid<-winequality.red$citric.acid
density<-winequality.red$density
pH<-winequality.red$pH
# Variable que se quiere predecir
acidity<-winequality.red$acidity
# Modelo de regresión lineal
modelo <- lm(acidity ~ citric.acid + density + pH)
summary(modelo)
```

```
##
## Call:
## lm(formula = acidity ~ citric.acid + density + pH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.58139 -0.47482 -0.02323  0.49081  2.39549
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -376.8421    14.0437  -26.83   <2e-16 ***
## citric.acid    1.9641     0.1379   14.25   <2e-16 ***
## density      402.5965    14.0285   28.70   <2e-16 ***
## pH            -4.8608     0.1843  -26.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7289 on 1178 degrees of freedom
## Multiple R-squared:  0.7374, Adjusted R-squared:  0.7368
## F-statistic:  1103 on 3 and 1178 DF,  p-value: < 2.2e-16
```

El modelo sugerido es:

acidity = -376.8421 + 1.9641 * citric.acid + 402.5965 * density – 4.8608 * pH

Vamos a predecir ahora el valor de la acidez para los valores ácido cítrico = 0.489, densidad = 0.998 y pH = 3.8.

```
data <- data.frame(citric.acid = 0.489, density = 0.998, pH = 3.8)
# Predicción de la acidez
predict(modelo, data)
```

```
##        1
## 7.438701
```

El valor de acidez que predice el modelo con estos valores de las variables regresoras es **7.438701**.

Nos proponemos ahora obtener un modelo de regresión logística para evaluar la probabilidad de que un vino alcance un valor de densidad superior a 1. La variable independiente es binaria e indica si un vino es muy denso, en el caso de que la variable *density* alcance un valor igual o superior a 1 (densidad del agua destilada). Se utiliza la muestra disponible para estimar el modelo con las variables más influyentes en la densidad, que son acidity, alcohol, residual.sugar y chlorides.

En primer lugar, se crea la variable binaria ***high.density***. Se le da el valor 1 si el vino tiene densidad igual o superior a 1 g/cm^3, y 0 en caso contrario.

```
# Creación de la variable binaria "high.density"
winequality.red$density[winequality.red$density >= 1]<1
```

```
winequality.red$density[winequality.red$density < 1]<-0
high.density<-winequality.red$density
high.density<-factor(high.density)
```

Seguidamente, establecemos las variables explicativas que se van a usar para predecir la densidad.

```
# Variables explicativas de la densidad
acidity<-winequality.red$acidity
alcohol<-winequality.red$alcohol
residual.sugar<-winequality.red$residual.sugar
chlorides<-winequality.red$chlorides
```

Con todos estos datos, el modelo de regresión logística estimado es:

```
# Estimación del modelo de regresión logística
reglog <- glm(high.density ~ acidity+alcohol+residual.sugar+chlorides, data = winequality.red, fam
ily = binomial, control = list(maxit = 1000))
summary(reglog)
```

```
##
## Call:
## glm(formula = high.density ~ acidity + alcohol + residual.sugar +
##     chlorides, family = binomial, data = winequality.red, control = list(maxit = 1000))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7816  -0.0736  -0.0213  -0.0073   3.4087
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -19.2219     4.4742  -4.296 1.74e-05 ***
## acidity          1.6387     0.2325   7.048 1.82e-12 ***
## alcohol         -1.3362     0.3947  -3.385 0.000711 ***
## residual.sugar   2.4316     0.5298   4.590 4.44e-06 ***
## chlorides       72.4817    21.9921   3.296 0.000981 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 294.12  on 1181  degrees of freedom
## Residual deviance: 135.26  on 1177  degrees of freedom
## AIC: 145.26
##
## Number of Fisher Scoring iterations: 9
```

high.density = -19.2219 + 1.6387 * acidity -1.3362 * alcohol + 2.4316 * residual.sugar + 72.4817 * chlorides

Creamos ahora un conjunto de datos para realizar la predicción de la densidad:

```
# Creación del dataset con los datos necesarios para la predicción
newdata = data.frame(acidity = 6.36,alcohol = 8.496, residual.sugar = 2.226, chlorides=0.198)
# Usamos la función predict() para calcular la probabilidad predicha. Para obtener la predicción,
se incluye el argumento type = "response"
predict(reglog, newdata, type="response")
```

El resultado estimado es:

```
##         1
## 0.4041297
```

La probabilidad pronosticada es 0.4041297, esto es, un 40.41 % de que la densidad sea 1 g/cm^3 o superior.

2.5.5. Representación de resultados

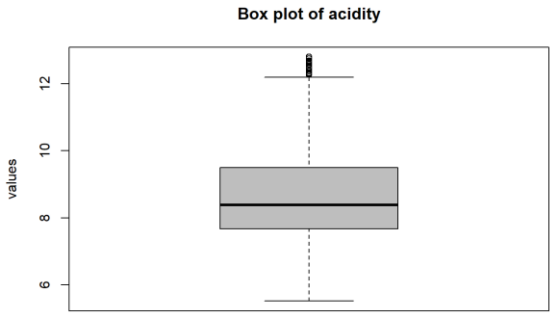
La siguiente tabla resume las principales variables del conjunto de datos obtenido una vez limpiado y preprocesado, siendo la base sobre la que se ha trabajado en los análisis.

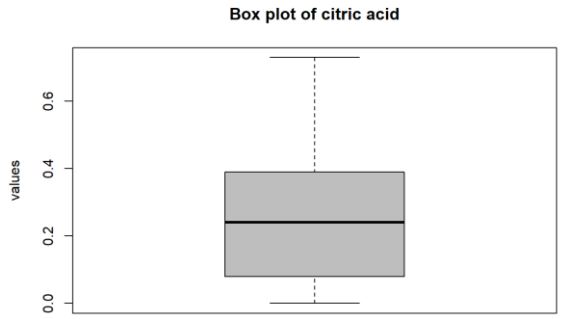
```
# Tabla resumen de las principales variables fisicoquímicas del conjunto de datos
summary(winequality.red)
```

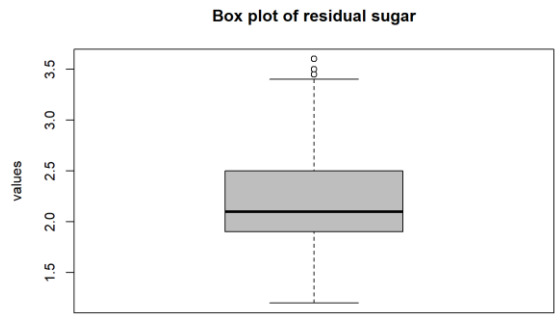
```
    acidity        citric.acid     residual.sugar    chlorides       total.sulfur.dioxide
 Min.   : 5.520   Min.   :0.0000   Min.   :1.200   Min.   :0.04200   Min.   :  6.00
 1st Qu.: 7.680   1st Qu.:0.0800   1st Qu.:1.900   1st Qu.:0.06900   1st Qu.: 22.00
 Median : 8.380   Median :0.2400   Median :2.100   Median :0.07800   Median : 36.00
 Mean   : 8.681   Mean   :0.2459   Mean   :2.183   Mean   :0.07817   Mean   : 41.79
 3rd Qu.: 9.498   3rd Qu.:0.3900   3rd Qu.:2.500   3rd Qu.:0.08675   3rd Qu.: 55.00
 Max.   :12.800   Max.   :0.7300   Max.   :3.600   Max.   :0.11600   Max.   :115.00
    density             pH          sulphates         alcohol         quality
 Min.   :0.9924   Min.   :2.980   Min.   :0.3300   Min.   : 8.70   Min.   :3.000
 1st Qu.:0.9955   1st Qu.:3.230   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
 Median :0.9966   Median :3.330   Median :0.6100   Median :10.10   Median :6.000
 Mean   :0.9966   Mean   :3.326   Mean   :0.6294   Mean   :10.37   Mean   :5.641
 3rd Qu.:0.9975   3rd Qu.:3.410   3rd Qu.:0.7000   3rd Qu.:11.00   3rd Qu.:6.000
 Max.   :1.0006   Max.   :3.680   Max.   :0.9400   Max.   :13.10   Max.   :8.000
```

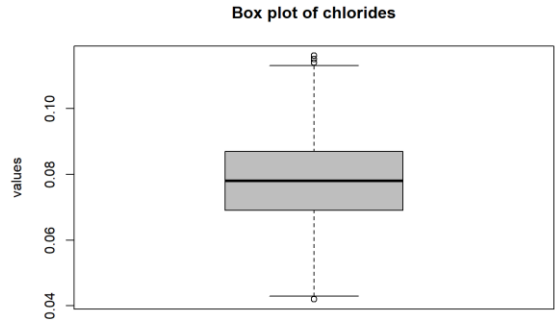
Una representación visual de los mismos datos la proporcionan los siguientes *boxplots*:

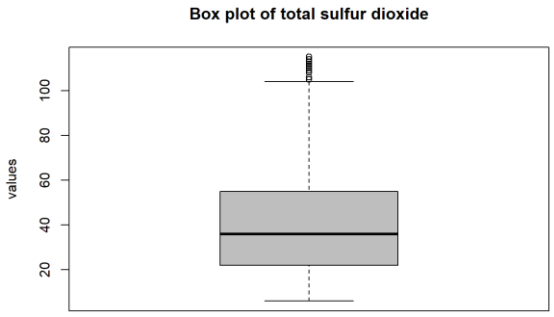
```
boxplot(winequality.red$acidity,main="Box plot of acidity", col="gray",ylab="values")
boxplot(winequality.red$citric.acid,main="Box plot of citric acid", col="gray",ylab="values")
boxplot(winequality.red$residual.sugar,main="Box plot of residual sugar",
col="gray",ylab="values")
boxplot(winequality.red$chlorides,main="Box plot of chlorides", col="gray",ylab="values")
boxplot(winequality.red$total.sulfur.dioxide,main="Box plot of total sulfur dioxide",
col="gray",ylab="values")
boxplot(winequality.red$density,main="Box plot of density", col="gray",ylab="values")
boxplot(winequality.red$pH,main="Box plot of pH", col="gray",ylab="values")
boxplot(winequality.red$sulphates,main="Box plot of sulphates", col="gray",ylab="values")
boxplot(winequality.red$alcohol,main="Box plot of alcohol", col="gray",ylab="values")
```

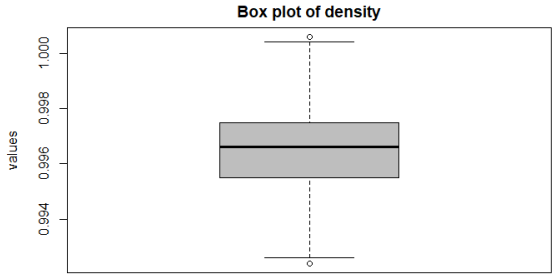


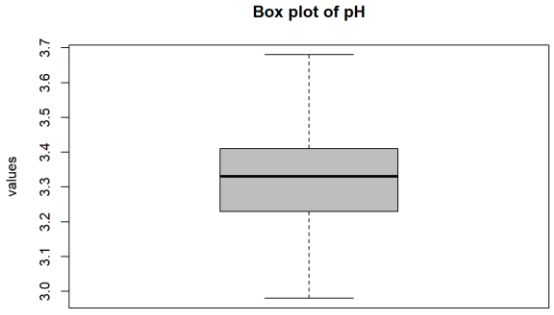

Box plot of citric acid
values
0.0
0.2
0.4
0.6

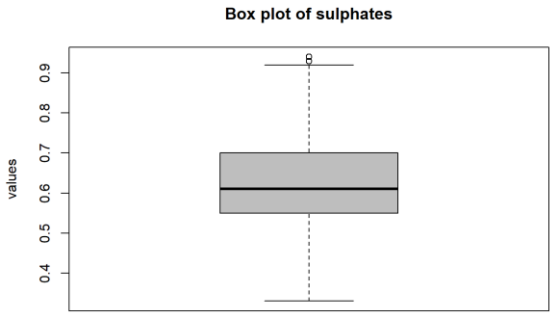

Box plot of residual sugar
values
1.5
2.0
2.5
3.0
3.5

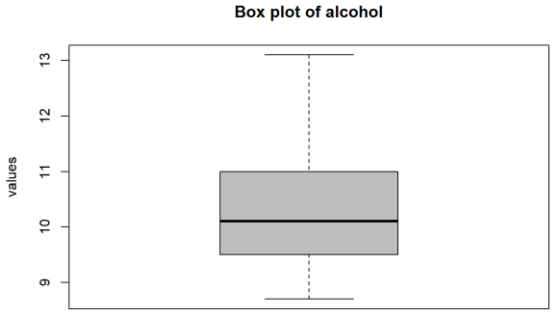

Box plot of chlorides
values
0.04
0.06
0.08
0.10


Box plot of total sulfur dioxide
values
20
40
60
80
100


Box plot of density
values
0.994
0.996
0.998
1.000


Box plot of pH
values
3.0
3.1
3.2
3.3
3.4
3.5
3.6
3.7





2.6. Conclusiones

Este trabajo ha desarrollado una serie de pruebas estadísticas sobre un conjunto de datos correspondiente a un listado de propiedades fisicoquímicas de cierto tipo de vinos rosados portugueses, con el objeto de cumplir, en la medida de lo posible, con un claro objetivo de predicción de la calidad y otras propiedades de los vinos marcado desde el comienzo. Mediante tablas y gráficos, hemos detectado los resultados arrojados por estas pruebas y hemos extraído conocimiento a partir de las mismas. En este sentido, los análisis de correlación y los contrastes de hipótesis han permitido conocer qué variables ejercían una mayor influencia sobre la calidad de los vinos y cómo se influenciaban las propiedades de los vinos entre sí. Por su parte, los modelos de regresión lineal han sido útiles para predecir variables a partir de unas características concretas. Lógicamente, los datos con los que se ha trabajado han sido preprocesados eliminando columnas innecesarias o redundantes, manejando ceros o elementos vacíos y valores extremos (*outliers*). En el caso de los ceros, se han mantenido porque se ha considerado que corresponden a valores reales con significado físico, representando a la magnitud dentro de un rango contemplado y previsible de valores. En cuanto a los valores *outliers*, se ha optado por eliminarlos atendiendo al gran volumen de datos del conjunto y a la voluntad de no pretender efectos negativos en las varianzas y en los resultados de las correlaciones y test. Si bien los

coeficientes de correlación en los análisis de regresión no han sido en ocasiones excesivamente robustos, considero que este trabajo cumple los objetivos marcados en el inicio y puede considerarse exitoso. En otras palabras, este trabajo responde al problema que inicialmente se plantea.

3 Recursos

Los siguientes recursos han sido de utilidad para la realización de esta práctica:

- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.
- Tutorial de Github (https://guides.github.com/activities/hello-world/)
- *Test for homogeneity of variances - Lavene's test and the Fligner Killeen test* (2016) [en línea]. bioSt@TS. [Consulta: 26 de diciembre de 2017] https://biostats.w.uib.no/test-for-homogeneity-of-variances-levenes-test/
- https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009