

Práctica 2: Limpieza y validación de los datos

ANTONIO SANCHEZ NAVARRO

24 de diciembre 2018

Table of Contents

Carga de los datos.....	1
Resolución	2
Integración y selección de los datos de interés.....	9
Detección de ceros y elementos vacíos por campo	10
Valores extremos	10
Exportación de los datos preprocesados.....	19
Análisis de resultados	19
Selección de los grupos de datos que se quieren comparar.....	19
Pruebas de normalidad y homogeneidad de la varianza	20
Pruebas estadísticas	31
Influencia de las variables fisicoquímicas en la calidad de los vinos.....	31
Matriz de correlación entre variables.....	33
Contrastes de hipótesis	37
Modelo de regresión lineal.....	37
Modelo de regresión logístico.....	40
Tabla resumen de los datos preprocesados y representación en forma de boxplots	41
Referencias.....	47

Carga de los datos

En esta actividad se usará el fichero winequality-red.csv del repositorio Github, el cual precisa tareas de preprocesado (limpieza, integración y validación) para posterior análisis. Los datos a tratar corresponden a variables físicoquímicas correspondientes a variantes rojas del vino portugués “Vinho Verde”, las cuales se prestan a tareas de clasificación o análisis de regresión. Las clases están ordenadas y no son equilibradas (por ejemplo, hay muchos más vinos normales que vinos excelentes o pobres).

El archivo se denomina *C:/Users/Antonio/Desktop/UOC/Tipología y ciclo de vida de los datos/PRAC2/winequality-red.csv*, contiene 1599 registros y 12 variables. Estas variables son: fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide, density, pH, sulphates, alcohol, quality

```
# Cargo el archivo de datos "winequality-red.csv" y valido que Los tipos
# de datos se interpretan correctamente
winequality.red <- read.csv("C:/Users/Antonio/Desktop/UOC/Tipología 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
```

Resolución

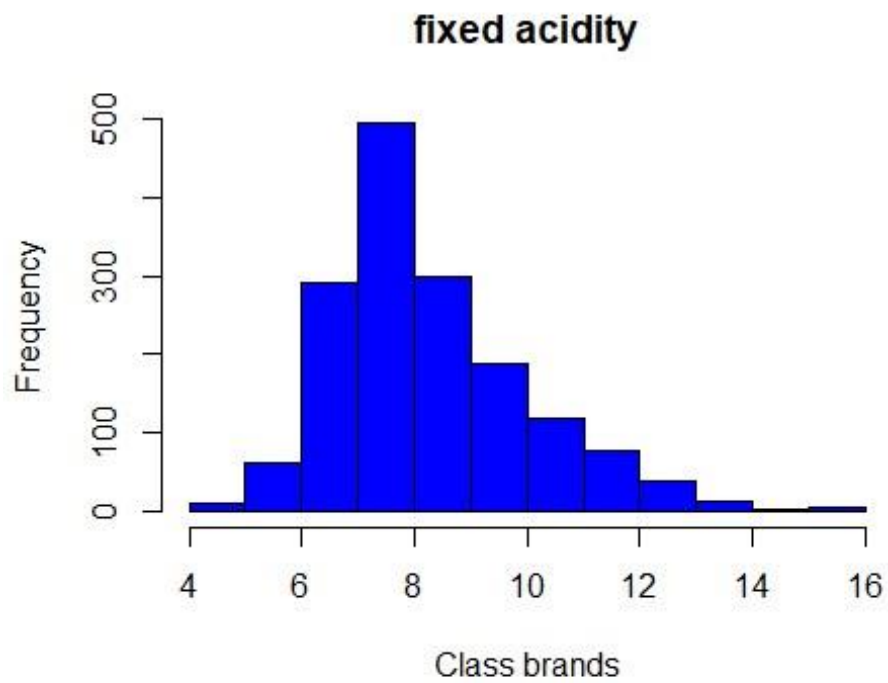
Examinamos el tipo de dato asociado a cada campo

```
# Tipo de dato asignado a cada campo sapply(winequality.red,
function(x) class(x))

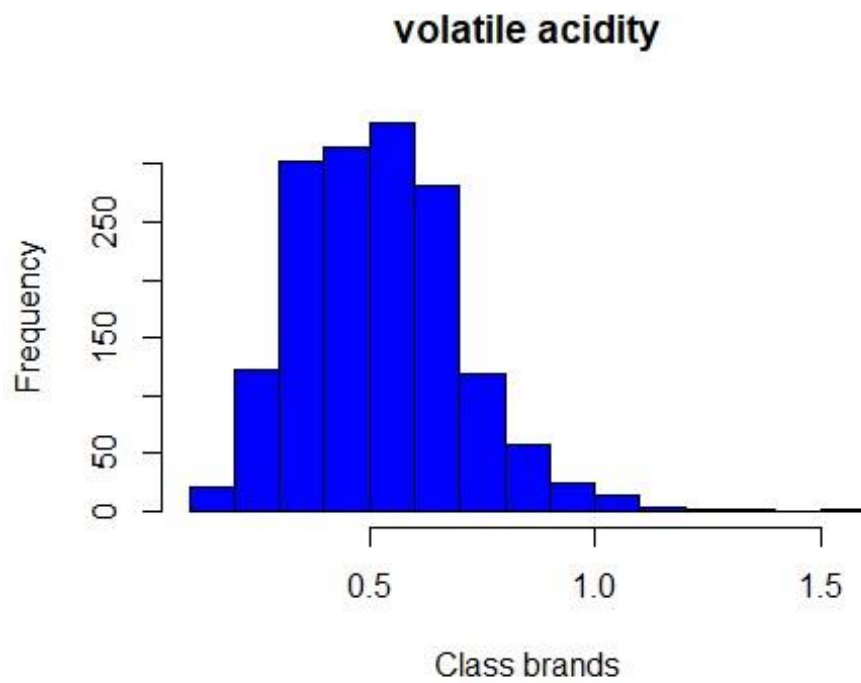
##      fixed.acidity      volatile.acidity      citric.acid
##      "numeric"         "numeric"         "numeric"
##      residual.sugar      chlorides      free.sulfur.dioxide
##      "numeric"         "numeric"         "numeric"
##      total.sulfur.dioxide      density      pH
##      "numeric" 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

Histograma de frecuencias absolutas para cada variable fisicoquímica

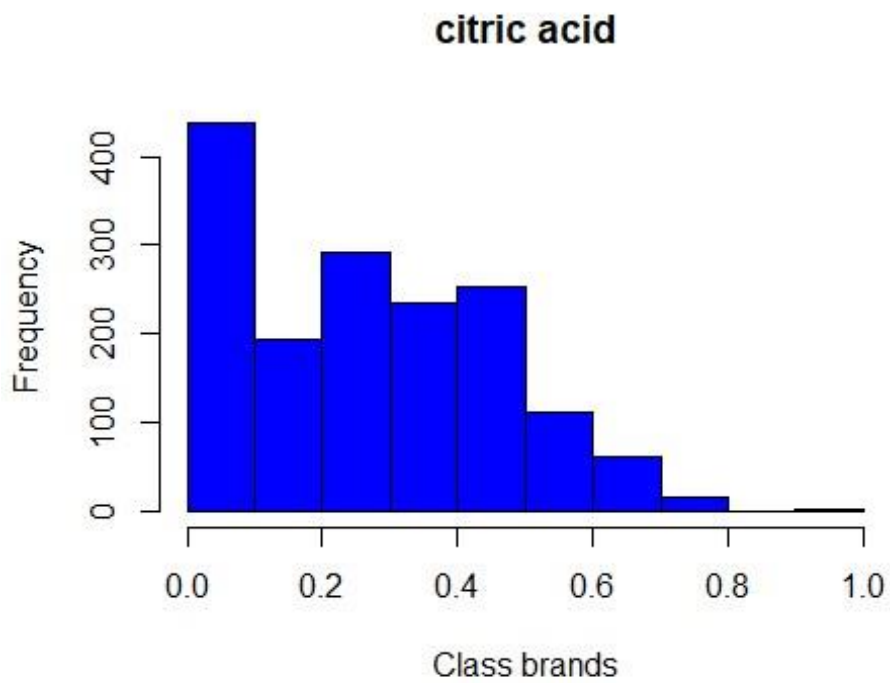
```
# Histograma de frecuencias absolutas de las variables fisicoquímicas
hist(winequality.red$fixed.acidity, main="fixed acidity", xlab="Class
brands", ylab="Frequency", col="blue")
```



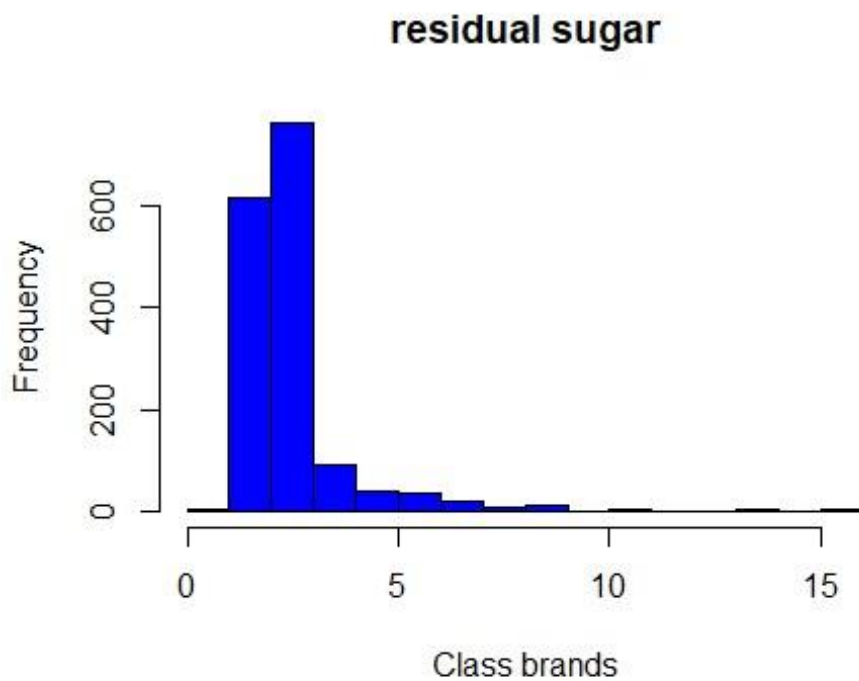
```
hist(winequality.red$volatile.acidity, main="volatile acidity",  
xlab="Class brands", ylab="Frequency", col="blue")
```



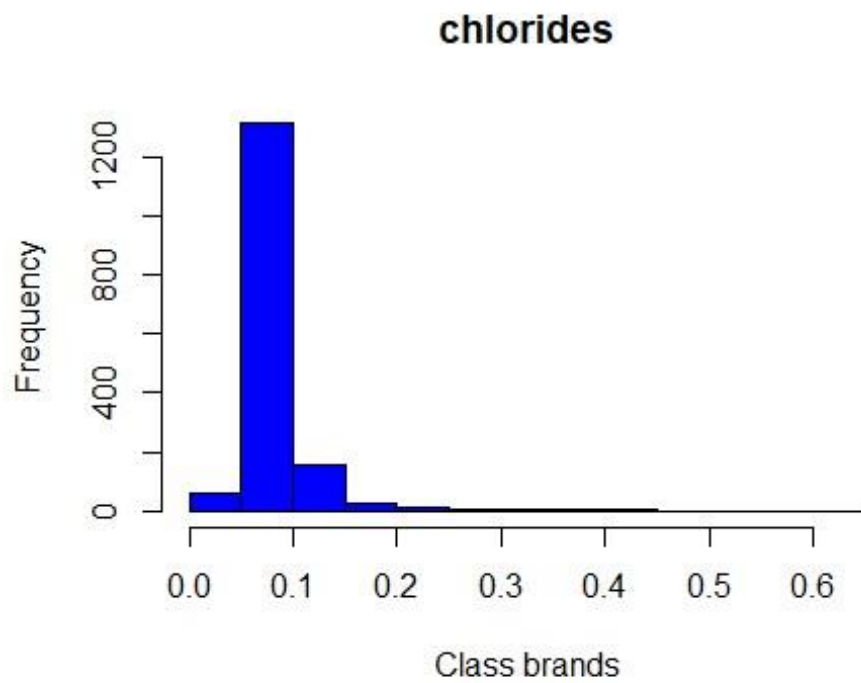
```
hist(winequality.red$citric.acid, main="citric acid", xlab="Class  
brands", ylab="Frequency", col="blue")
```



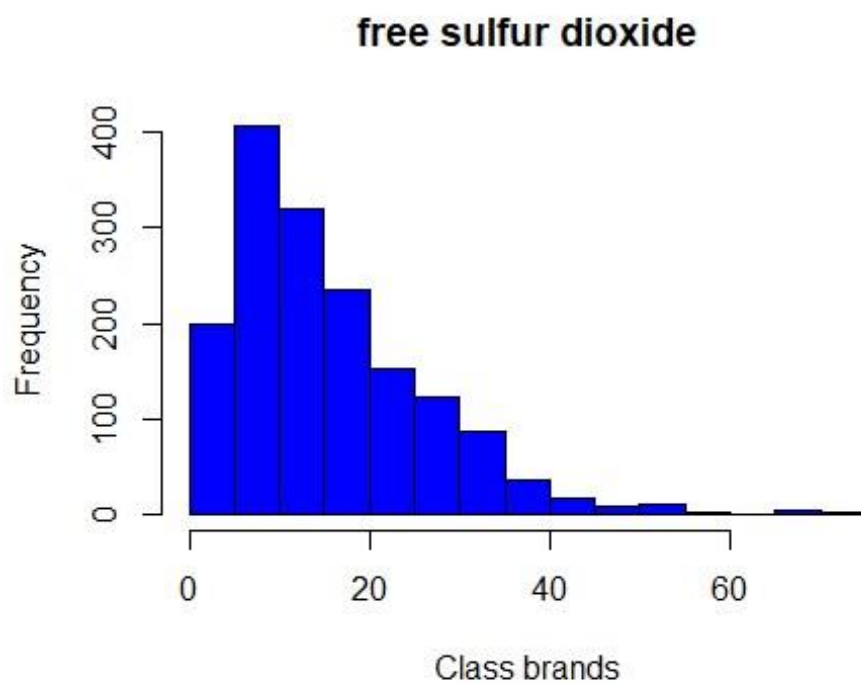
```
hist(winequality.red$residual.sugar, main="residual sugar", xlab="Class brands",  
      ylab="Frequency", col="blue")
```



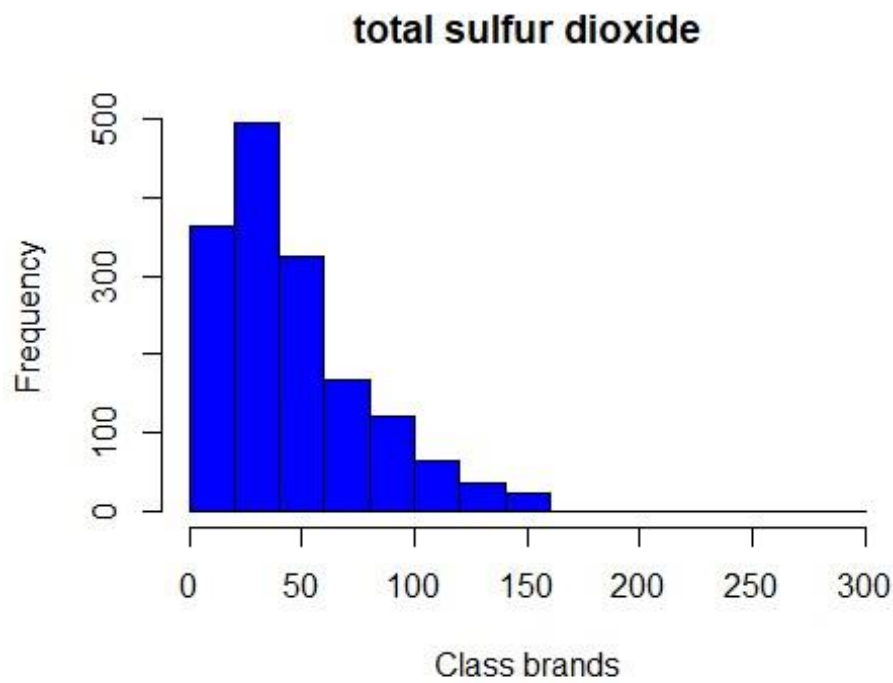
```
hist(winequality.red$chlorides, main="chlorides", xlab="Class brands",  
      ylab="Frequency", col="blue")
```



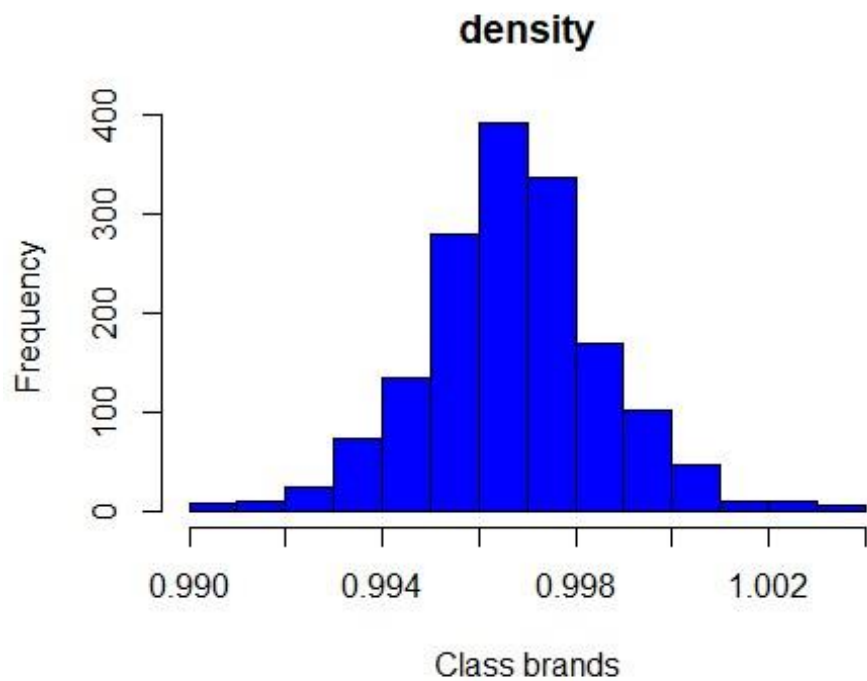
```
hist(winequality.red$free.sulfur.dioxide, main="free sulfur dioxide",  
xlab="Class brands", ylab="Frequency", col="blue")
```



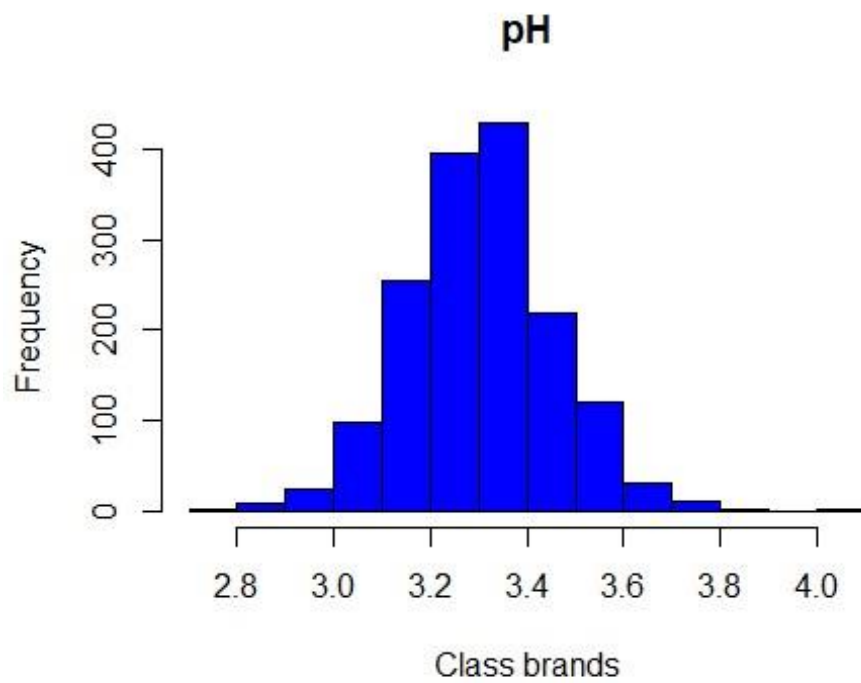
```
hist(winequality.red$total.sulfur.dioxide, main="total sulfur dioxide",  
xlab="Class brands", ylab="Frequency", col="blue")
```



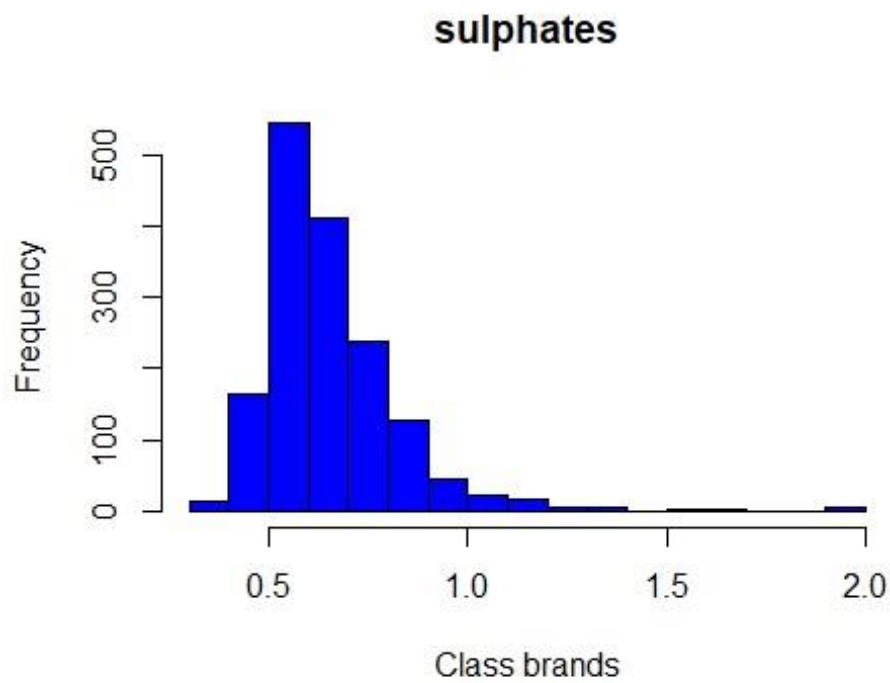
```
hist(winequality.red$density, main="density", xlab="Class brands",  
ylab="Frequency", col="blue")
```



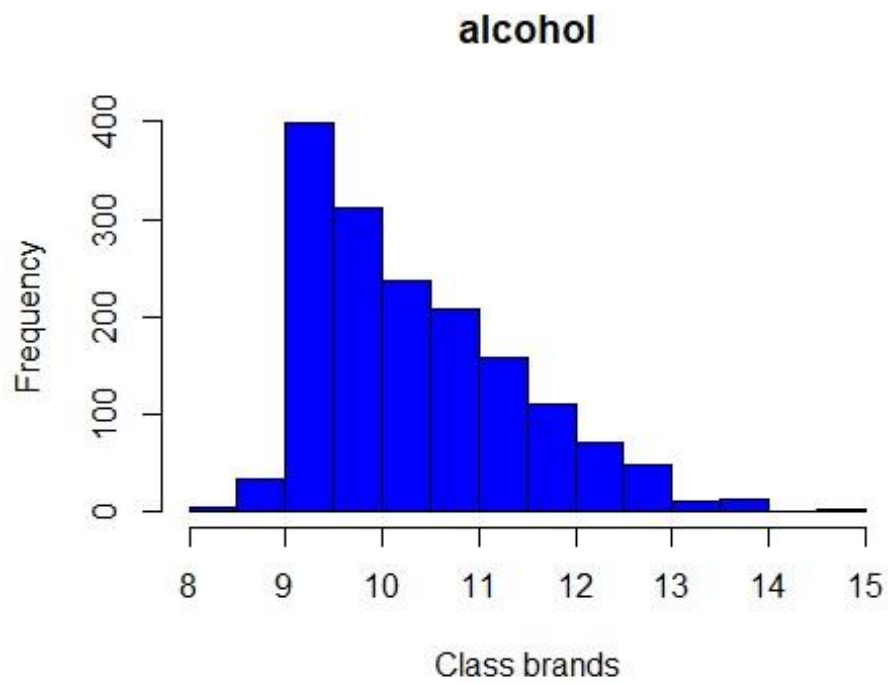
```
hist(winequality.red$pH, main="pH", xlab="Class brands",  
ylab="Frequency", col="blue")
```



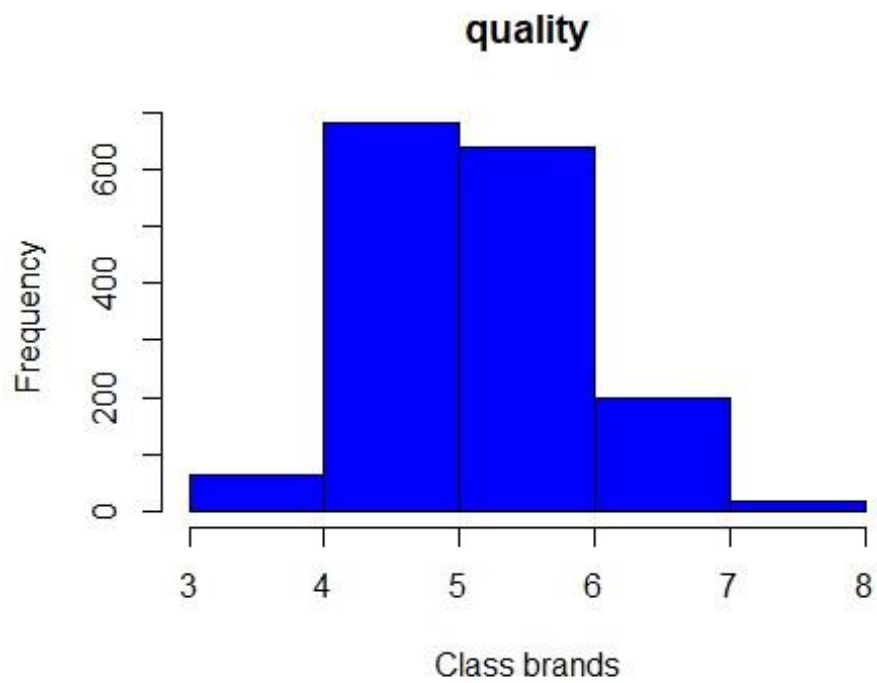
```
hist(winequality.red$sulphates, main="sulphates", xlab="Class brands",  
ylab="Frequency", col="blue")
```



```
hist(winequality.red$alcohol, main="alcohol", xlab="Class brands",  
ylab="Frequency", col="blue")
```



```
hist(winequality.red$quality, main="quality", xlab="Class brands",  
ylab="Frequency", col="blue", breaks = 5)
```



Integración y selección de los datos de interés

En esta sección, eliminamos columnas innecesarias o redundantes y fijamos el número de cifras decimales que deben contemplar. Se establece la columna "quality" de tipo numérico para facilitar cálculos y resultados posteriores.

```
# 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

# 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"
```

Detección de ceros y elementos vacíos por campo

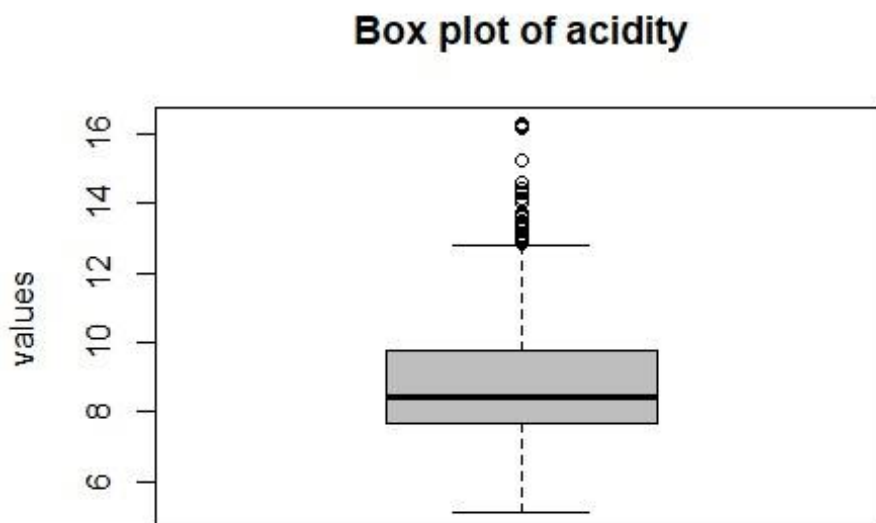
En esta sección, se lleva a cabo la detección de ceros y elementos vacíos por campo

```
# 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
```

Valores extremos

Identificamos outliers de cada variable fisicoquímica mediante diagramas de caja y usando la función `boxplots.stats()` de R.

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

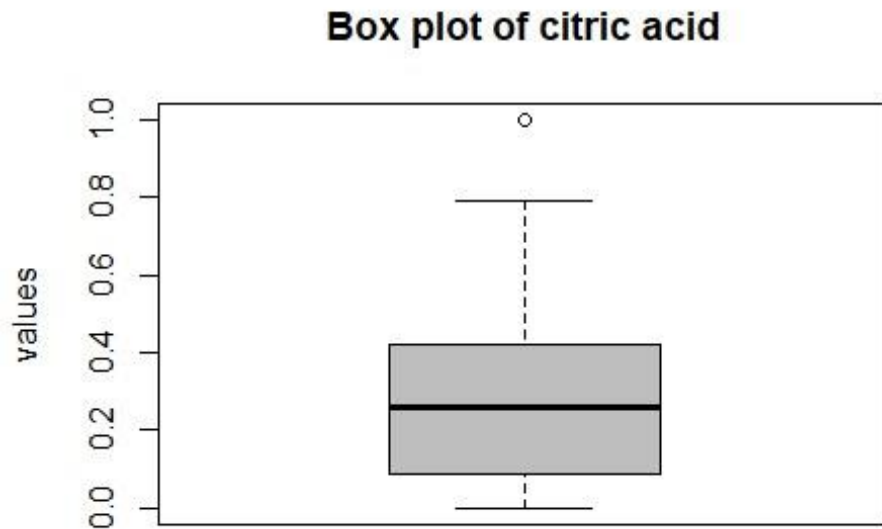


```

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")

```



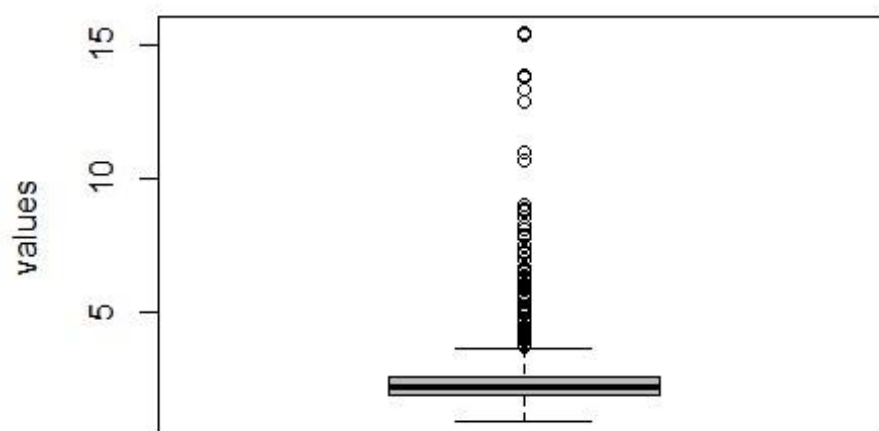
```

boxplot.stats(winequality.red$citric.acid)$out
## [1] 1

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

```

Box plot of residual sugar

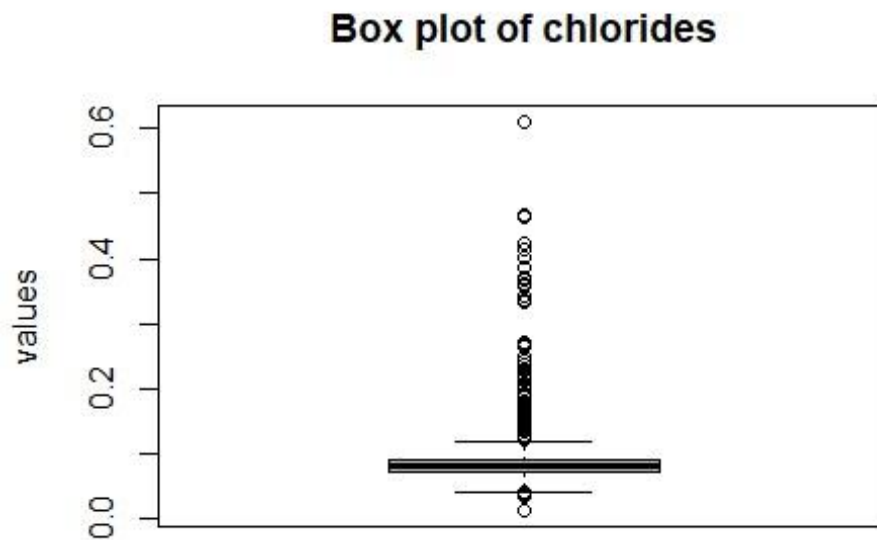


```
boxplot.stats(winequality.red$residual.sugar)$out
```

```
## [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")
```

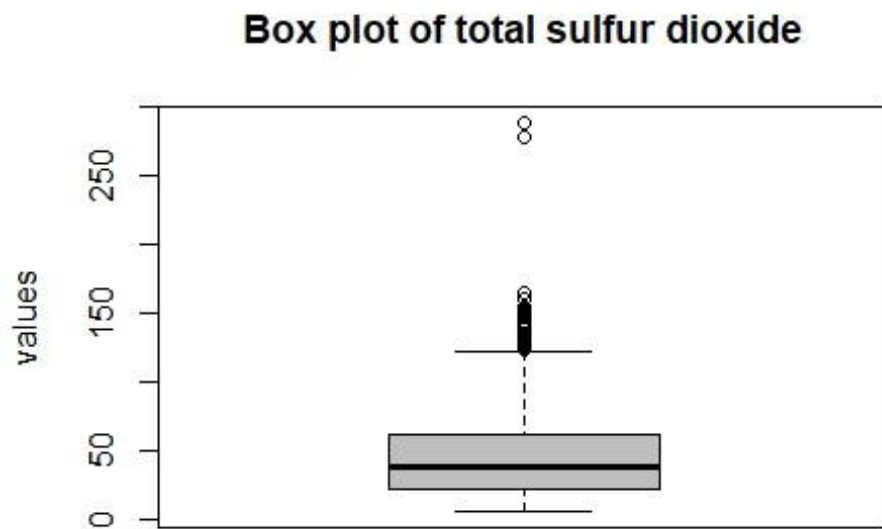


```
boxplot.stats(winequality.red$chlorides)$out
```

```
## [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
```

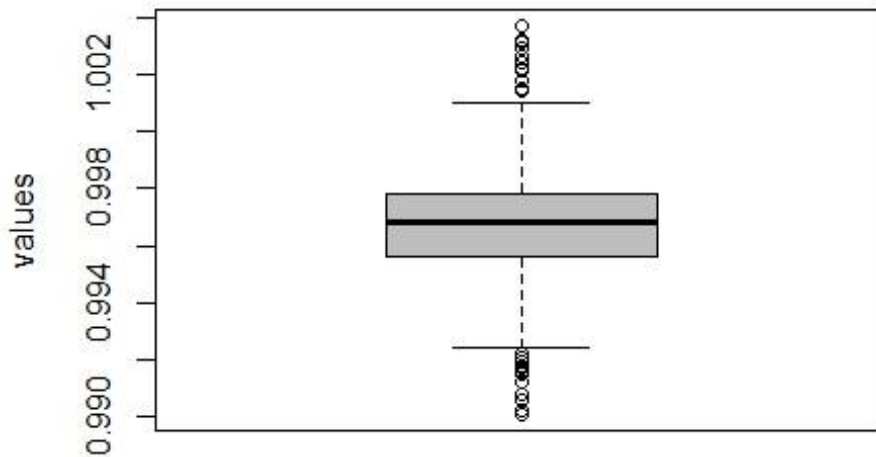
```
0.415
## [100] 0.153 0.415 0.267 0.123 0.214 0.214 0.169 0.205 0.205 0.039
0.235
## [111] 0.230 0.038

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")
```

Box plot of density



```
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
```

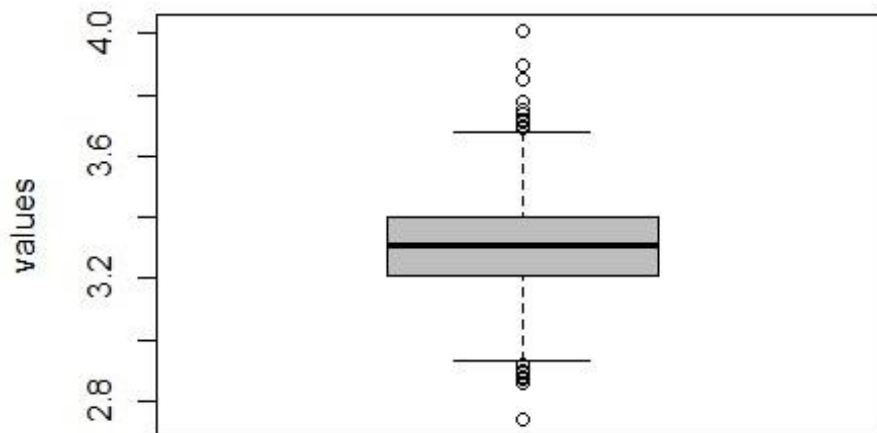
```
## [31] 0.9901 0.9901 0.9902 0.9922 0.9915 0.9916 0.9908 0.9908 0.9919  
1.0037
```

```
## [41] 1.0037 1.0024 0.9918 1.0024 0.9918
```

```
boxplot(winequality.red$pH,main="Box plot of pH",
```

```
col="gray",ylab="values")
```

Box plot of pH



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

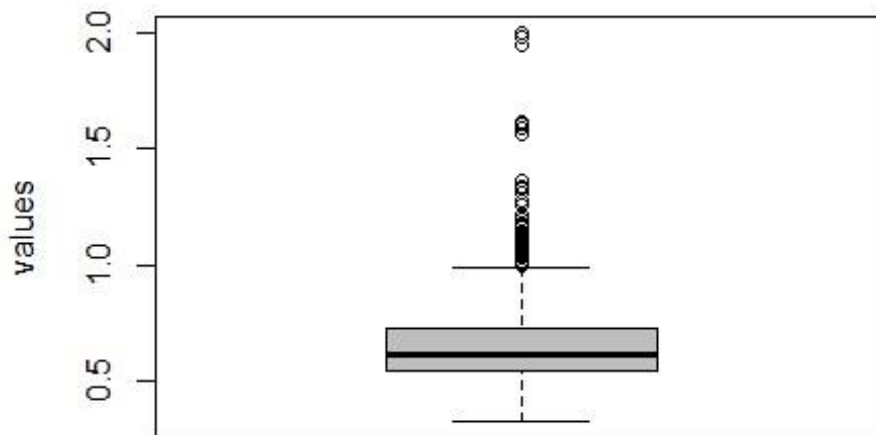
```
## [1] 3.90 3.75 3.85 2.74 3.69 3.69 2.88 2.86 3.74 2.92 2.92 2.92 3.72  
2.87
```

```
## [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")
```

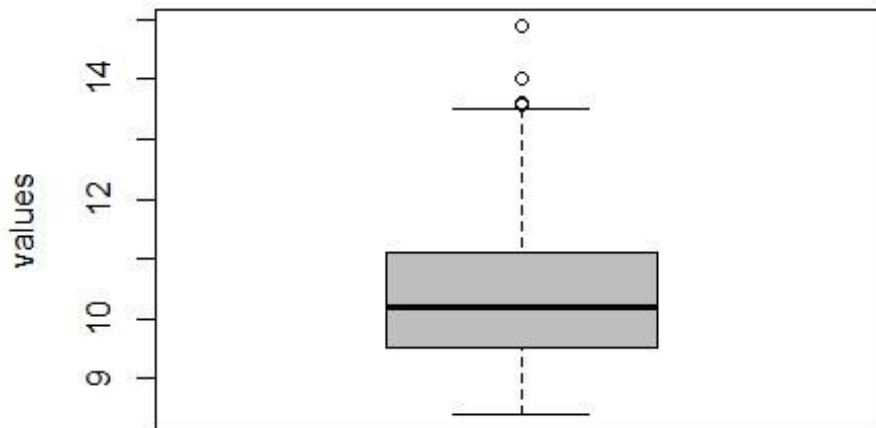

Box plot of sulphates



```
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")
```

Box plot of alcohol



```
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

# 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.dioxide),]

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),]

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

## [1] 10

nrow(winequality.red)

## [1] 1182

```

Exportación de los datos preprocesados

Una vez limpiados los datos, los guardamos en un fichero llamado winequalityred_data_clean.csv

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

Análisis de resultados

Selección de los grupos de datos que se quieren comparar

Establecemos grupos dentro del conjunto de datos para posteriores análisis y comparaciones.

```

# 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 cantidad 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),]

```

Pruebas de normalidad y homogeneidad de la varianza

Pruebas de normalidad de Anderson-Darling

```

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

```

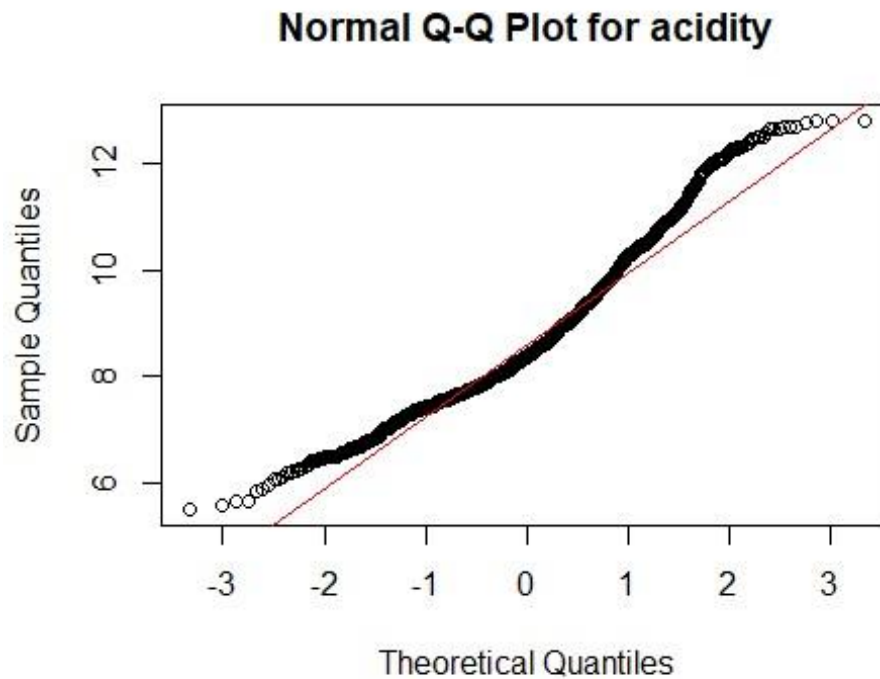
Test de normalidad Shapiro-Wilk y gráficos Q-Q

```

qqnorm(winequality.red$acidity, main = "Normal Q-Q Plot for acidity")

```

```
qqline(winequality.red$acidity, col = "red")
```



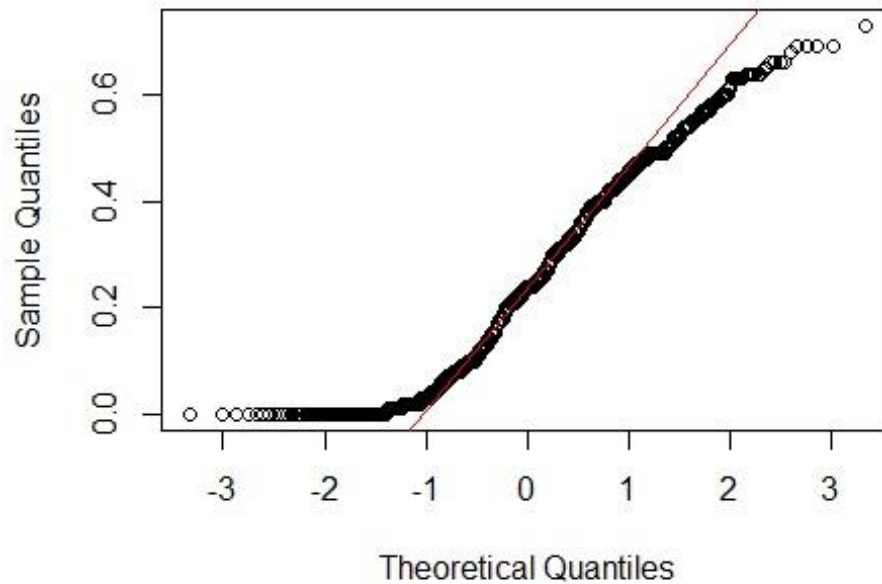
```
shapiro.test(winequality.red$acidity)
```

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

```
qqnorm(winequality.red$citric.acid, main = "Normal Q-Q Plot for citric  
acid")
```

```
qqline(winequality.red$citric.acid, col = "red")
```

Normal Q-Q Plot for citric acid



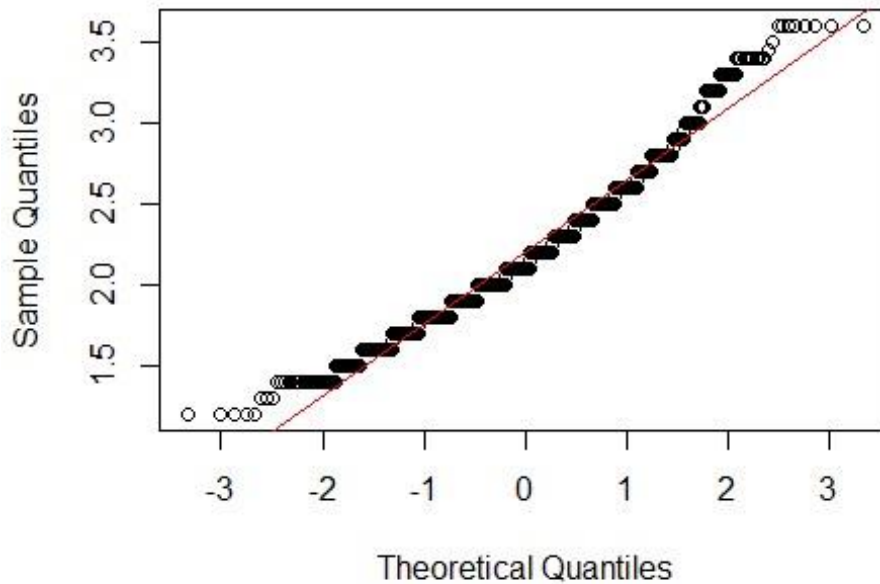
```
shapiro.test(winequality.red$citric.acid)
```

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

```
qqnorm(winequality.red$residual.sugar, main = "Normal Q-Q Plot for  
redidual sugar")
```

```
qqline(winequality.red$residual.sugar, col = "red")
```

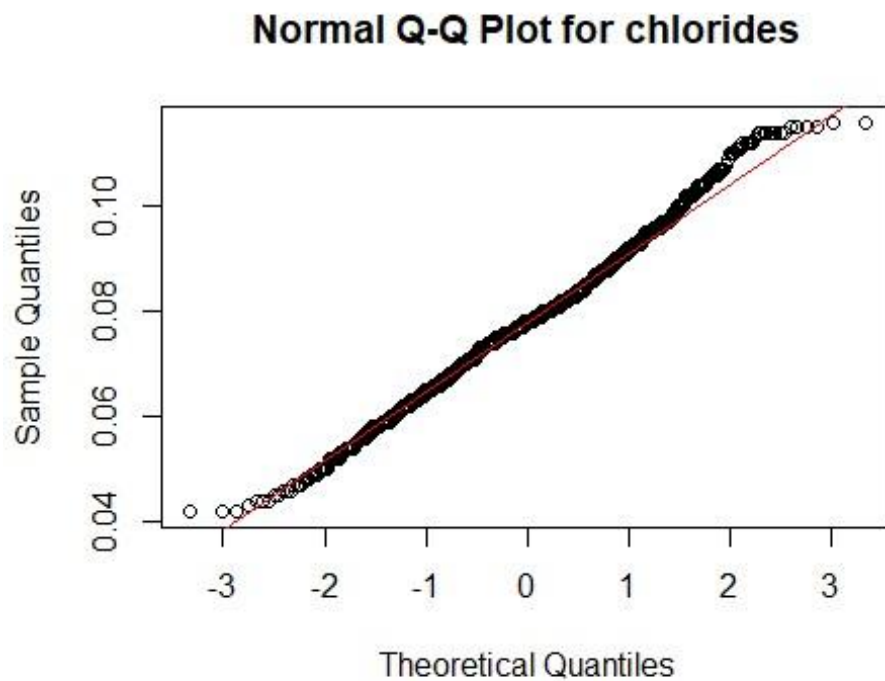
Normal Q-Q Plot for residual sugar



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

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

qqnorm(winequality.red$chlorides, main = "Normal Q-Q Plot for chlorides")
qqline(winequality.red$chlorides, col = "red")
```



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

```
##
```

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

```
##
```

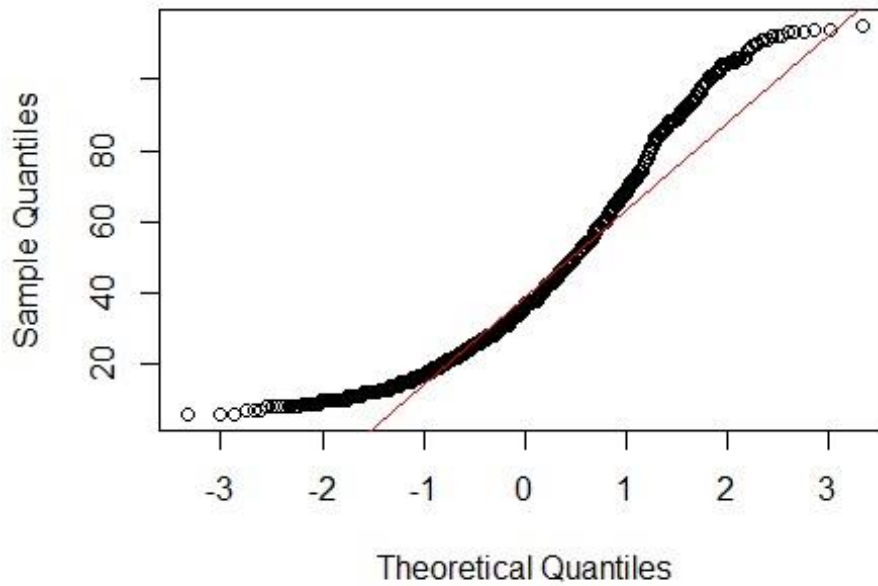
```
## data: winequality.red$chlorides ##
```

```
W = 0.99382, p-value = 8.049e-05
```

```
qqnorm(winequality.red$total.sulfur.dioxide, main = "Normal Q-Q Plot for  
total sulfur dioxide")
```

```
qqline(winequality.red$total.sulfur.dioxide, col = "red")
```

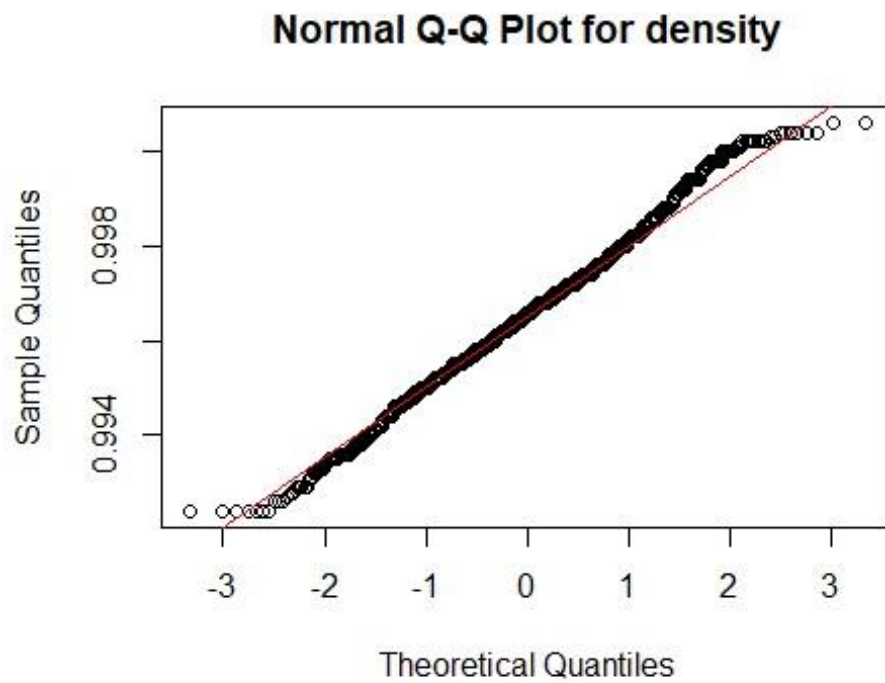

Normal Q-Q Plot for total sulfur dioxide



```
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

qqnorm(winequality.red$density, main = "Normal Q-Q Plot for density")
qqline(winequality.red$density, col = "red")
```

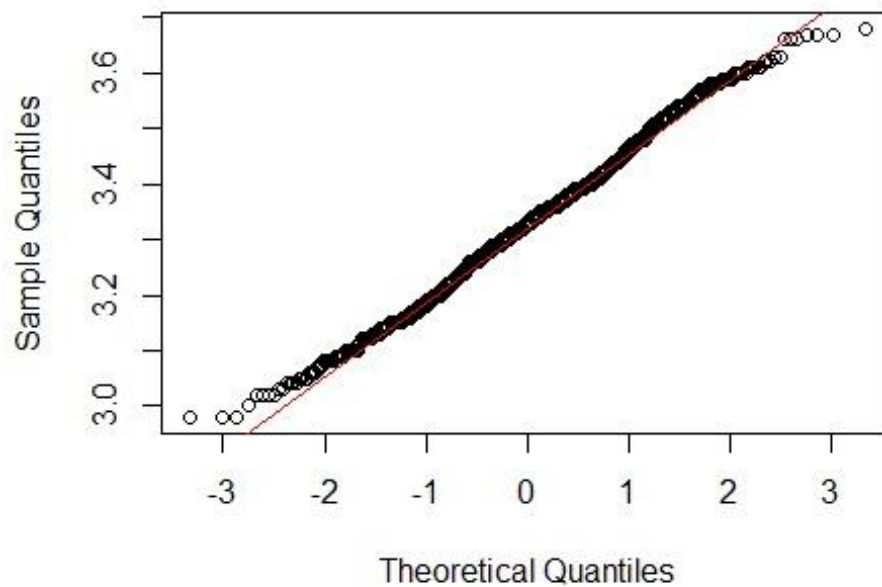


```
shapiro.test(winequality.red$density)

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

qqnorm(winequality.red$pH, main = "Normal Q-Q Plot for pH")
qqline(winequality.red$pH, col = "red")
```

Normal Q-Q Plot for pH



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

```
##
```

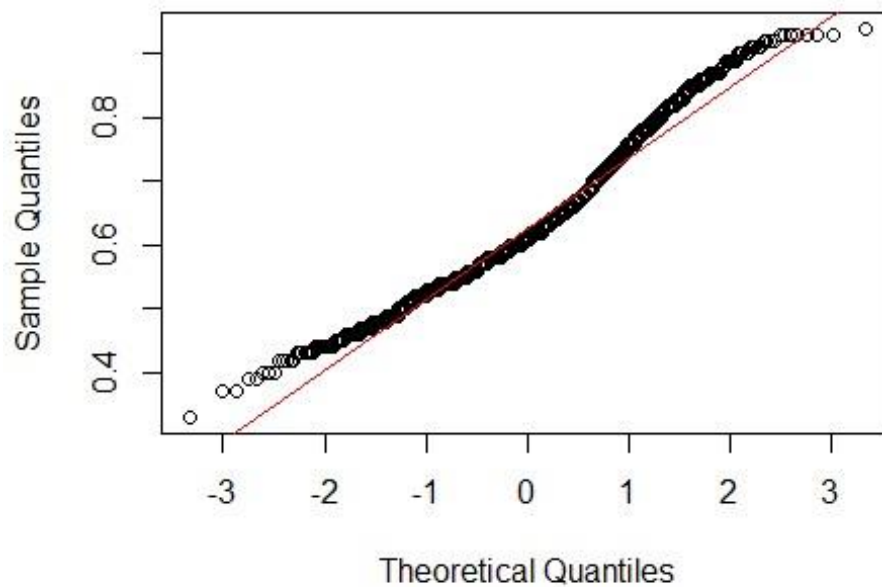
```
## Shapiro-Wilk normality test ##
```

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

```
qqnorm(winequality.red$sulphates, main = "Normal Q-Q Plot for sulphates")
```

```
qqline(winequality.red$sulphates, col = "red")
```

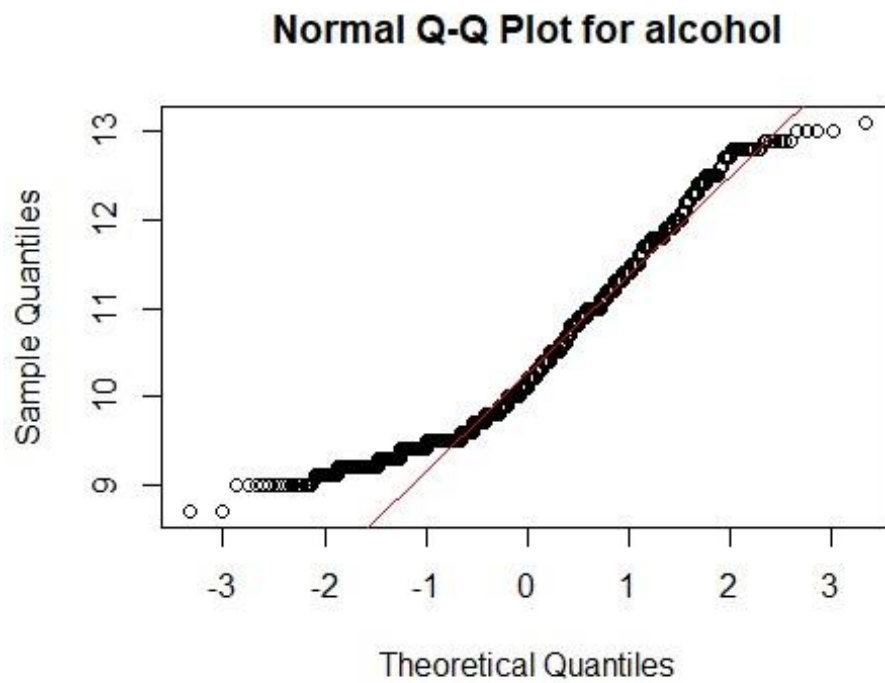
Normal Q-Q Plot for sulphates



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

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

qqnorm(winequality.red$alcohol, main = "Normal Q-Q Plot for alcohol")
qqline(winequality.red$alcohol, col = "red")
```

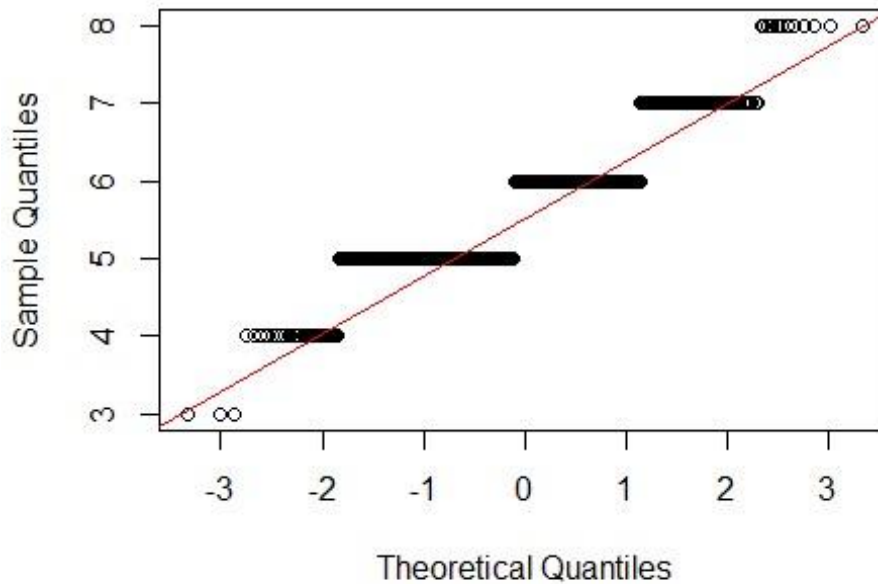


```
shapiro.test(winequality.red$alcohol)

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

qqnorm(winequality.red$quality, main = "Normal Q-Q Plot for quality")
qqline(winequality.red$quality, col = "red")
```

Normal Q-Q Plot for quality



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

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

Estudio de la homogeneidad de las varianzas. Test de Fligner-Kileen

```
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

```

Pruebas estadísticas

Influencia de las variables fisicoquímicas en la calidad de los vinos

```

corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable fisicoquimica
# 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]
}
}

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[,

```

```
## length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[, ##
length(winequality.red)], : Cannot compute exact p-value with ties

## Warning in cor.test.default(winequality.red[, i], winequality.red[,
## length(winequality.red)], : Cannot compute exact p-value with ties

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
```


Matriz de correlación entre variables

```
library(PerformanceAnalytics)

## Loading required package: xts ##

Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##      first, last
```

```

## The following objects are masked from 'package:data.table':
##
##     first, last
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:gplots':
##
##     textplot
##
## The following object is masked from 'package:graphics':
##
##     legend

```

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", "density", "pH", "sulphates", "alcohol", "quality")
cor(data)

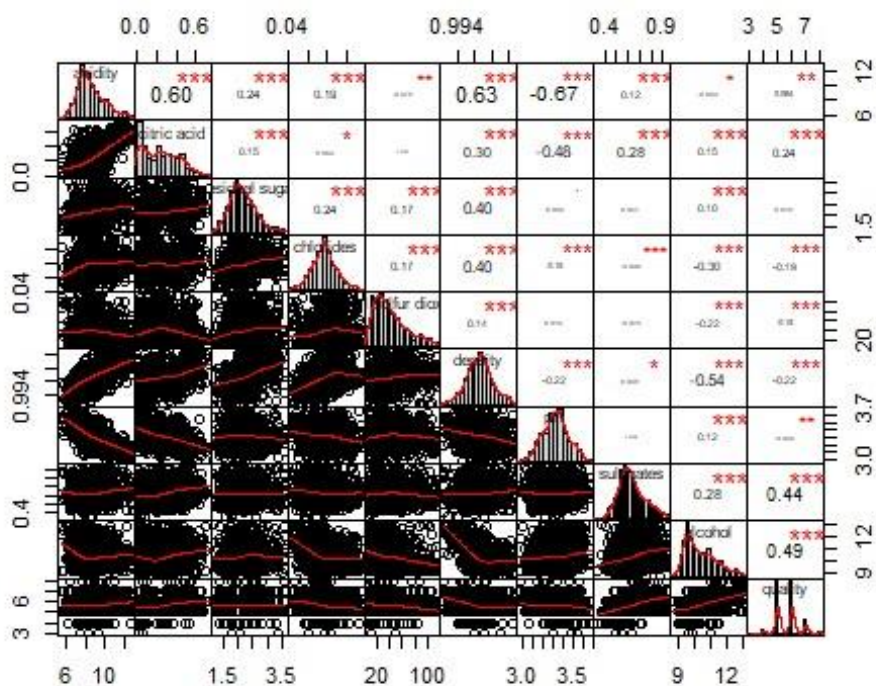
```

	acidity	citric acid	residual sugar	chlorides
acidity	1.00000000	0.603353709	0.24484897	0.19254182
citric acid	0.60335371	1.00000000	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  1.000000000 -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)

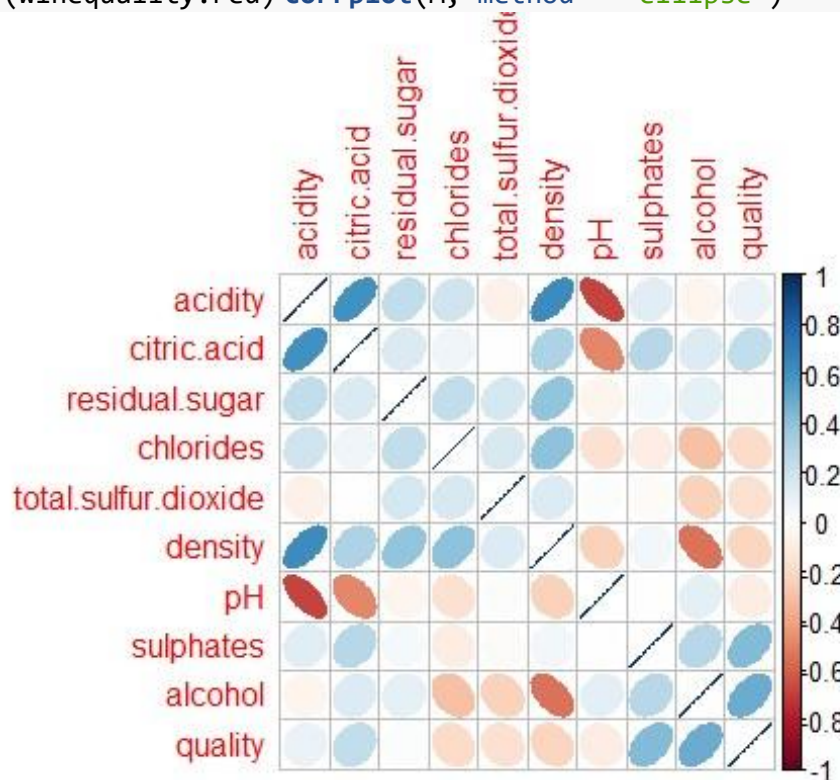
```



```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

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



Contrastes de hipótesis

¿La calidad de los vinos con densidad inferior a la media supera la de los vinos con densidad por encima de la media?

```
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

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
```

¿La calidad de los vinos con menos sal es igual o diferente a la de los vinos más salados?

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

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
```

Modelo de regresión lineal

Modelo de regresión multilíneal para predecir la calidad

```

# 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 = winequality.red)
modelo4 <- lm(quality ~ alcohol + sulphates + citric.acid + density +
total.sulfur.dioxide, data = winequality.red)

# 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
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
```

Predicción de la calidad con el modelo de regresión lineal

```
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
```

Modelo de regresión multilineal para predecir la acidez

```
# 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

```

Predecimos la acidez para unos valores de ácido cítrico, densidad y pH

```

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

##      1
## 7.438701

```

Modelo de regresión logístico

```

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

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE ## [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE

winequality.red$density[winequality.red$density < 1]<-0
high.density<-winequality.red$density
high.density<-
factor(high.density) # Variables explicativas de La
densidad acidity<-winequality.red$acidity alcohol<-
winequality.red$alcohol

```



```

residual.sugar<-winequality.red$residual.sugar chlorides<-
winequality.red$chlorides
# Estimación del modelo de regresión Logística
reglog <- glm(high.density ~ acidity+alcohol+residual.sugar+chlorides,
data = winequality.red, family = 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

# 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")

##           1
## 0.4041297

```

Tabla resumen de los datos preprocesados y representación en forma de boxplots

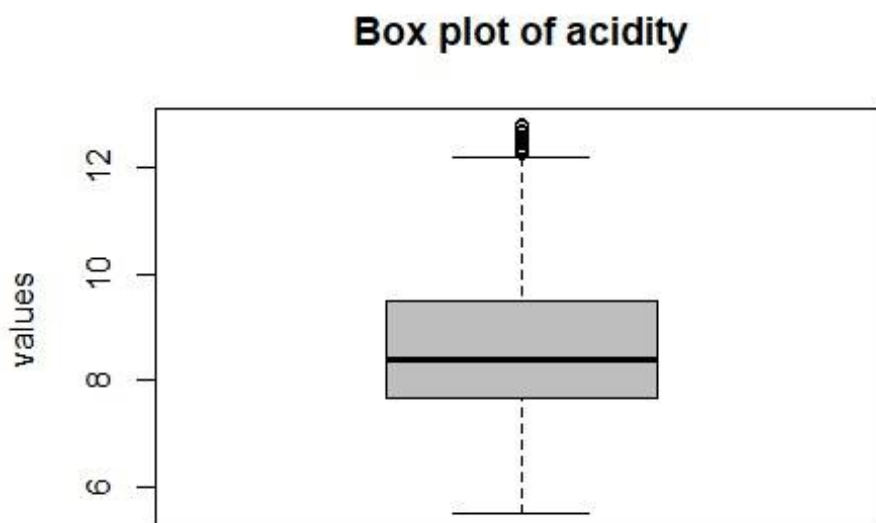
```

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

```

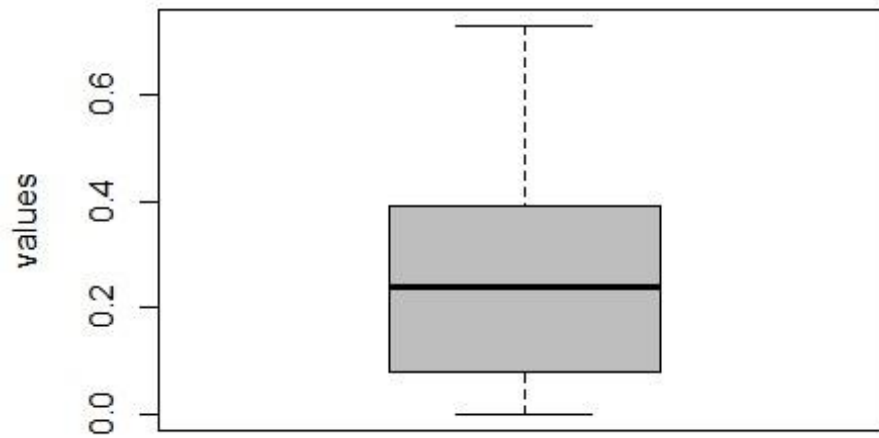
```
##      acidity      citric.acid      residual.sugar      chlorides
## Min.   : 5.520    Min.   :0.0000    Min.   :1.200    Min.   :0.04200
## 1st Qu.: 7.680    1st Qu.:0.0800    1st Qu.:1.900    1st Qu.:0.06900
## Median : 8.380    Median :0.2400    Median :2.100    Median :0.07800
## Mean   : 8.681    Mean   :0.2459    Mean   :2.183    Mean   :0.07817
## 3rd Qu.: 9.498    3rd Qu.:0.3900    3rd Qu.:2.500    3rd Qu.:0.08675
## Max.   :12.800    Max.   :0.7300    Max.   :3.600    Max.   :0.11600
## total.sulfur.dioxide  density      pH      sulphates
## Min.   : 6.00      Min.   :0.00000    Min.   :2.980    Min.   :0.3300
## 1st Qu.: 22.00     1st Qu.:0.00000    1st Qu.:3.230    1st Qu.:0.5500
## Median : 36.00     Median :0.00000    Median :3.330    Median :0.6100
## Mean   : 41.79     Mean   :0.02708    Mean   :3.326    Mean   :0.6294
## 3rd Qu.: 55.00     3rd Qu.:0.00000    3rd Qu.:3.410    3rd Qu.:0.7000
## Max.   :115.00     Max.   :1.00060    Max.   :3.680    Max.   :0.9400
##      alcohol      quality
## Min.   : 8.70     Min.   :3.000
## 1st Qu.: 9.50     1st Qu.:5.000
## Median :10.10     Median :6.000
## Mean   :10.37     Mean   :5.641
## 3rd Qu.:11.00     3rd Qu.:6.000    ##
Max.   :13.10     Max.   :8.000
```

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



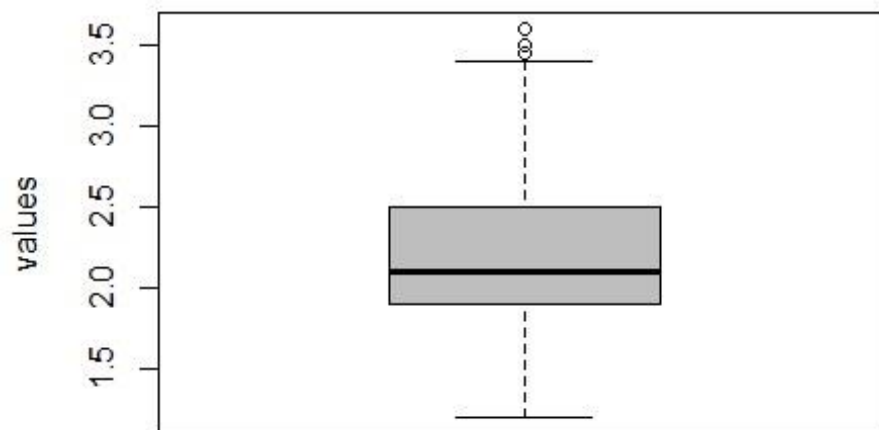
```
boxplot(winequality.red$citric.acid,main="Box plot of citric acid",
col="gray",ylab="values")
```

Box plot of citric acid



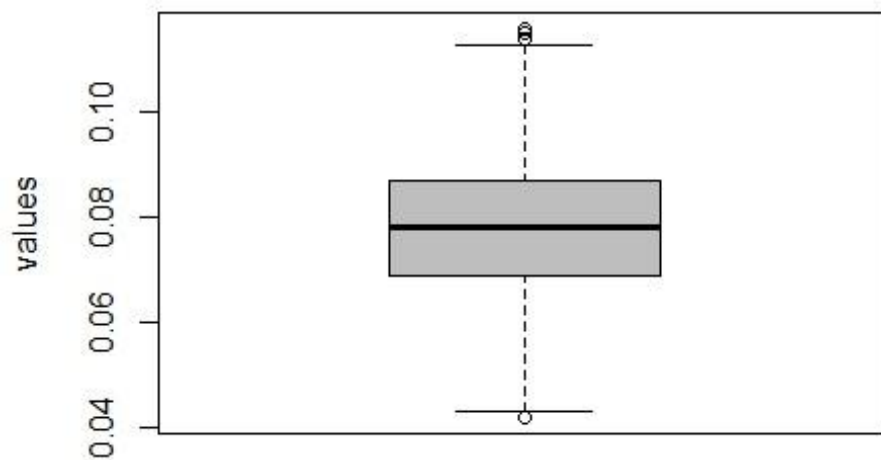
```
boxplot(winequality.red$residual.sugar,main="Box plot of residual sugar",  
col="gray",ylab="values")
```

Box plot of residual sugar



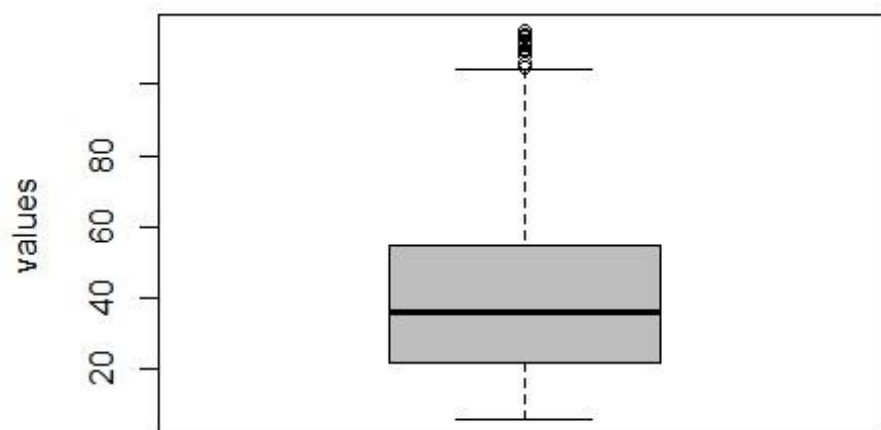
```
boxplot(winequality.red$chlorides,main="Box plot of chlorides",  
col="gray",ylab="values")
```

Box plot of chlorides

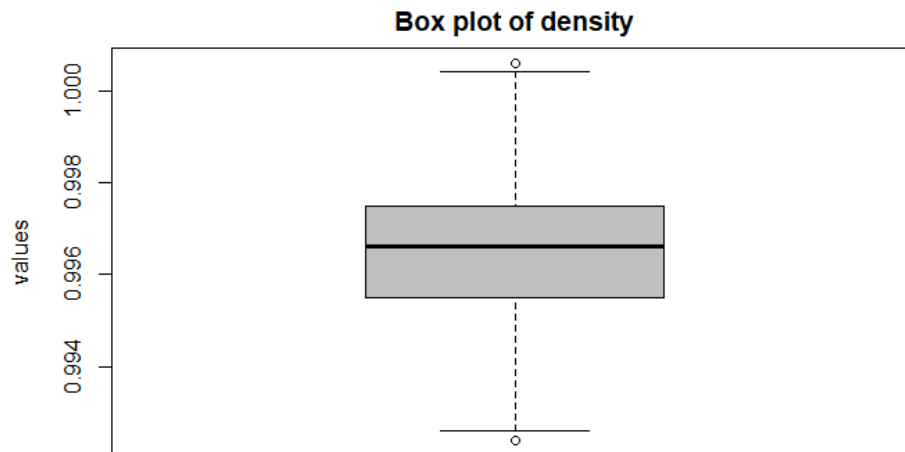


```
boxplot(winequality.red$total.sulfur.dioxide,main="Box plot of total  
sulfur dioxide", col="gray",ylab="values")
```

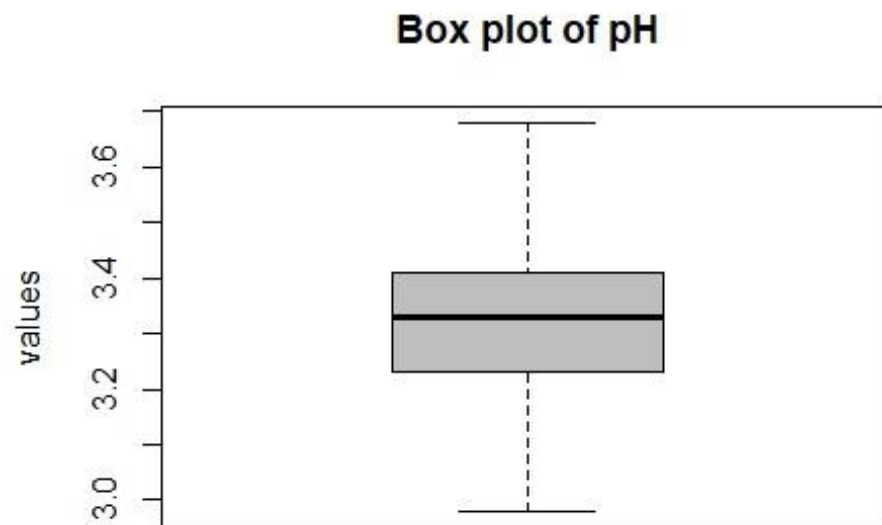
Box plot of total sulfur dioxide



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

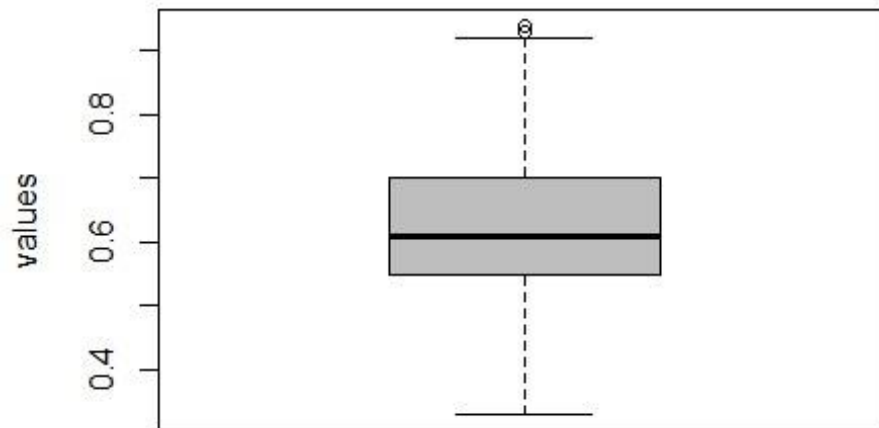


```
boxplot(winequality.red$pH,main="Box plot of pH",  
col="gray",ylab="values")
```



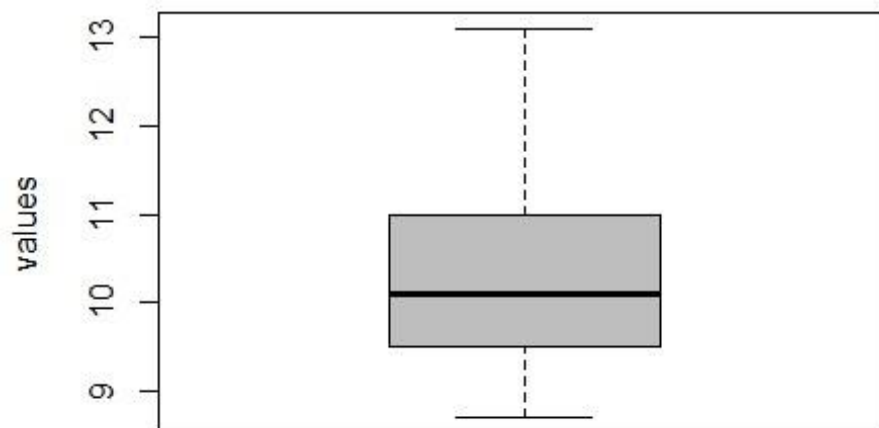
```
boxplot(winequality.red$sulphates,main="Box plot of sulphates",  
col="gray",ylab="values")
```

Box plot of sulphates



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

Box plot of alcohol



Referencias

Squire, Megan (2015). Clean Data. Packt Publishing Ltd.

Jiawei Han, Micheline 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/>)