

Práctica 2: Limpieza y validación de los datos

Tipología y ciclo de vida de los datos

Manuel Cerezo y Alfredo Delsors

10 junio 2019

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

Para llevar a cabo la descripción de Dataset importamos el csv con los datos de los vinos tintos de la página <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>.

```
# Carga de datos
redwine <- read.csv("winequality-red.csv",header=TRUE, sep=";")
```

Comprobamos que los tipos de datos asignados por R se corresponden a los indicados en <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality.names>

```
#Tipo de dato asignado a cada campo
sapply(redwine, 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"

Observamos que los datos asignados por R son los especificados en el dataset, 11 variables numéricas y una entera, que se describen a continuación.

1.1 Descripción del conjunto de datos

El conjunto de datos está constituido por 11 variables de tipo numérico con cada una de las características del vino tinto y una variable independiente con el resultado de la calidad de éste.

El dataset esta compuesto por 1599 registros que corresponden a 1599 vinos tintos diferentes, con las siguientes variables de entrada (basadas en pruebas fisicoquímicas):

- acidez fija
- acidez volatil
- ácido cítrico
- azucar residual
- cloruros
- dióxido de sulfuro libre
- dióxido de sulfuro total
- densidad
- pH
- sulfatos
- alcohol

Y la siguiente variable de salida (basada en datos sensoriales):

- calidad (puntuación entre 0 y 10)

1.1.1 Resumen de las variables

```
options(knitr.kable.NA = '')  
  
kable(summary(redwine)[,c(1,2,3,4,5,6)])
```

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
Min. : 4.60	Min. :0.1200	Min. :0.000	Min. : 0.900	Min. :0.01200	Min. : 1.00
1st Qu.: 7.10	1st Qu.:0.3900	1st Qu.:0.090	1st Qu.: 1.900	1st Qu.:0.07000	1st Qu.: 7.00
Median : 7.90	Median :0.5200	Median :0.260	Median : 2.200	Median :0.07900	Median :14.00
Mean : 8.32	Mean :0.5278	Mean :0.271	Mean : 2.539	Mean :0.08747	Mean :15.87
3rd Qu.: 9.20	3rd Qu.:0.6400	3rd Qu.:0.420	3rd Qu.: 2.600	3rd Qu.:0.09000	3rd Qu.:21.00
Max. :15.90	Max. :1.5800	Max. :1.000	Max. :15.500	Max. :0.61100	Max. :72.00

```
kable(summary(redwine)[,c(7,8,9,10,11,12)])
```

total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
Min. : 6.00	Min. :0.9901	Min. :2.740	Min. :0.3300	Min. : 8.40	Min. :3.000
1st Qu.: 22.00	1st Qu.:0.9956	1st Qu.:3.210	1st Qu.:0.5500	1st Qu.: 9.50	1st Qu.:5.000
Median : 38.00	Median :0.9968	Median :3.310	Median :0.6200	Median :10.20	Median :6.000
Mean : 46.47	Mean :0.9967	Mean :3.311	Mean :0.6581	Mean :10.42	Mean :5.636
3rd Qu.: 62.00	3rd Qu.:0.9978	3rd Qu.:3.400	3rd Qu.:0.7300	3rd Qu.:11.10	3rd Qu.:6.000
Max. :289.00	Max. :1.0037	Max. :4.010	Max. :2.0000	Max. :14.90	Max. :8.000

1.1.2 Valores nulos

```
sapply(redwine, function(x) sum(is.na(x)))
```

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

No hay valores nulos.

1.1.3 Valores perdidos

```
conperdidos <- !complete.cases(redwine)  
cat("Hay", length(which(conperdidos)), "observaciones con valores perdidos")  
  
## Hay 0 observaciones con valores perdidos
```

1.2 Importancia y objetivos de los análisis

A partir de este conjunto de datos se plantea la problemática de determinar qué variables son las que determinan la calidad de un vino tinto.

Además, se podrán crear modelos de regresión que permitirán predecir la calidad de un vino en función de sus características, así como realizar contrastes de hipótesis que ayuden a identificar propiedades interesantes en las muestras que puedan ser inferidas a partir de los datos.

Estos análisis adquieren una gran relevancia tanto en sectores vinícolas como en restaurantes o licorerías. Por ejemplo, las bodegas podrían conocer la calidad de un vino tinto a partir de sus características y ponerle precio, o los restaurantes, podrían sugerir un vino según el entrante seleccionado o elegir los vinos para su bodega.

1.3 Pregunta que se pretende responder en este análisis

¿Cuál es la calidad de un vino tinto, dadas sus características fisicoquímicas?

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

En este caso no descartaremos ninguna variable, ya que las usaremos todas para calcular la calidad del vino.

3. Limpieza de los datos.

3.1. Ceros y datos vacíos

En la descripción del dataset, la única variable con límite es la calidad, que estará entre 0 y 10, el resto no tienen límites definidos, por lo que no podremos descartar un registro que sea 0 por ser inválido.

Al tratarse de variables numéricas, podríamos descartar los valores nulos, pero como vimos en 1.1.2, no hay valores nulos en el conjunto de datos.

A continuación vamos a buscar los valores que son exactamente 0.

```
# Números de valores cero por variable
cerofixedacidity<- as.data.frame(table(redwine[,1]=="0"))
cerovolacity<- as.data.frame(table(redwine[,2]=="0"))
cerocitricacid<- as.data.frame(table(redwine[,3]=="0"))
cerofressugar<- as.data.frame(table(redwine[,4]=="0"))
cerochlorides<- as.data.frame(table(redwine[,5]=="0"))
cerofreesulf<- as.data.frame(table(redwine[,6]=="0"))
cerototalsulf<- as.data.frame(table(redwine[,7]=="0"))
cerodensity<- as.data.frame(table(redwine[,8]=="0"))
ceroph<- as.data.frame(table(redwine[,9]=="0"))
cerosulph<- as.data.frame(table(redwine[,10]=="0"))
ceroalcohol<- as.data.frame(table(redwine[,11]=="0"))
ceroquality<- as.data.frame(table(redwine[,12]=="0"))
print(cerocitricacid)
```

```
##      Var1 Freq
## 1 FALSE 1467
## 2  TRUE  132
```

La unica variable que contiene 0 es la citric acid. Para resolverlo vamos a coger el valor medio de dicha variable y lo pondremos donde hayan valores 0

```
# Mediana de la variable citric acid
citridacidvalue<-median(redwine$citric.acid, na.rm = FALSE)
# cambio el valor 0 por el de la mediana
redwine$citric.acid[redwine$citric.acid == "0"] <- citridacidvalue
```

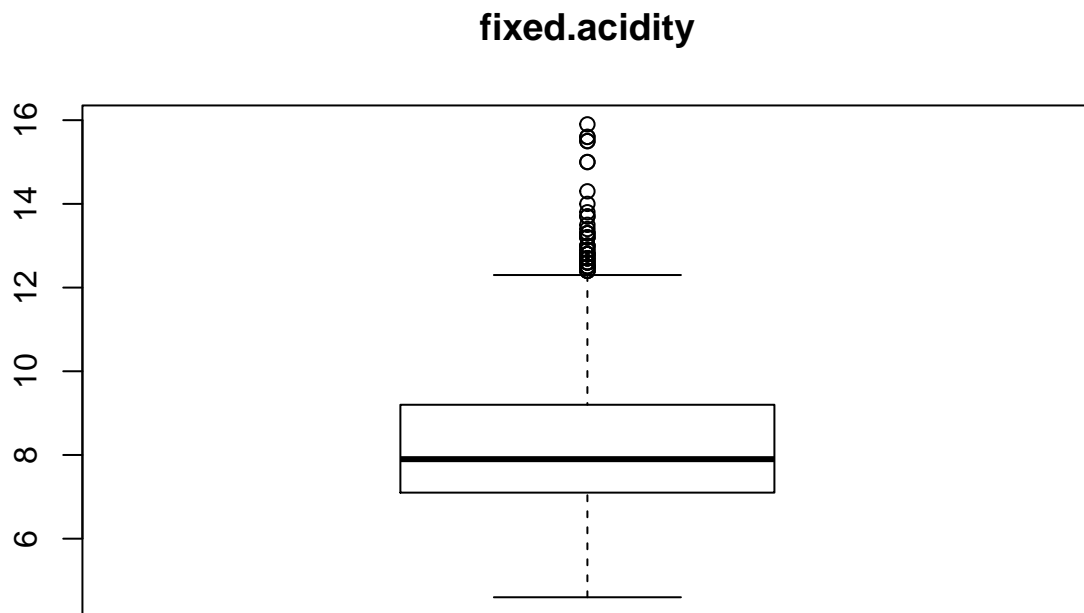
3.2. Identificación y tratamiento de valores extremos.

La manera mas sencilla de buscar valores extremos es mediante el diagrama de caja (boxplot). En este se puede ver los cuartiles, la media y los valores fuera de la 'normalidad'. Para identificarlos usaremos la función boxplot.stats:

```
#Boxplot fixed.acidity
boxplot.stats(redwine$fixed.acidity)$out
```

```
## [1] 12.8 12.8 15.0 15.0 12.5 13.3 13.4 12.4 12.5 13.8 13.5 12.6 12.5 12.8
## [15] 12.8 14.0 13.7 13.7 12.7 12.5 12.8 12.6 15.6 12.5 13.0 12.5 13.3 12.4
## [29] 12.5 12.9 14.3 12.4 15.5 15.5 15.6 13.0 12.7 13.0 12.7 12.4 12.7 13.2
## [43] 13.2 13.2 15.9 13.3 12.9 12.6 12.6
```

```
boxplotFacidity<-boxplot(redwine$fixed.acidity ,main="fixed.acidity", COL="gray")
```

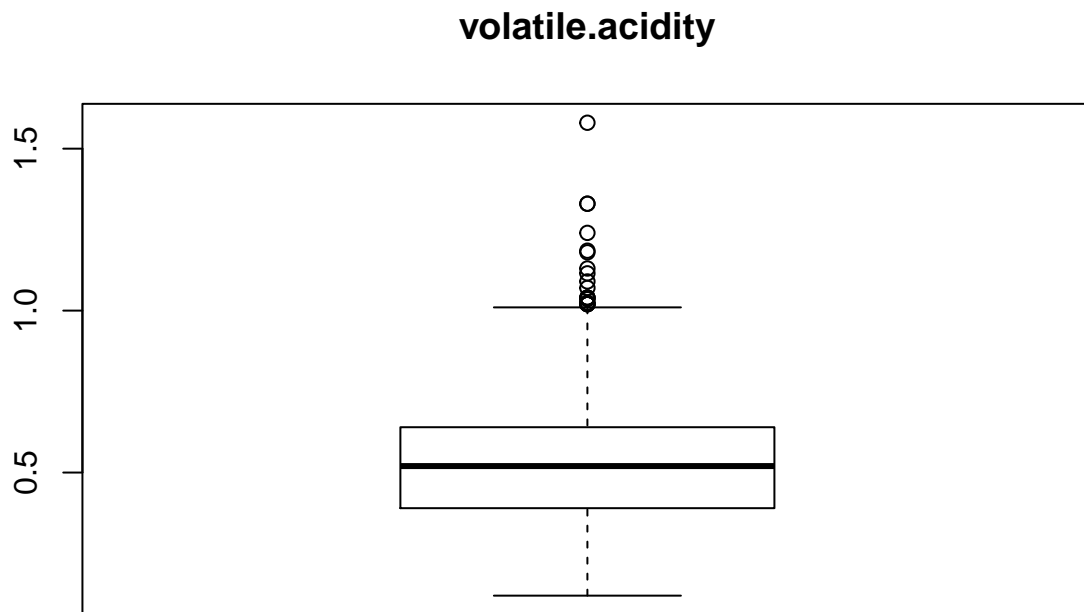


```
#Boxplot volatile.acidity
boxplot.stats(redwine$volatile.acidity)$out
```

```
## [1] 1.130 1.020 1.070 1.330 1.330 1.040 1.090 1.040 1.240 1.185 1.020
```

```
## [12] 1.035 1.025 1.115 1.020 1.020 1.580 1.180 1.040
```

```
boxplotV acidity<-boxplot(redwine$volatile.acidity ,main="volatile.acidity", COL="gray")
```



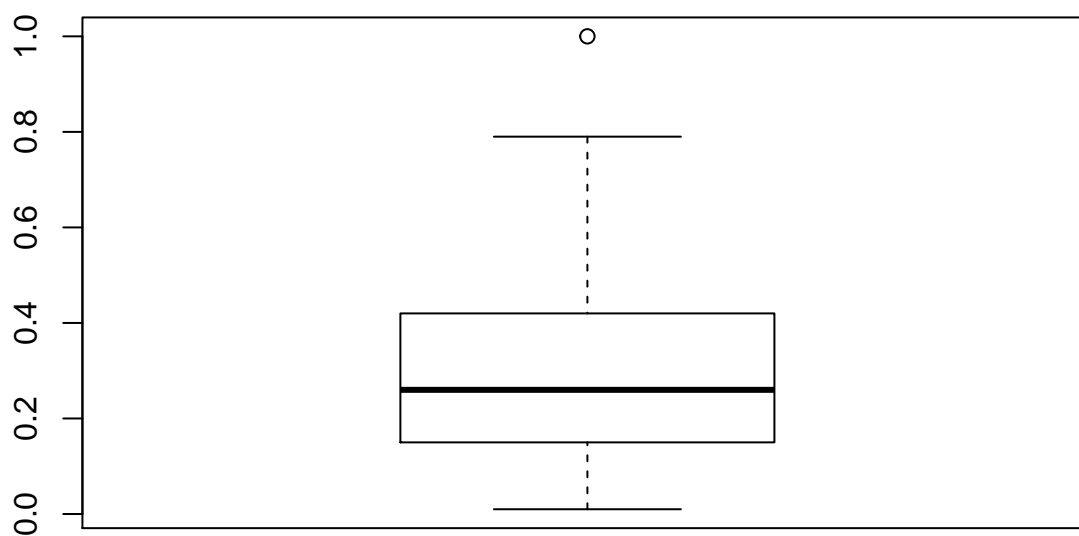
```
#Boxplot citric.acid
```

```
boxplot.stats(redwine$citric.acid)$out
```

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

```
boxplotcitric<-boxplot(redwine$citric.acid ,main="citric.acid", COL="gray")
```

citric.acid

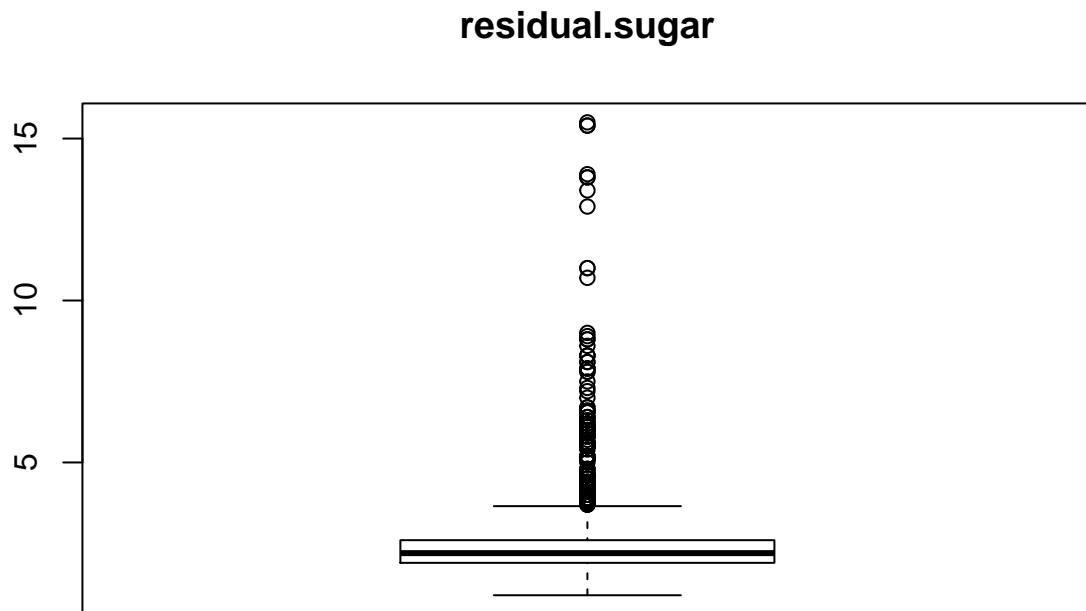


```
#Boxplot residual.sugar
```

```
boxplot.stats(redwine$residual.sugar)$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
```

```
boxplotRsugar<-boxplot(redwine$residual.sugar ,main="residual.sugar", COL="gray")
```



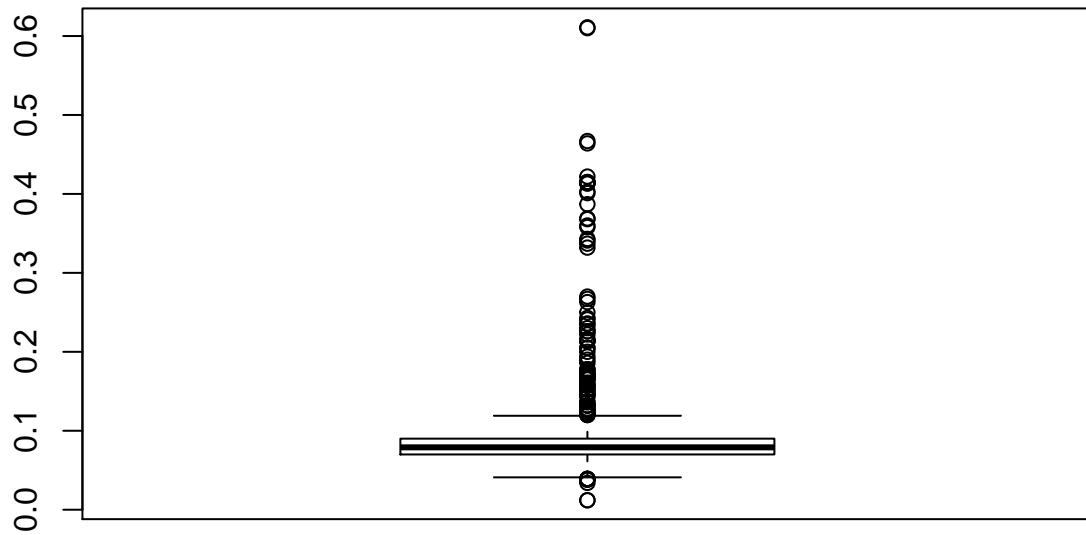
```
#Boxplot chlorides
```

```
boxplot.stats(redwine$chlorides)$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
```

```
boxplotchlorides<-boxplot(redwine$chlorides ,main="chlorides", COL="gray")
```

chlorides

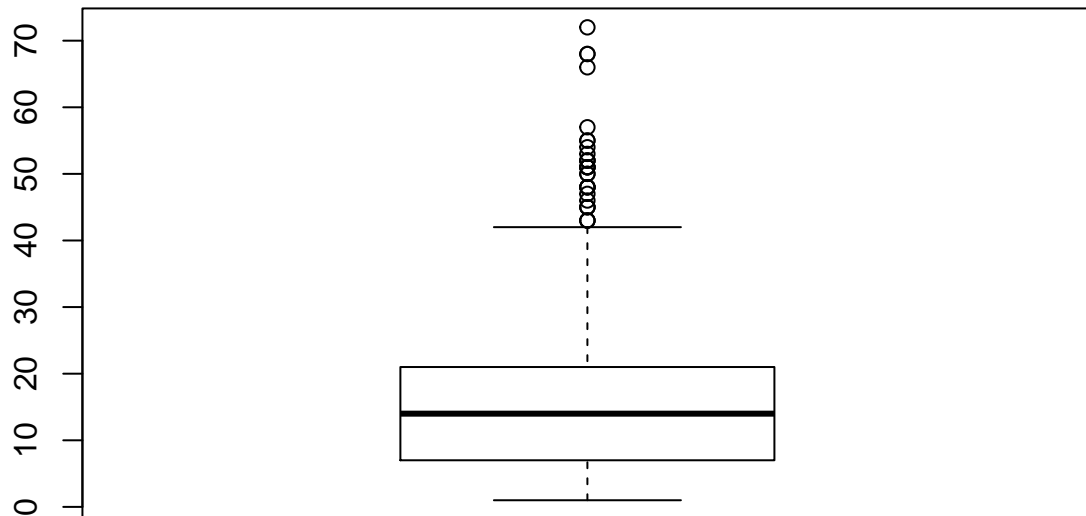


```
#Boxplot free.sulfur.dioxide
boxplot.stats(redwine$free.sulfur.dioxide)$out

## [1] 52 51 50 68 68 43 47 54 46 45 53 52 51 45 57 50 45 48 43 48 72 43 51
## [24] 51 52 55 55 48 48 66

boxplotFsulfur<-boxplot(redwine$free.sulfur.dioxide ,main="free.sulfur.dioxide", COL="gray")
```


free.sulfur.dioxide



```
#Boxplot total.sulfur.dioxide
```

```
boxplot.stats(redwine$total.sulfur.dioxide)$out
```

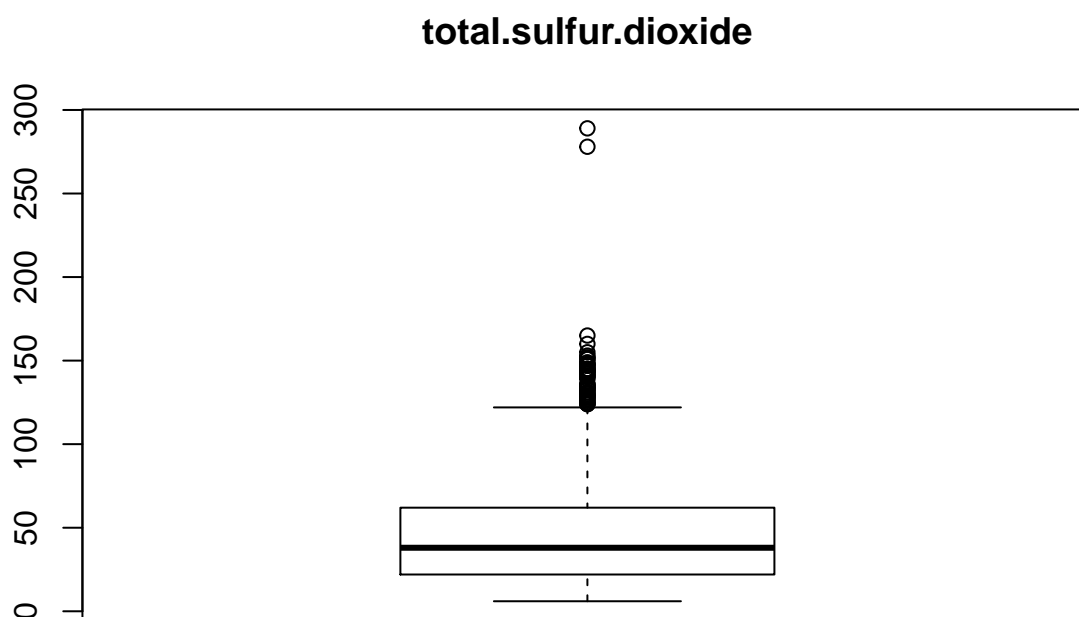
```
## [1] 145 148 136 125 140 136 133 153 134 141 129 128 129 128 143 144 127
```

```
## [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
```

```
boxplotTsulfur<-boxplot(redwine$total.sulfur.dioxide ,main="total.sulfur.dioxide", COL="gray")
```

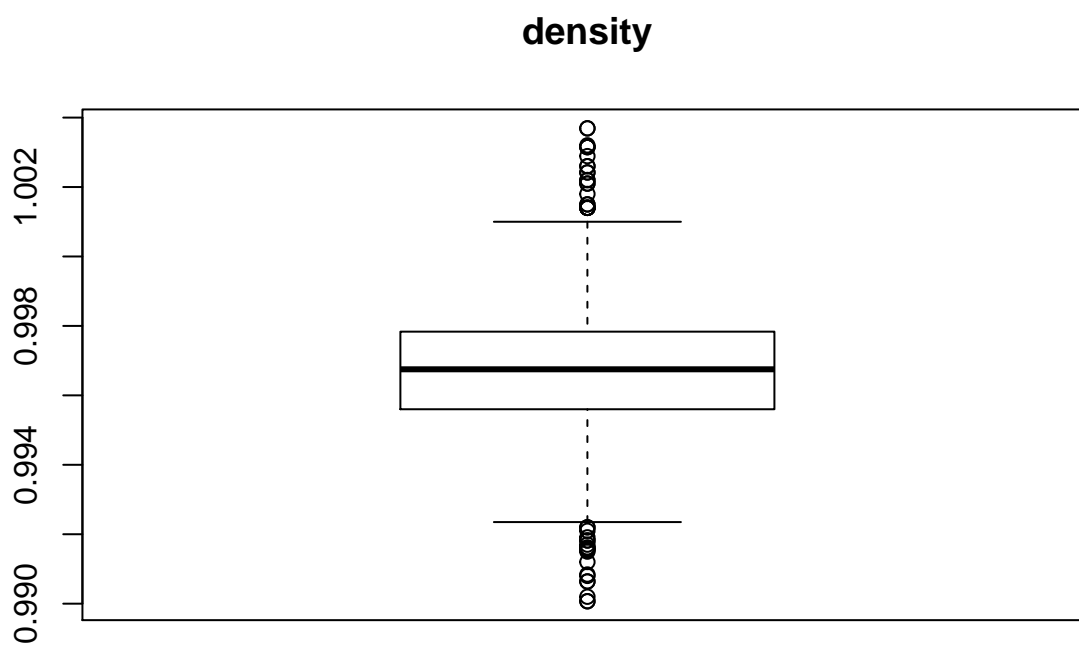


```
#Boxplot density
```

```
boxplot.stats(redwine$density)$out
```

```
## [1] 0.99160 0.99160 1.00140 1.00150 1.00150 1.00180 0.99120 1.00220
## [9] 1.00220 1.00140 1.00140 1.00140 1.00140 1.00320 1.00260 1.00140
## [17] 1.00315 1.00315 1.00315 1.00210 1.00210 0.99170 0.99220 1.00260
## [25] 0.99210 0.99154 0.99064 0.99064 1.00289 0.99162 0.99007 0.99007
## [33] 0.99020 0.99220 0.99150 0.99157 0.99080 0.99084 0.99191 1.00369
## [41] 1.00369 1.00242 0.99182 1.00242 0.99182
```

```
boxplotdensity<-boxplot(redwine$density ,main="density", COL="gray")
```



```
#Boxplot pH
```

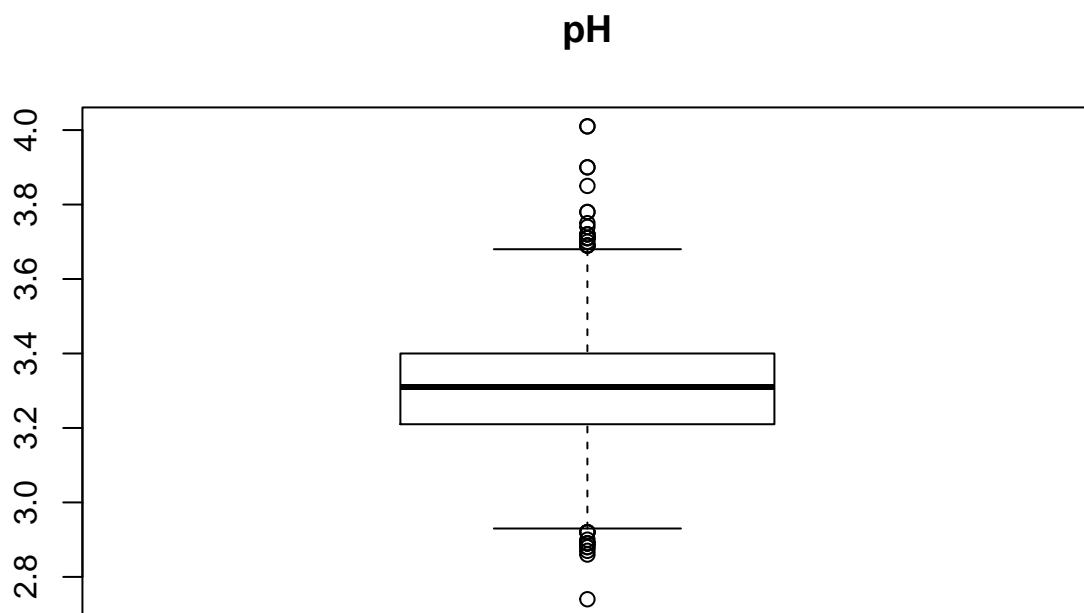
```
boxplot.stats(redwine$pH)$out
```

```
## [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
```

```
boxplotpH<-boxplot(redwine$pH ,main="pH", COL="gray")
```

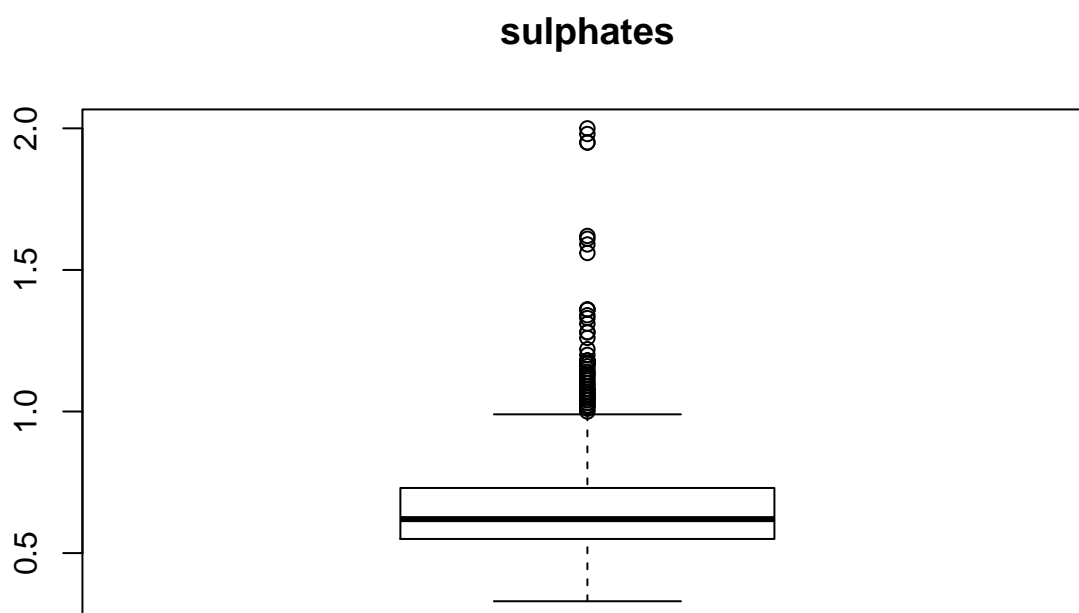


```
#Boxplot sulphates
```

```
boxplot.stats(redwine$sulphates)$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
```

```
boxplotsulphates<-boxplot(redwine$sulphates ,main="sulphates", COL="gray")
```



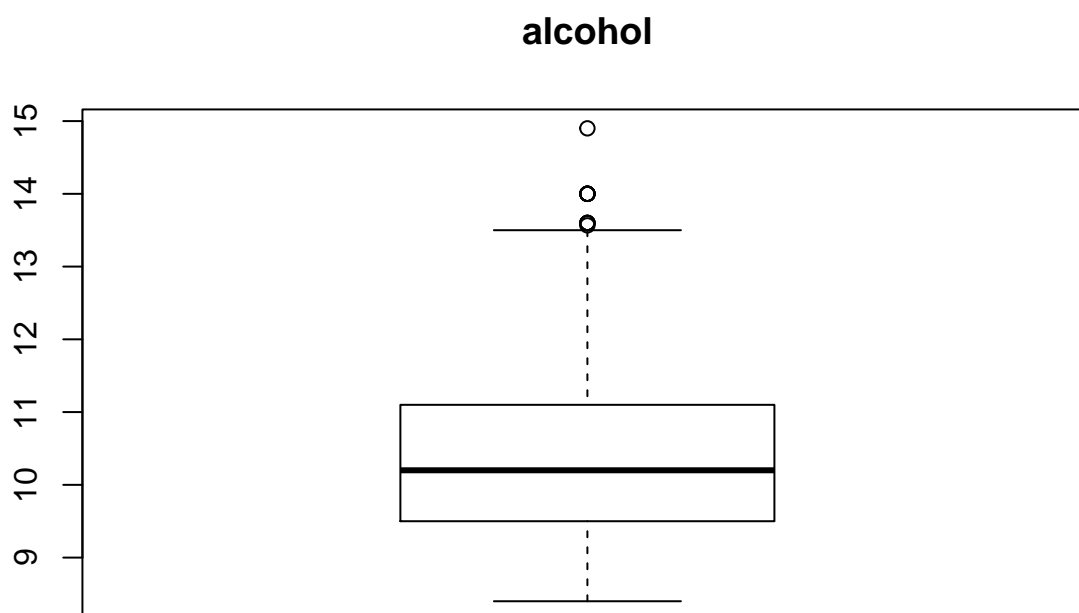
```
#Boxplot alcohol
```

```
boxplot.stats(redwine$alcohol)$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
```

```
boxplotalcohol<-boxplot(redwine$alcohol ,main="alcohol", COL="gray")
```

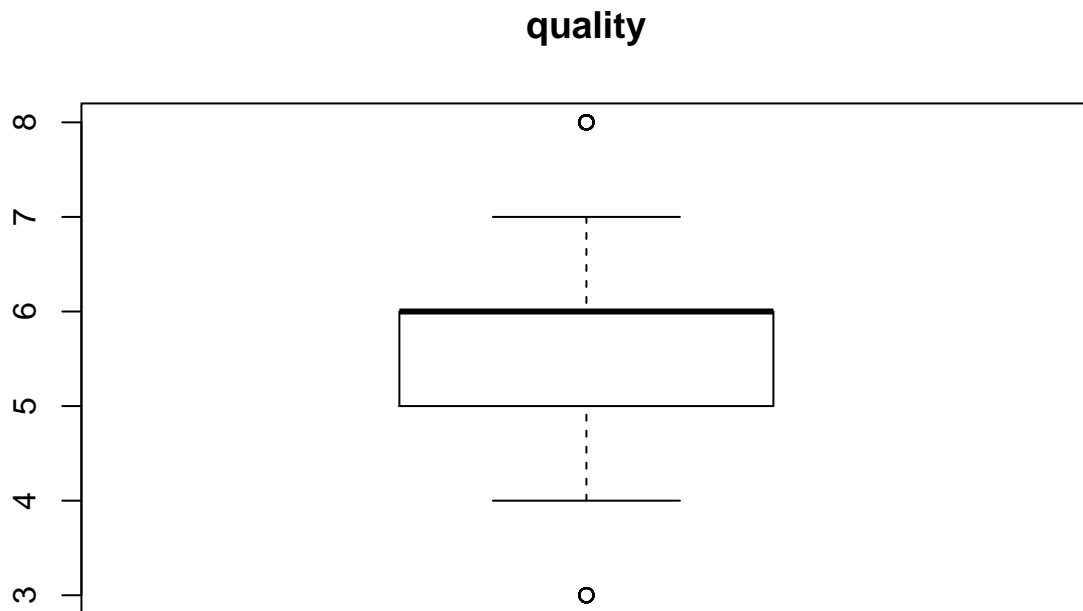


```
#Boxplot quality
```

```
boxplot.stats(redwine$quality)$out
```

```
## [1] 8 8 8 8 8 3 8 8 8 3 8 3 8 3 3 8 8 8 8 8 3 3 8 8 3 3 3 8
```

```
boxplotquality<-boxplot(redwine$quality ,main="quality", COL="gray")
```



Todos los extremos que se observan en los diagramas anteriores, son valores razonables de las distintas variables, por lo que no haremos ninguna modificación.

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

En este caso vamos a analizar todas las variables para saber su influencia en la calidad del vino, por lo que podrá ser de utilidad agrupar los vinos con calidad baja (menor que 5), media (5 o 6) y alta (mayor que 6), para poder analizar si existe alguna relación entre las variables dentro de cada grupo.

```
# Agrupación para vinos de calida alta
redwine7 <- redwine[redwine$quality >= 7,]
# Agrupación para vinos de calida baja
redwine4 <- redwine[redwine$quality <= 4,]
# Agrupación para vinos de calida media
redwine5a7 <- redwine[redwine$quality > 4 & redwine$quality < 7 ,]
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

4.2.1 Comprobacion de la normalidad

Para comprobar que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la *prueba de normalidad de Anderson-Darling*. Si en la prueba se obtiene un p-valor superior al nivel de significación prefijado de 0,05, se considera que la variable en cuestión sigue una distribución normal.

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

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

Según este test ninguna de las variables sigue una distribución normal. Podríamos usar también los *tests de Kolmogorov-Smirnov y Shapiro-Wilk*.

```
#Test Kolmogorov-Smirnov fixed.acidity
ks.test(redwine$fixed.acidity, pnorm, mean(redwine$fixed.acidity), sd(redwine$fixed.acidity))
```

```
## Warning in ks.test(redwine$fixed.acidity, pnorm,
## mean(redwine$fixed.acidity), : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$fixed.acidity
## D = 0.1105, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

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

```
##
## Shapiro-Wilk normality test
##
## data: redwine$fixed.acidity
## W = 0.94203, p-value < 2.2e-16
```

```
#Test Kolmogorov-Smirnov volatile.acidity
ks.test(redwine$volatile.acidity, pnorm, mean(redwine$volatile.acidity), sd(redwine$volatile.acidity))
```



```

## Warning in ks.test(redwine$volatile.acidity, pnorm,
## mean(redwine$volatile.acidity), : ties should not be present for the
## Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$volatile.acidity
## D = 0.054662, p-value = 0.0001416
## alternative hypothesis: two-sided
shapiro.test(redwine$volatile.acidity)

##
## Shapiro-Wilk normality test
##
## data: redwine$volatile.acidity
## W = 0.97434, p-value = 2.693e-16
#Test Kolmogorov-Smirnov citric.acid
ks.test(redwine$citric.acid, pnorm, mean(redwine$citric.acid), sd(redwine$citric.acid))

## Warning in ks.test(redwine$citric.acid, pnorm, mean(redwine$citric.acid), :
## ties should not be present for the Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$citric.acid
## D = 0.092916, p-value = 2.044e-12
## alternative hypothesis: two-sided
shapiro.test(redwine$citric.acid)

##
## Shapiro-Wilk normality test
##
## data: redwine$citric.acid
## W = 0.97095, p-value < 2.2e-16
#Test Kolmogorov-Smirnov residual.sugar
ks.test(redwine$residual.sugar, pnorm, mean(redwine$residual.sugar), sd(redwine$residual.sugar))

## Warning in ks.test(redwine$residual.sugar, pnorm,
## mean(redwine$residual.sugar), : ties should not be present for the
## Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$residual.sugar
## D = 0.26068, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$residual.sugar)

##
## Shapiro-Wilk normality test
##

```

```

## data: redwine$residual.sugar
## W = 0.56608, p-value < 2.2e-16
#Test Kolmogorov-Smirnov chlorides
ks.test(redwine$chlorides, pnorm, mean(redwine$chlorides), sd(redwine$chlorides))

## Warning in ks.test(redwine$chlorides, pnorm, mean(redwine$chlorides),
## sd(redwine$chlorides)): ties should not be present for the Kolmogorov-
## Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$chlorides
## D = 0.25964, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$chlorides)

##
## Shapiro-Wilk normality test
##
## data: redwine$chlorides
## W = 0.48425, p-value < 2.2e-16
#Test Kolmogorov-Smirnov free.sulfur.dioxide
ks.test(redwine$free.sulfur.dioxide, pnorm, mean(redwine$free.sulfur.dioxide), sd(redwine$free.sulfur.dioxide))

## Warning in ks.test(redwine$free.sulfur.dioxide, pnorm,
## mean(redwine$free.sulfur.dioxide), : ties should not be present for the
## Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$free.sulfur.dioxide
## D = 0.11124, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$free.sulfur.dioxide)

##
## Shapiro-Wilk normality test
##
## data: redwine$free.sulfur.dioxide
## W = 0.90184, p-value < 2.2e-16
#Test Kolmogorov-Smirnov total.sulfur.dioxide
ks.test(redwine$total.sulfur.dioxide, pnorm, mean(redwine$total.sulfur.dioxide), sd(redwine$total.sulfur.dioxide))

## Warning in ks.test(redwine$total.sulfur.dioxide, pnorm,
## mean(redwine$total.sulfur.dioxide), : ties should not be present for the
## Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$total.sulfur.dioxide
## D = 0.12098, p-value < 2.2e-16

```

```

## alternative hypothesis: two-sided
shapiro.test(redwine$total.sulfur.dioxide)

##
##  Shapiro-Wilk normality test
##
## data:  redwine$total.sulfur.dioxide
## W = 0.87322, p-value < 2.2e-16
#Test Kolmogorov-Smirnov density
ks.test(redwine$density, pnorm, mean(redwine$density), sd(redwine$density))

## Warning in ks.test(redwine$density, pnorm, mean(redwine$density),
## sd(redwine$density)): ties should not be present for the Kolmogorov-Smirnov
## test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  redwine$density
## D = 0.044787, p-value = 0.003274
## alternative hypothesis: two-sided
shapiro.test(redwine$density)

##
##  Shapiro-Wilk normality test
##
## data:  redwine$density
## W = 0.99087, p-value = 1.936e-08
#Test Kolmogorov-Smirnov pH
ks.test(redwine$pH, pnorm, mean(redwine$pH), sd(redwine$pH))

## Warning in ks.test(redwine$pH, pnorm, mean(redwine$pH), sd(redwine$pH)):
## ties should not be present for the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  redwine$pH
## D = 0.040368, p-value = 0.01091
## alternative hypothesis: two-sided
shapiro.test(redwine$pH)

##
##  Shapiro-Wilk normality test
##
## data:  redwine$pH
## W = 0.99349, p-value = 1.712e-06
#Test Kolmogorov-Smirnov sulphates
ks.test(redwine$sulphates, pnorm, mean(redwine$sulphates), sd(redwine$sulphates))

## Warning in ks.test(redwine$sulphates, pnorm, mean(redwine$sulphates),
## sd(redwine$sulphates)): ties should not be present for the Kolmogorov-
## Smirnov test

```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$sulphates
## D = 0.12479, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$sulphates)

##
## Shapiro-Wilk normality test
##
## data: redwine$sulphates
## W = 0.83304, p-value < 2.2e-16
#Test Kolmogorov-Smirnov alcohol
ks.test(redwine$alcohol, pnorm, mean(redwine$alcohol), sd(redwine$alcohol))

## Warning in ks.test(redwine$alcohol, pnorm, mean(redwine$alcohol),
## sd(redwine$alcohol)): ties should not be present for the Kolmogorov-Smirnov
## test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$alcohol
## D = 0.12145, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$alcohol)

##
## Shapiro-Wilk normality test
##
## data: redwine$alcohol
## W = 0.92884, p-value < 2.2e-16
#Test Kolmogorov-Smirnov quality
ks.test(redwine$quality, pnorm, mean(redwine$quality), sd(redwine$quality))

## Warning in ks.test(redwine$quality, pnorm, mean(redwine$quality),
## sd(redwine$quality)): ties should not be present for the Kolmogorov-Smirnov
## test

##
## One-sample Kolmogorov-Smirnov test
##
## data: redwine$quality
## D = 0.24982, p-value < 2.2e-16
## alternative hypothesis: two-sided
shapiro.test(redwine$quality)

##
## Shapiro-Wilk normality test
##
## data: redwine$quality
## W = 0.85759, p-value < 2.2e-16
```

Vemos que todos los test, al tener un valor p-valor mas pequeño que el 0.05, niegan la hipótesis nula que asume la normalidad.

De todas maneras, al tratarse de una muestra grande, por el teorema central del límite, podemos asumir que su distribución aproxima bien una distribución normal.

4.2.2 Homogeneidad de la varianza

Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación del *test de Fligner-Killeen*. En este caso, estudiaremos la homogeneidad de cada variable según su calidad. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
#Test Fligner-Killeen quality respecto a fixed.acidity
fligner.test(fixed.acidity ~ quality, data = redwine)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: fixed.acidity by quality
## Fligner-Killeen:med chi-squared = 34.635, df = 5, p-value =
## 1.779e-06
```

```
#Test Fligner-Killeen quality respecto a volatile.acidity
fligner.test(volatile.acidity ~ quality, data = redwine)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: volatile.acidity by quality
## Fligner-Killeen:med chi-squared = 30.826, df = 5, p-value =
## 1.014e-05
```

```
#Test Fligner-Killeen quality respecto a citric.acid
fligner.test(citric.acid ~ quality, data = redwine)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: citric.acid by quality
## Fligner-Killeen:med chi-squared = 10.974, df = 5, p-value =
## 0.05189
```

```
#Test Fligner-Killeen quality respecto a residual.sugar
fligner.test(residual.sugar ~ quality, data = redwine)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: residual.sugar by quality
## Fligner-Killeen:med chi-squared = 7.9984, df = 5, p-value = 0.1563
```

```
#Test Fligner-Killeen quality respecto a chlorides
fligner.test(chlorides ~ quality, data = redwine)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: chlorides by quality
```

```

## Fligner-Killeen:med chi-squared = 15.082, df = 5, p-value =
## 0.01002
#Test Fligner-Killeen quality respecto a free.sulfur.dioxide
fligner.test(free.sulfur.dioxide ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: free.sulfur.dioxide by quality
## Fligner-Killeen:med chi-squared = 14.137, df = 5, p-value =
## 0.01476
#Test Fligner-Killeen quality respecto a total.sulfur.dioxide
fligner.test(total.sulfur.dioxide ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: total.sulfur.dioxide by quality
## Fligner-Killeen:med chi-squared = 137.27, df = 5, p-value <
## 2.2e-16
#Test Fligner-Killeen quality respecto a density
fligner.test(density ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: density by quality
## Fligner-Killeen:med chi-squared = 49.395, df = 5, p-value =
## 1.842e-09
#Test Fligner-Killeen quality respecto a pH
fligner.test(pH ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: pH by quality
## Fligner-Killeen:med chi-squared = 1.2419, df = 5, p-value = 0.9408
#Test Fligner-Killeen quality respecto a sulphates
fligner.test(sulphates ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: sulphates by quality
## Fligner-Killeen:med chi-squared = 9.4044, df = 5, p-value =
## 0.09398
#Test Fligner-Killeen quality respecto a alcohol
fligner.test(alcohol ~ quality, data = redwine)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: alcohol by quality

```

```
## Fligner-Killeen:med chi-squared = 135.61, df = 5, p-value <
## 2.2e-16
```

En este caso podemos decir que las variables *volatile.acidity*, *fixed.acidity*, *free.sulfur.dioxide*, *total.sulfur.dioxide*, *sulphates*, *density* y *alcohol* tienen varianzas estadísticamente diferentes a *quality*, al tener un p-valor menor que el nivel de significación del 0.05.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

4.3.1 Coeficiente de correlación de Spearman

En primer lugar, procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre la calidad del vino.

Para ello, se utilizará el *coeficiente de correlación de Spearman*, que no conlleva una suposición en la distribución de los datos, puesto que hemos visto que tenemos datos que no siguen una distribución normal.

Vamos a comprobar cada una de las variables:

```
#Test correlación Spearman quality respecto a fixed.acidity
cor.test(redwine$quality,redwine$fixed.acidity, method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$fixed.acidity, method
## = "spearman"): Cannot compute exact p-value with ties
```

```
##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$fixed.acidity
## S = 603652045, p-value = 4.801e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.1140837
```

```
#Test correlación Spearman quality respecto a volatile.acidity
cor.test(redwine$quality,redwine$volatile.acidity , method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$volatile.acidity,
## method = "spearman"): Cannot compute exact p-value with ties
```

```
##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$volatile.acidity
## S = 940754860, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## -0.3806465
```

```
#Test correlación Spearman quality respecto a citric.acid
cor.test(redwine$quality,redwine$citric.acid , method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$citric.acid, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
```

```

## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$citric.acid
## S = 535596641, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.2139614

#Test correlaci3n Spearman quality respecto a residual.sugar
cor.test(redwine$quality,redwine$residual.sugar, method="spearman")

## Warning in cor.test.default(redwine$quality, redwine$residual.sugar, method
## = "spearman"): Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$residual.sugar
## S = 659549989, p-value = 0.2002
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.03204817

#Test correlaci3n Spearman quality respecto a chlorides
cor.test(redwine$quality,redwine$chlorides, method="spearman")

## Warning in cor.test.default(redwine$quality, redwine$chlorides, method =
## "spearman"): Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$chlorides
## S = 810797848, p-value = 1.883e-14
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.1899223

#Test correlaci3n Spearman quality respecto a free.sulfur.dioxide
cor.test(redwine$quality,redwine$free.sulfur.dioxide, method="spearman")

## Warning in cor.test.default(redwine$quality, redwine$free.sulfur.dioxide, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$free.sulfur.dioxide
## S = 720158572, p-value = 0.02288
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.05690065

```



```
#Test correlación Spearman quality respecto a total.sulfur.dioxide  
cor.test(redwine$quality,redwine$total.sulfur.dioxide, method="spearman")
```

```
## Warning in cor.test.default(redwine$quality,  
## redwine$total.sulfur.dioxide, : Cannot compute exact p-value with ties  
  
##  
## Spearman's rank correlation rho  
##  
## data: redwine$quality and redwine$total.sulfur.dioxide  
## S = 815439962, p-value = 2.046e-15  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.1967351
```

```
#Test correlación Spearman quality respecto a density  
cor.test(redwine$quality,redwine$density, method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$density, method =  
## "spearman"): Cannot compute exact p-value with ties  
  
##  
## Spearman's rank correlation rho  
##  
## data: redwine$quality and redwine$density  
## S = 802043202, p-value = 9.918e-13  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.1770741
```

```
#Test correlación Spearman quality respecto a pH  
cor.test(redwine$quality,redwine$pH, method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$pH, method =  
## "spearman"): Cannot compute exact p-value with ties  
  
##  
## Spearman's rank correlation rho  
##  
## data: redwine$quality and redwine$pH  
## S = 711144697, p-value = 0.08085  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.04367193
```

```
#Test correlación Spearman quality respecto a sulphates  
cor.test(redwine$quality,redwine$sulphates, method="spearman")
```

```
## Warning in cor.test.default(redwine$quality, redwine$sulphates, method =  
## "spearman"): Cannot compute exact p-value with ties  
  
##  
## Spearman's rank correlation rho  
##  
## data: redwine$quality and redwine$sulphates
```

```
## S = 424463207, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.3770602

#Test correlaci3n Spearman quality respecto a alcohol
cor.test(redwine$quality,redwine$alcohol, method="spearman")

## Warning in cor.test.default(redwine$quality, redwine$alcohol, method =
## "spearman"): Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: redwine$quality and redwine$alcohol
## S = 355321833, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.4785317
```

Segun el coeficiente de correlaci3n de Spearman, los atributos que tienen mayor influencia en la calidad del vino tinto son el alcohol, los sulfatos y la acidez volátil.

4.3.2 Contraste de hipótesis

Procederemos ahora a hacer un contraste de hipótesis sobre las cuatro variables con el p-valor más bajo que hemos encontrado en el apartado anterior. Para ello buscaremos la mediana de las tres variables, para que nos sirva de valor límite. Compararemos si, es cuando el valor de la variable es menor al valor medio cuando influye en la calidad, o por el contrario, si influye cuando es superior al valor medio.

```
#medianas de volatile.acidity, total.sulfur.dioxide, alcohol y density
volatile.acidity.median = median(redwine$volatile.acidity)
volatile.acidity.median
```

```
## [1] 0.52
```

```
alcohol.median = median(redwine$alcohol)
alcohol.median
```

```
## [1] 10.2
```

```
total.sulfur.dioxide.median = median(redwine$total.sulfur.dioxide)
total.sulfur.dioxide.median
```

```
## [1] 38
```

```
density.median = median(redwine$density)
density.median
```

```
## [1] 0.99675
```

Separaremos las muestras según estos valores:

```
#nuevos dataframes con separacion de volatile.acidity
redwineVAinf <-redwine[redwine$volatile.acidity < volatile.acidity.median,]$quality
redwineVASup <-redwine[redwine$volatile.acidity > volatile.acidity.median,]$quality
#nuevos dataframes con separacion de alcohol
```

```

redwineAinf <-redwine[redwine$alcohol < alcohol.median,]$quality
redwineAsup <-redwine[redwine$alcohol > alcohol.median,]$quality
#nuevos dataframes con separacion de total.sulfur.dioxide
redwineTSinf <-redwine[redwine$total.sulfur.dioxide < total.sulfur.dioxide.median,]$quality
redwineTSsup <-redwine[redwine$total.sulfur.dioxide > total.sulfur.dioxide.median,]$quality

#nuevos dataframes con separacion de density
redwineDinf <-redwine[redwine$density < density.median,]$quality
redwineDsup <-redwine[redwine$density > density.median,]$quality

```

Los valores medios los hemos obviado para simplificar en la practica los calculos, ya que al ser una muestra grande la influencia de estos datos coincidentes no afectan tanto.

Al ser una distribución mayor a 30, se supone una normalidad por el Teorema del Límite Central, usaremos el test T de Student.

Planteamos ahora el primer caso que consiste en el contraste de hipótesis de dos muestras sobre la diferencia de medias (la primera muestra es de los vinos con acidez volátil de menos de 0.52 y la segunda muestra es de los vinos con acidez volátil mayores a 0.52), el cual es unilateral atendiendo a la formulación de la hipótesis alternativa:

$$\begin{cases} H_0 : \mu_1 - \mu_2 = 0 \\ H_1 : \mu_1 - \mu_2 < 0 \end{cases}$$

Donde μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda.

Tomaremos un nivel de significación del 0,05.

```

#test T de Student sobre los dataframes separados de volatile.acidity
t.test(redwineVAinf, redwineVASup, alternative = "less")

```

```

##
##  Welch Two Sample t-test
##
## data:  redwineVAinf and redwineVASup
## t = 13.118, df = 1552.9, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.5720762
## sample estimates:
## mean of x mean of y
##  5.890166  5.381865

```

Planteamos ahora el segundo caso que consiste en el contraste de hipótesis de dos muestras sobre la diferencia de medias (la primera muestra es de los vinos con la variable alcohol inferior a 10.2 y la segunda muestra es de los vinos con la variable alcohol mayores a 10.2), el cual es unilateral atendiendo a la formulación de la hipótesis alternativa:

$$\begin{cases} H_0 : \mu_1 - \mu_2 = 0 \\ H_1 : \mu_1 - \mu_2 < 0 \end{cases}$$

Donde μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda.

Tomaremos un nivel de significación del 0,05.

```
#test T de Student sobre los dataframes separados de alcohol
t.test(redwineAinf, redwineAsup, alternative = "less")
```

```
##
## Welch Two Sample t-test
##
## data: redwineAinf and redwineAsup
## t = -17.869, df = 1411, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.6105772
## sample estimates:
## mean of x mean of y
##  5.310302  5.982827
```

Planteamos ahora el tercer caso que consiste en el contraste de hipótesis de dos muestras sobre la diferencia de medias (la primera muestra es de los vinos con sulfatos inferior a 38 y la segunda muestra es de los vinos con sulfatos mayores a 38), el cual es unilateral atendiendo a la formulación de la hipótesis alternativa:

$$\begin{cases} H_0 : \mu_1 - \mu_2 = 0 \\ H_1 : \mu_1 - \mu_2 < 0 \end{cases}$$

Donde μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda.

Tomaremos un nivel de significación del 0,05.

```
#test T de Student sobre los dataframes separados de total.sulfur.dioxide
t.test(redwineTSinf, redwineTSsup, alternative = "less")
```

```
##
## Welch Two Sample t-test
##
## data: redwineTSinf and redwineTSsup
## t = 5.7836, df = 1529.6, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.3003768
## sample estimates:
## mean of x mean of y
##  5.750630  5.516796
```

El valor que tenemos inferior el p-valor al del nivel de significación es que corresponde al *alcohol*. En este aceptamos que si el vino tiene más alcohol el vino tiene mas calidad.

En el resto de ejemplos sobre las variables *density* y *total.sulfur.dioxide*, el p valor es 1, por tanto aceptamos la hipótesis nula indicándonos que estos valores no afectan a la calidad del vino al tener medias parecidas

4.3.3 Regresion Lineal

Uno de los objetivos al inspeccionar el dataset era poder predecir la calidad del vino dependiendo de las variables. De esta manera vamos a proceder a calcular mediante un modelo de regresión lineal usando las variables más influyentes detectadas en el primer punto de este primer apartado.

Entrenamos varios modelos de regresión lineal utilizando validación cruzada, y elegiremos el modelo mas eficiente comparando su coeficiente de determinación.

4.3.3.1 Modelo 1: sulphates, alcohol, volatile.acidity

```
library(caret)

## Loading required package: lattice
## Loading required package: ggplot2
## Registered S3 methods overwritten by 'ggplot2':
##   method      from
##   [.quosures   rlang
##   c.quosures   rlang
##   print.quosures rlang
##
## Attaching package: 'ggplot2'
## The following objects are masked from 'package:psych':
##
##   %+%, alpha
modelo1 <- train(
  quality ~ sulphates+alcohol+volatile.acidity,
  redwine, method = "lm", trControl = trainControl(method = "cv", number = 10))
print(modelo1)

## Linear Regression
##
## 1599 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1439, 1439, 1440, 1439, 1440, 1438, ...
## Resampling results:
##
##    RMSE      Rsquared   MAE
## 0.6572691  0.3413718  0.5163557
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

4.3.3.2 Modelo 2: sulphates, fixed.acidity, volatile.acidity, chlorides, density

```
modelo2 <- train(
  quality ~ sulphates+fixed.acidity+volatile.acidity+chlorides+density,
  redwine, method = "lm", trControl = trainControl(method = "cv", number = 10))
print(modelo2)

## Linear Regression
##
## 1599 samples
##    5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1438, 1440, 1439, 1439, 1438, 1441, ...
## Resampling results:
```

```
##
##      RMSE      Rsquared  MAE
##    0.6919319  0.269035  0.5537877
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

4.3.3.3 Modelo 3: sulphates, alcohol, citric.acid, fixed.acidity, volatile.acidity, chlorides, density

```
modelo3 <- train(
  quality ~ sulphates+alcohol+citric.acid+fixed.acidity+volatile.acidity+chlorides+density,
  redwine, method = "lm", trControl = trainControl(method = "cv", number = 10))
print(modelo3)
```

```
## Linear Regression
##
## 1599 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1440, 1439, 1439, 1440, 1438, 1439, ...
## Resampling results:
##
##      RMSE      Rsquared  MAE
##    0.6537025  0.3504917  0.5125666
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

4.3.3.4 Modelo 4: sulphates, citric.acid, fixed.acidity, volatile.acidity, chlorides, density

```
modelo4 <- train(
  quality ~ sulphates+citric.acid+fixed.acidity+volatile.acidity+chlorides+density,
  redwine, method = "lm", trControl = trainControl(method = "cv", number = 10))
print(modelo4)
```

```
## Linear Regression
##
## 1599 samples
##    6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1438, 1439, 1438, 1438, 1440, 1439, ...
## Resampling results:
##
##      RMSE      Rsquared  MAE
##    0.6910981  0.2700403  0.5539663
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

4.3.3.5 Modelo 5: sulphates, alcohol, fixed.acidity, volatile.acidity, chlorides, density

```

modelo5 <- train(
  quality ~ sulphates+alcohol+fixed.acidity+volatile.acidity+chlorides+density,
  redwine, method = "lm", trControl = trainControl(method = "cv", number = 10))
print(modelo5)

```

```

## Linear Regression
##
## 1599 samples
##    6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1440, 1439, 1440, 1438, 1438, 1439, ...
## Resampling results:
##
##    RMSE          Rsquared    MAE
## 0.6545101  0.3466818  0.5125363
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

```

4.3.3.6 Elección del modelo

Comparamos el coeficiente de determinación de los modelos obtenidos para elegir el que ajuste mejor.

```

# 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,
5, summary(modelo5)$r.squared),
ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes

```

```

##      Modelo      R^2
## [1,]      1 0.3358973
## [2,]      2 0.2727333
## [3,]      3 0.3506027
## [4,]      4 0.2727651
## [5,]      5 0.3483335

```

En este caso elegiremos el modelo 3 porque tiene el coeficiente de determinación más alto, es el que mejor explica la variabilidad de la muestra y mayor capacidad predictiva tiene.

Ahora, empleando este modelo, podemos predecir la calidad del vino a partir de sus características, como en el siguiente ejemplo:

```

newdata <- data.frame(
  alcohol=6, citric.acid=0.56, fixed.acidity=6.2, volatile.acidity=0.65,
  chlorides=0.08, density= 0.88, sulphates=0.25)

predict = predict(modelo3, newdata)

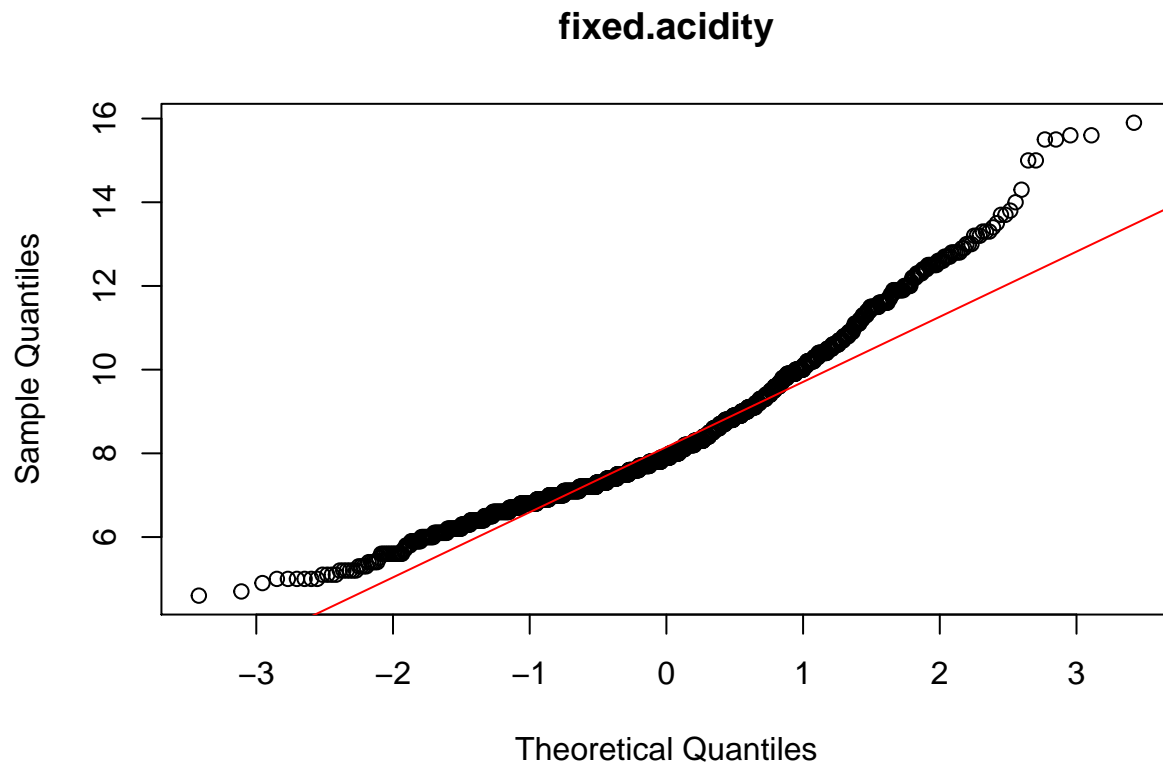
```

El modelo estima una calidad de 7 para un vino tinto con las características indicadas.

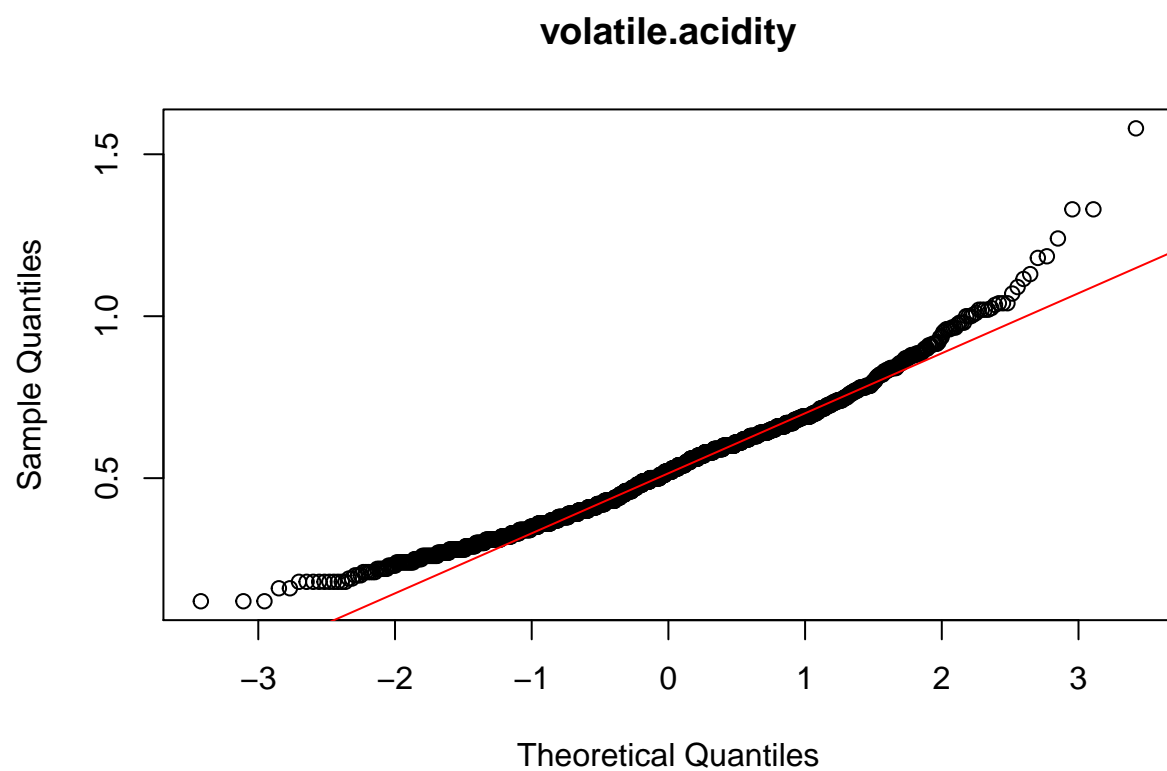
5. Representación de los resultados a partir de tablas y gráficas.

Al tratarse de una población que por el tamaño se acerca de la normalidad en su distribución, optaremos por los gráficos qq de cada variable, de manera que compararemos los cuantiles de la distribución observada con los cuantiles teóricos de una distribución normal, y cuanto más se aproxime los datos a una normal, más alineados se mostrarán sus puntos a la recta.

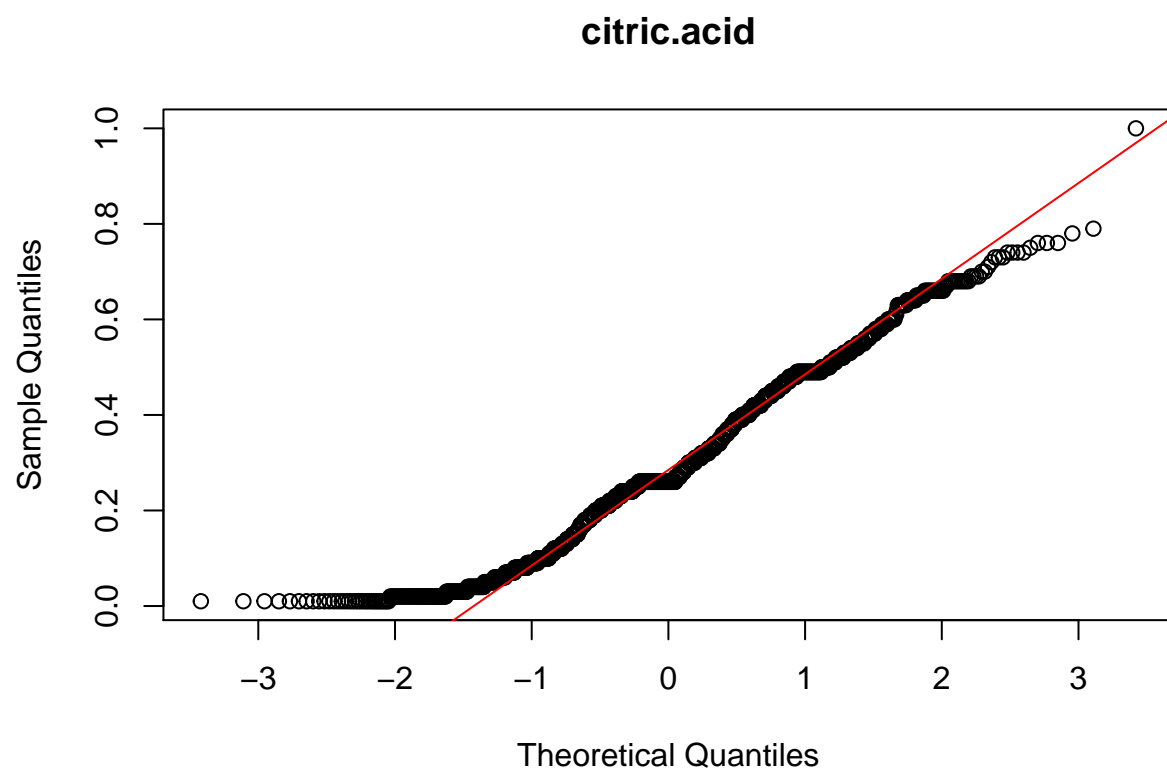
```
#gráfico qq fixed.acidity  
qqnorm(redwine$fixed.acidity, main="fixed.acidity")  
qqline(redwine$fixed.acidity, col=2)
```



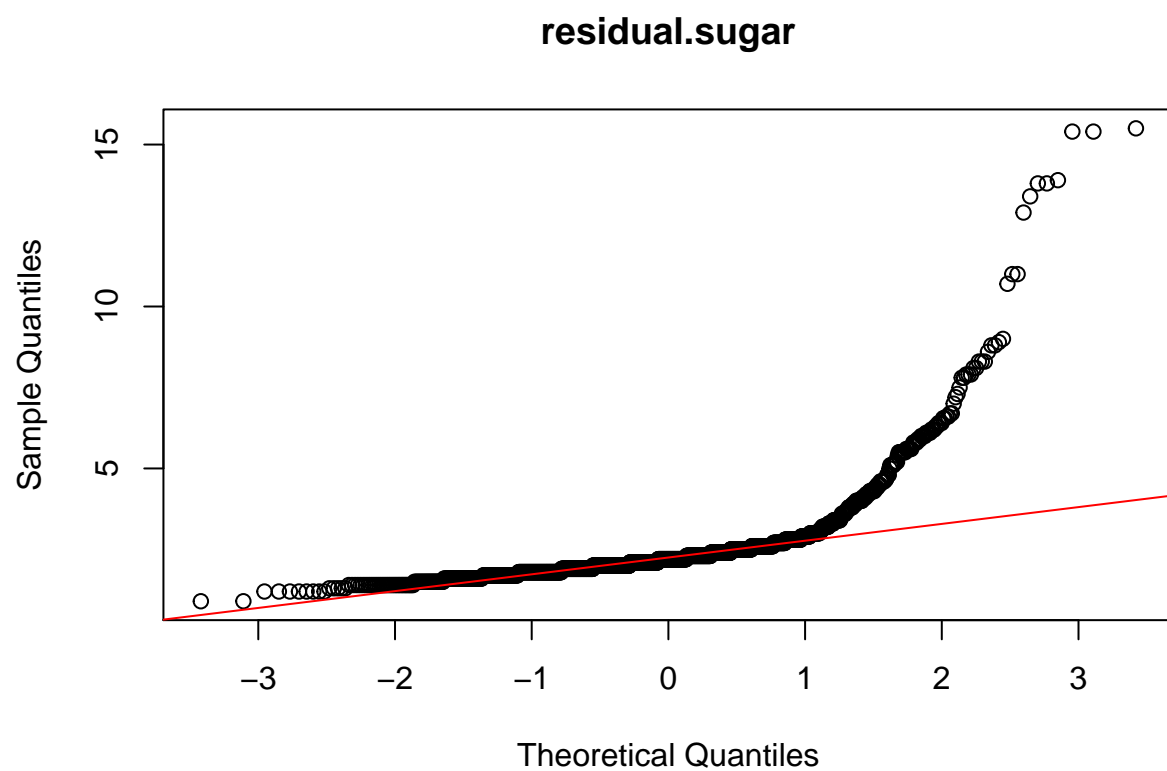
```
#gráfico qq volatile.acidity  
qqnorm(redwine$volatile.acidity, main="volatile.acidity")  
qqline(redwine$volatile.acidity, col=2)
```

```
#gráfico qq citric.acid  
qqnorm(redwine$citric.acid, main="citric.acid")  
qqline(redwine$citric.acid, col=2)
```

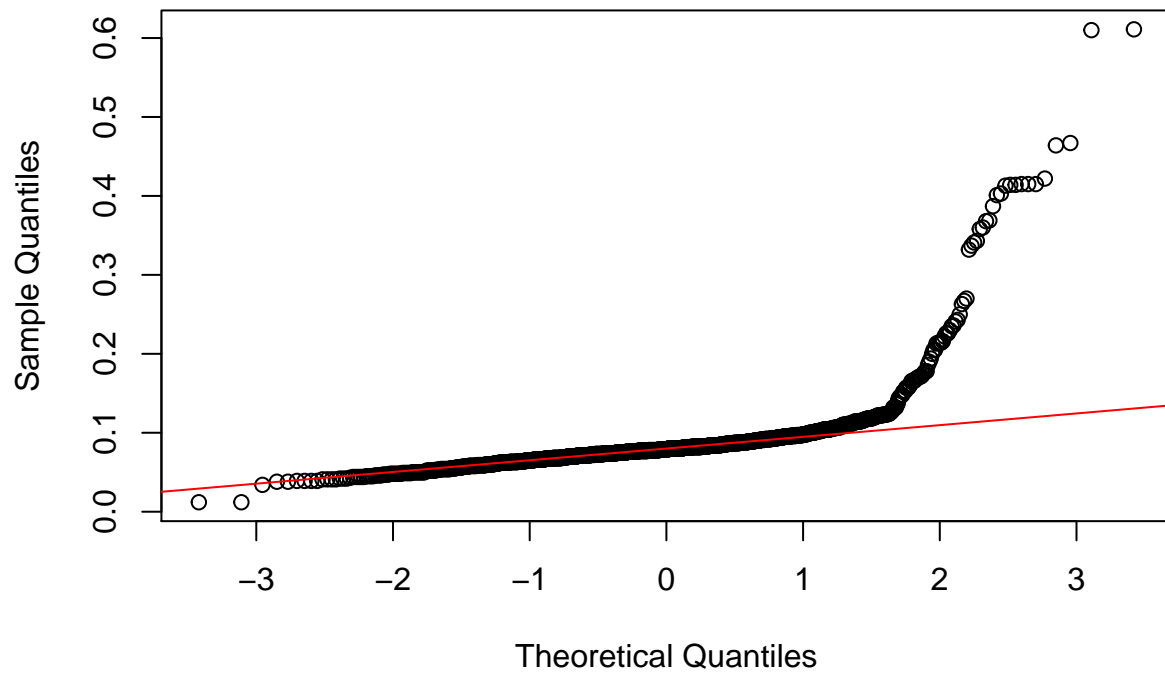


```
#gráfico qq residual.sugar  
qqnorm(redwine$residual.sugar, main="residual.sugar")  
qqline(redwine$residual.sugar, col=2)
```



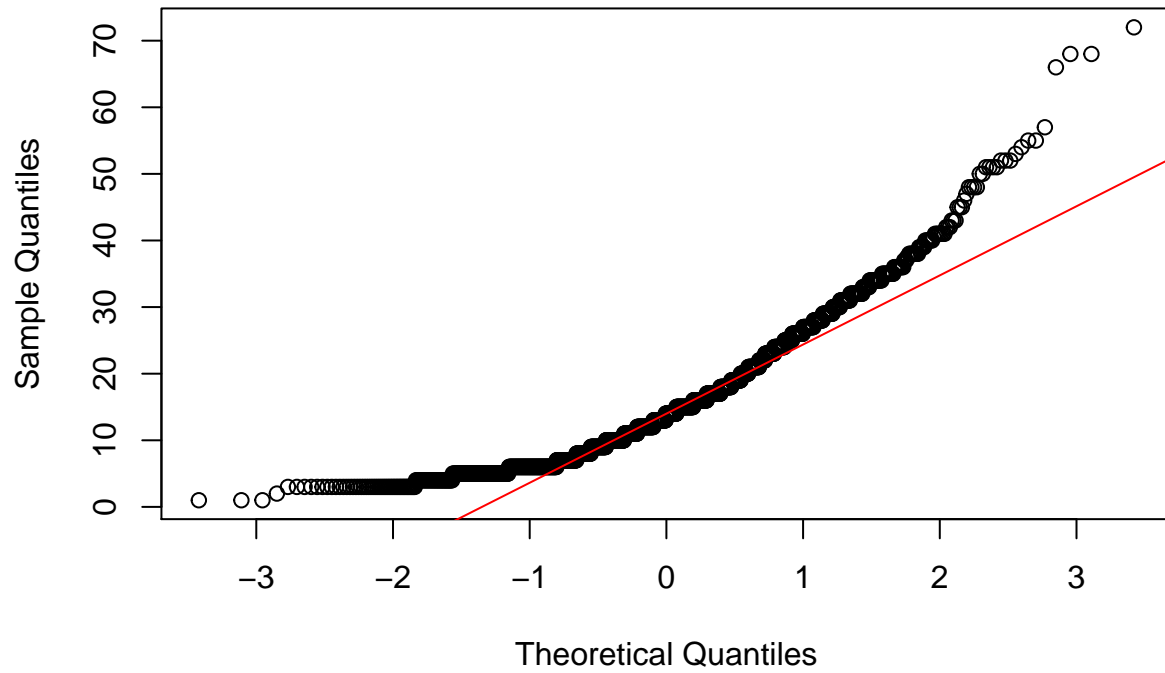
```
#gráfico qq chlorides  
qqnorm(redwine$chlorides, main="chlorides")  
qqline(redwine$chlorides, col=2)
```

chlorides

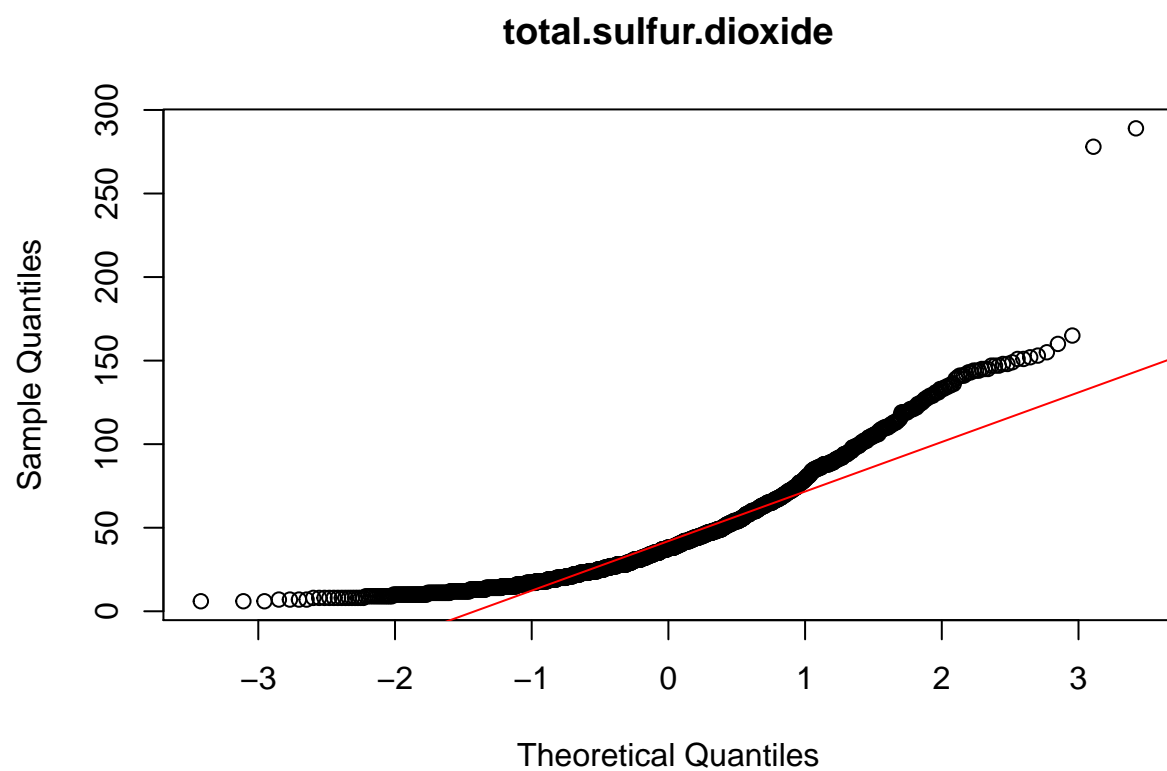


```
#gráfico qq free.sulfur.dioxide  
qqnorm(redwine$free.sulfur.dioxide, main="free.sulfur.dioxide")  
qqline(redwine$free.sulfur.dioxide, col=2)
```

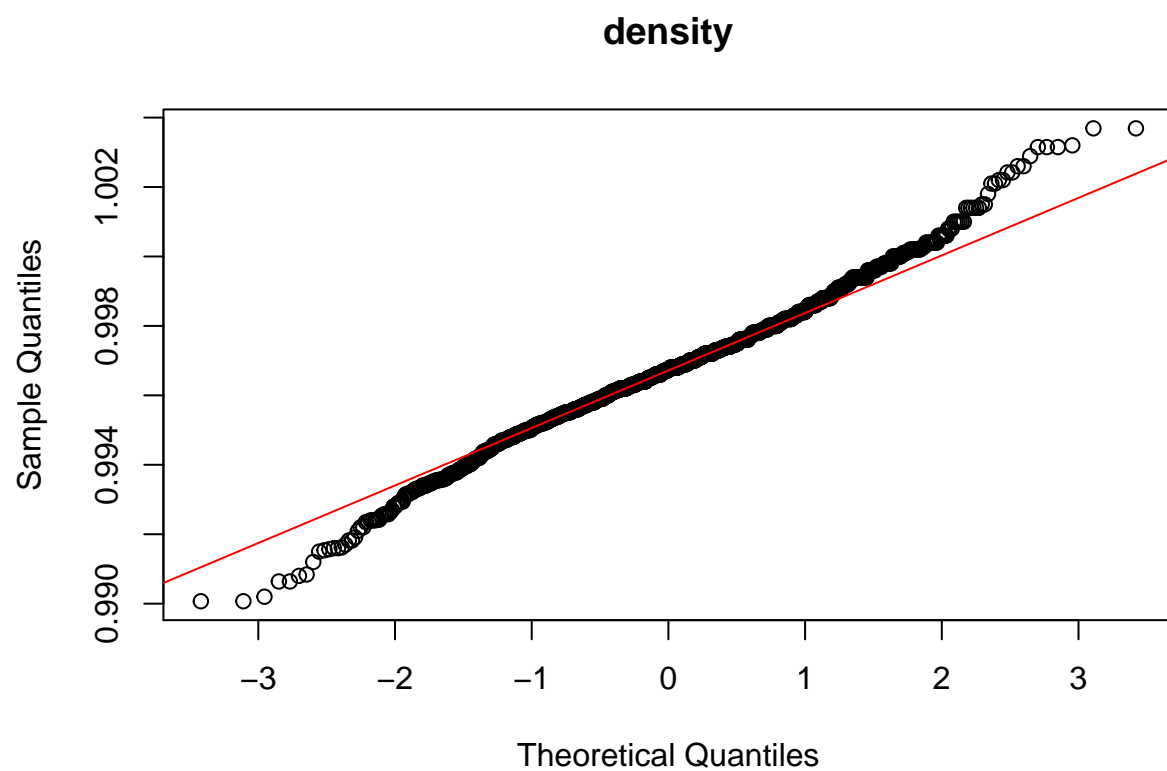
free.sulfur.dioxide



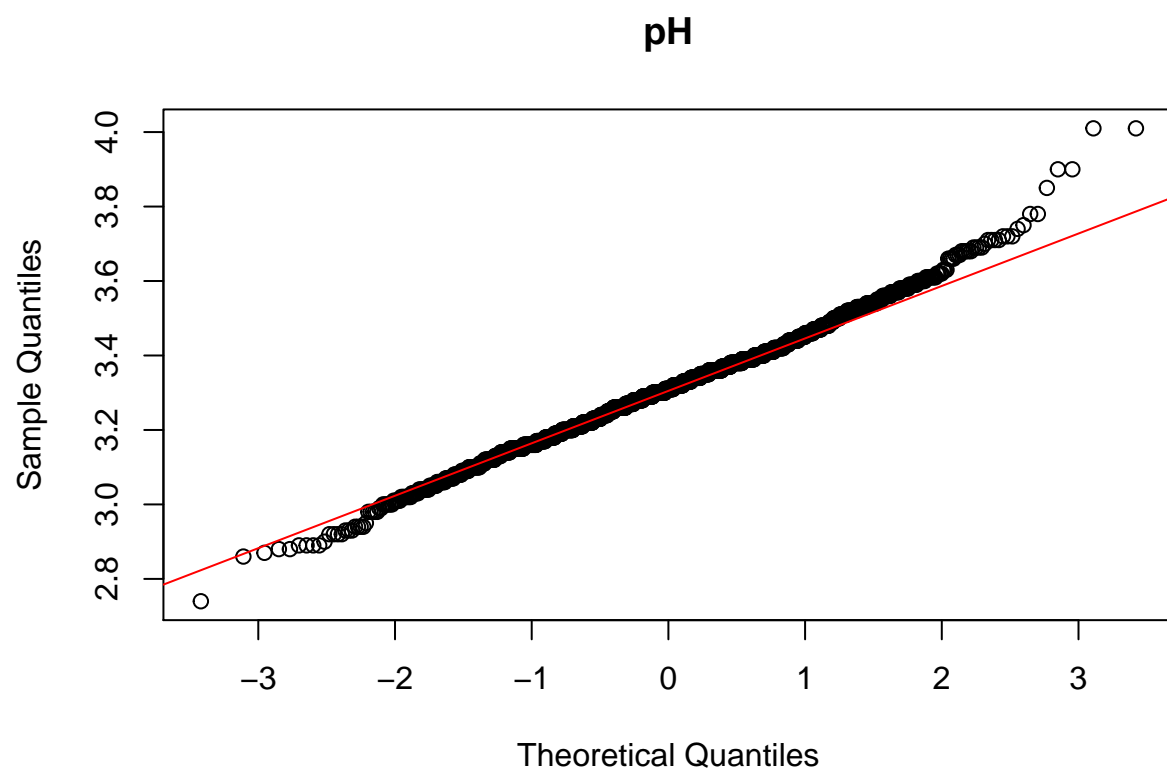
```
#gráfico qq total.sulfur.dioxide  
qqnorm(redwine$total.sulfur.dioxide, main="total.sulfur.dioxide")  
qqline(redwine$total.sulfur.dioxide, col=2)
```



```
#gráfico qq density  
qqnorm(redwine$density, main="density")  
qqline(redwine$density, col=2)
```

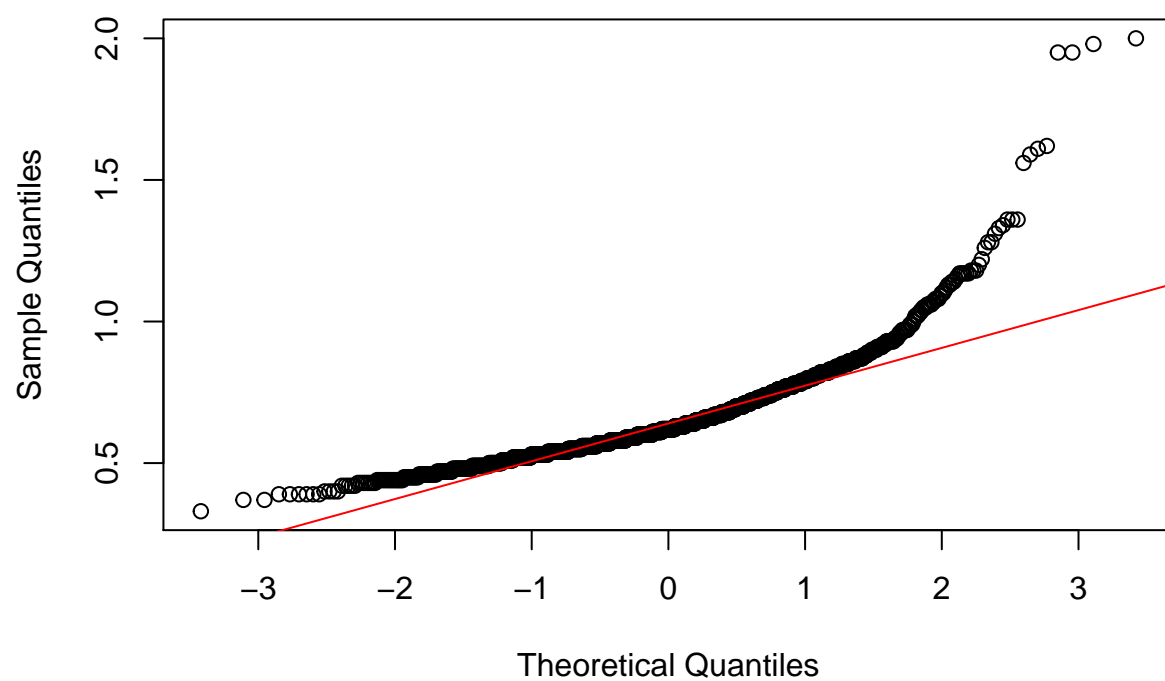


```
#gráfico qq pH  
qqnorm(redwine$pH, main="pH")  
qqline(redwine$pH, col=2)
```



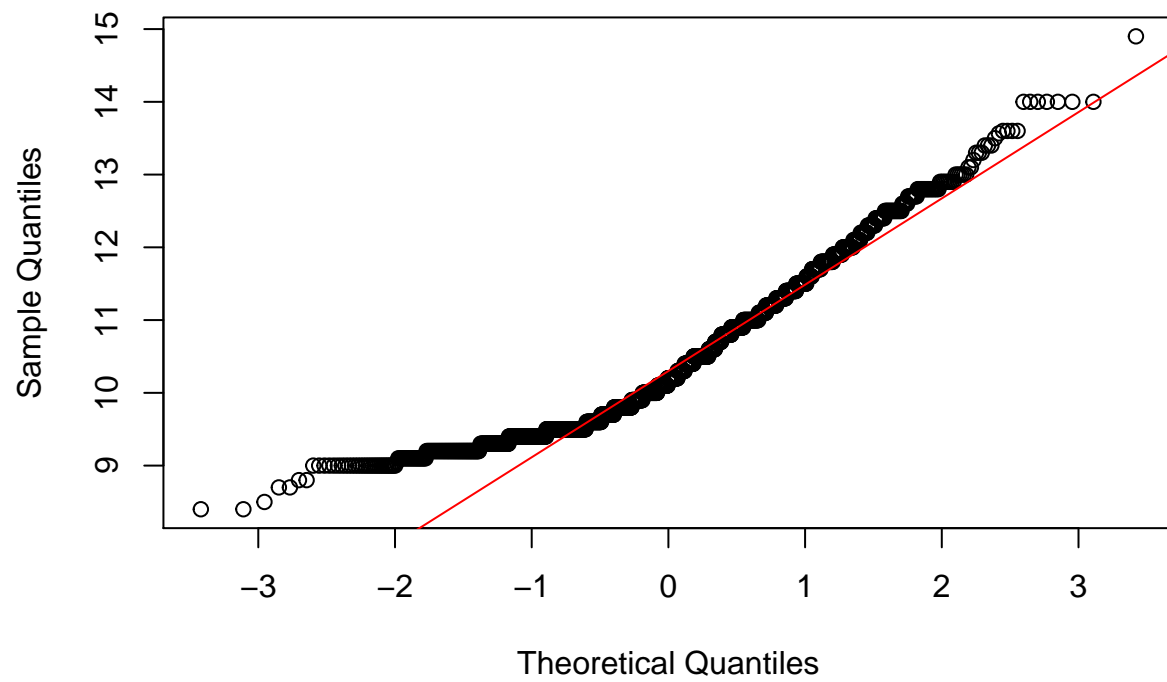
```
#gráfico qq sulphates  
qqnorm(redwine$sulphates, main="sulphates")  
qqline(redwine$sulphates, col=2)
```


sulphates

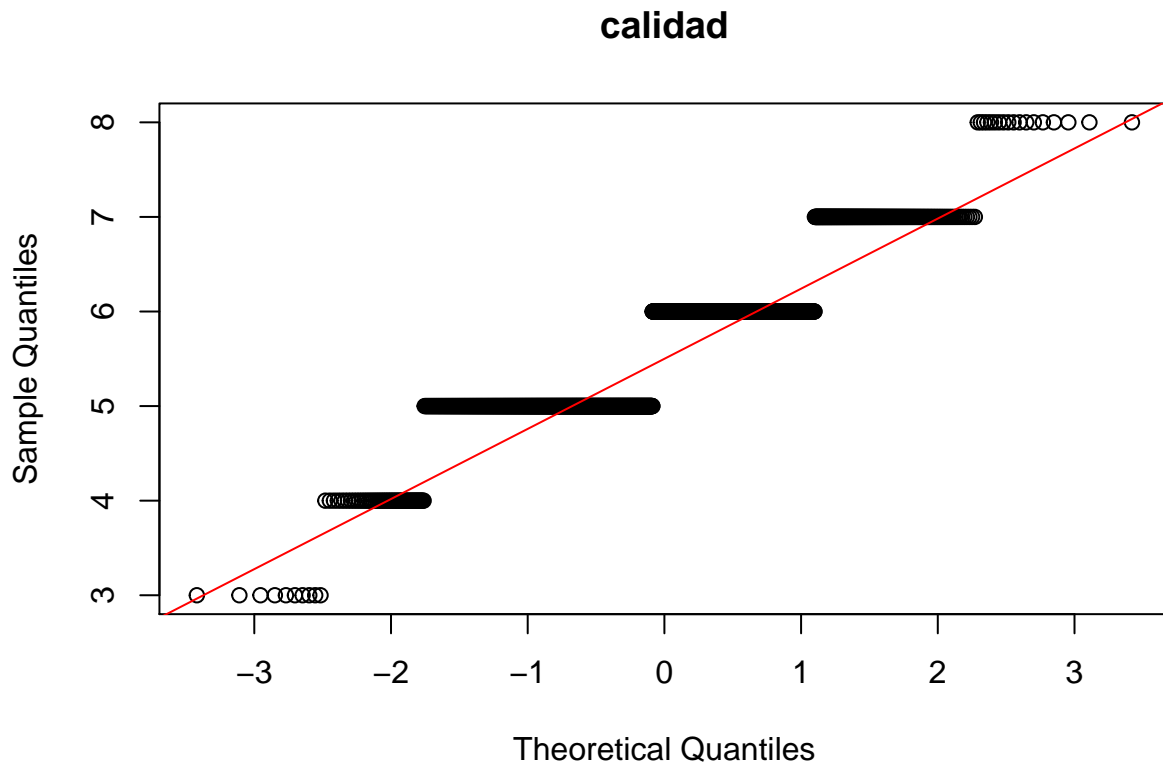


```
#gráfico qq alcohol  
qqnorm(redwine$alcohol, main="alcohol")  
qqline(redwine$alcohol, col=2)
```

alcohol



```
#gráfico qq quality  
qqnorm(redwine$quality, main="calidad")  
qqline(redwine$quality,col=2)
```



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

Hemos podido extraer algunas conclusiones generales acerca de la calidad de un vino tinto, como que su calidad será mayor cuanto menor sea su acidez volátil o mayor su nivel de alcohol, por ejemplo.

También hemos diseñado varios modelos de regresión lineal para predecir la calidad del vino tinto, utilizando factores distintos en cada uno de ellos y entrenándolos con validación cruzada. Hemos seleccionado el de mayor capacidad predictiva, escogiendo para ello el modelo con mayor coeficiente de determinación.

De esta manera, hemos podido predecir la calidad de un vino a partir de sus características fisicoquímicas, de manera que una bodega, por ejemplo, podrá fijar el precio de un vino a partir de su calidad, o un restaurante, escoger el de mayor calidad en un nivel de precio. Es decir, hemos podido responder al problema planteado, conocer la calidad de un vino tinto a partir de sus características fisicoquímicas.

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y validación de los datos.

La práctica se ha realizado en formato R Markdown para construir el documento con R de forma dinámica, con las respuestas a las preguntas y pedazos de código R embebidos. La entrega consiste en un archivo Rmd y el PDF que se genera automáticamente a partir del archivo anterior.

8. Contribuciones

- Investigación previa: MC, AD
- Redacción de las respuestas: MC, AD
- Desarrollo código: MC, AD