Práctica 2: Limpieza y análisis de datos

Carlos Tejedor González

7 de junio, 2021

Contents

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

El dataset elegido se trata del llamado Red Wine Quality, disponible en Kaggle https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 y también en su fuente original, el repositorio UCI: https://archive.ics.uci.edu/ml/datasets/wine+quality

Las variables que nos ofrece este dataset son:

1. **fixed acidity**
2. **volatile acidity**
3. **citric acid**
4. **residual sugar**
5. **chlorides**
6. **free sulfur dioxide**
7. **total sulfur dioxide**

8. **density**
9. **pH**
10. **sulphates**
11. **alcohol**
12. **quality**

Las variables de la 1 a la 11 se tratan de indicadores que se obtienen de las diferentes muestras de vino mediante análisis fisicoquímicos.

La variable 12, quality, nos indica en una escala del 1 al 10 la calidad percibida del vino.

Existe, en otro dataset, un conjunto idéntico del mismo autor pero sobre vino blanco denominado "White Wine Quality" que también vamos a utilizar: https://www.kaggle.com/piyushagni5/white-wine-quality

La problemática que aquí nos encontramos es ¿Qué características del vino influyen en su calidad y en qué medida? y también si los vinos blancos tienen mayor calidad que los tintos.

Todo esto lo vamos a tratar de responder con un análisis estadístico realizado tras un limpieza de los datos de que disponemos.

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

En primer lugar vamos a cargar el dataset y comprobamos que la lectura es correcta:

```
red_wine<-read.csv("winequality-red.csv", sep=",")
str(red_wine)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

Añadimos una nueva variable para indicar que se trata de vino tinto.

```
red_wine$type <- "Red"
```

Procedemos a la lectura del dataset sobre vino blanco.

```
white_wine<-read.csv("winequality-white.csv", sep=";")
str(white_wine)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
```

```
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
```

Las variables son idénticas a las del vino tinto, por lo que solamente añadimos la variable para indicar que se trata de vino blanco.

```
white_wine$type <- "White"
```

Hacemos una integración vertical de los de los dos dataset con los dos tipos de vinos:

```
wine<-rbind(red_wine,white_wine)
```

Por otra parte, tal y como viene representada la información, no procedería hacer selección alguna. Es cierto que en el análisis posterior se pueden descubrir variables que no sean interesantes porque no influyan en la calidad del vino (quality), que es la variable a predecir, pero en este momento tenemos que utilizar todas ellas con la información más completa posible para encontrar las posibles relaciones que existan pues hacerlo ahora resultaría prematuro.

3. Limpieza de los datos.

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

Comprobamos la existencia de valores nulos.

```
colSums(is.na(wine))
```

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

Hacemos lo mismo con valores vacíos.

```
colSums(wine=="")
```

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

Existencia de valores 0.

```
colSums(wine==0)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                  151
##       residual.sugar            chlorides  free.sulfur.dioxide
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
##                 type
##                    0
```

No existen valores vacíos o "NA", pero hemos encontrado algunos valores 0 en la variable "citric.acid" que no tienen por qué ser un error o valor perdido, puesto que es una variable númerica. De todos modos vamos a ver como se distribuye la misma por si nos diera alguna pista al respecto:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

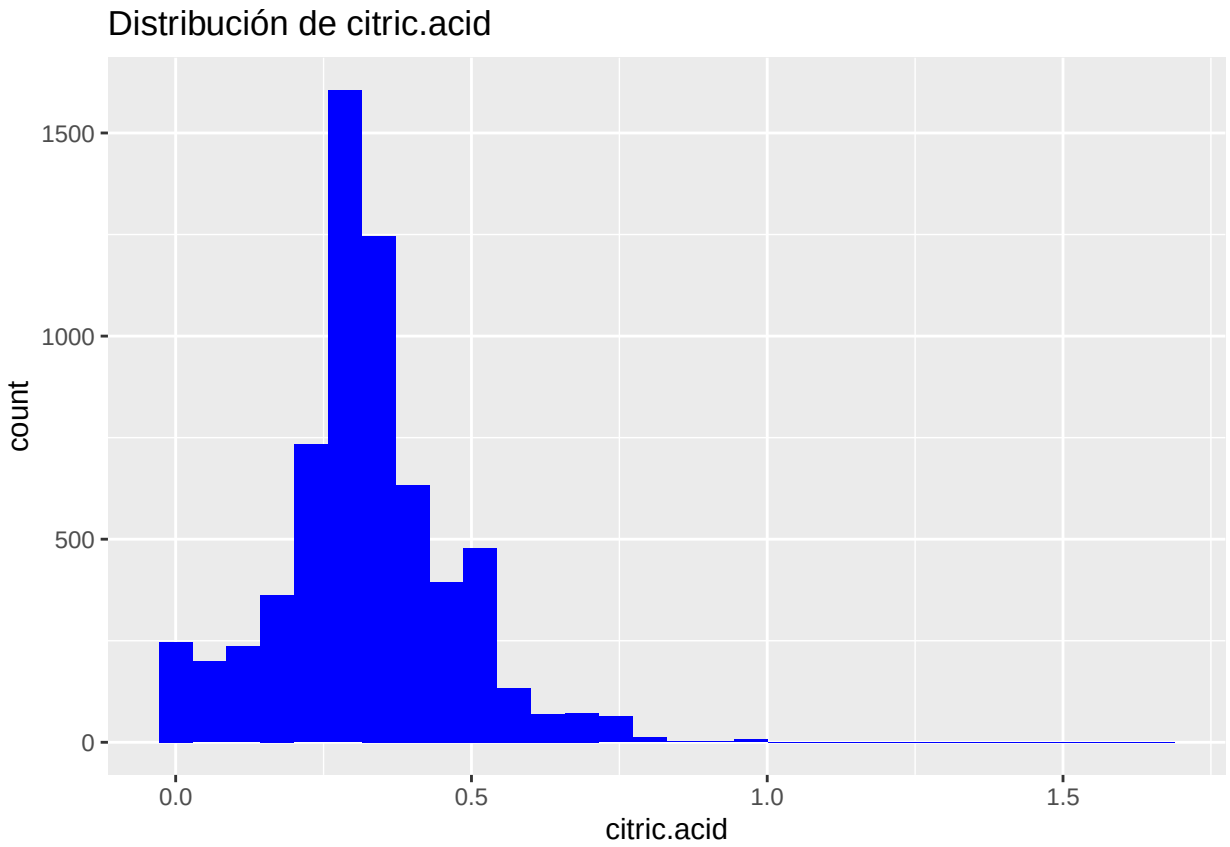
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

wine%>%ggplot(aes(citric.acid))+geom_histogram(fill="blue")+ggtitle("Distribución de citric.acid")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



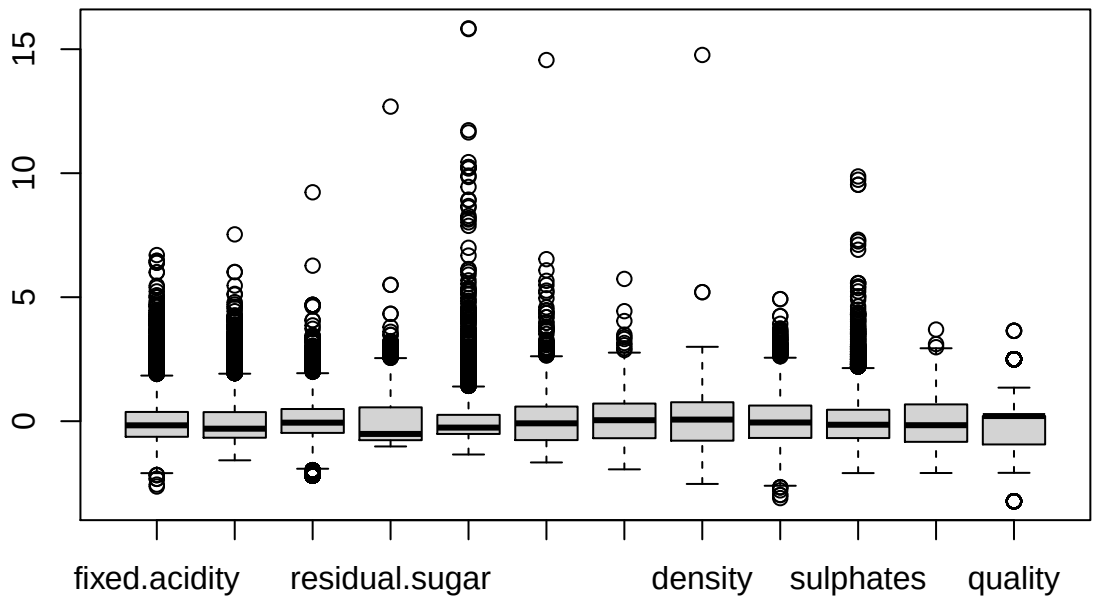
Por la distribución de la variable, se puede concluir que los valores 0 son un valor correcto y, por lo tanto, no hay que hacer limpieza alguna.

Nos hemos encontrado con un dataset realmente limpio, algo tan positivo como inusual.

3.2. Identificación y tratamiento de valores extremos.

Para la identificación de los valores extremos, vamos a representar un boxplot de cada variable numérica. Previamente con sclae() ponemos todas las variables en la misma escala, ya que evidentemente cada una de ellas es diferente.

```
boxplot(scale(wine[0:12]))
```

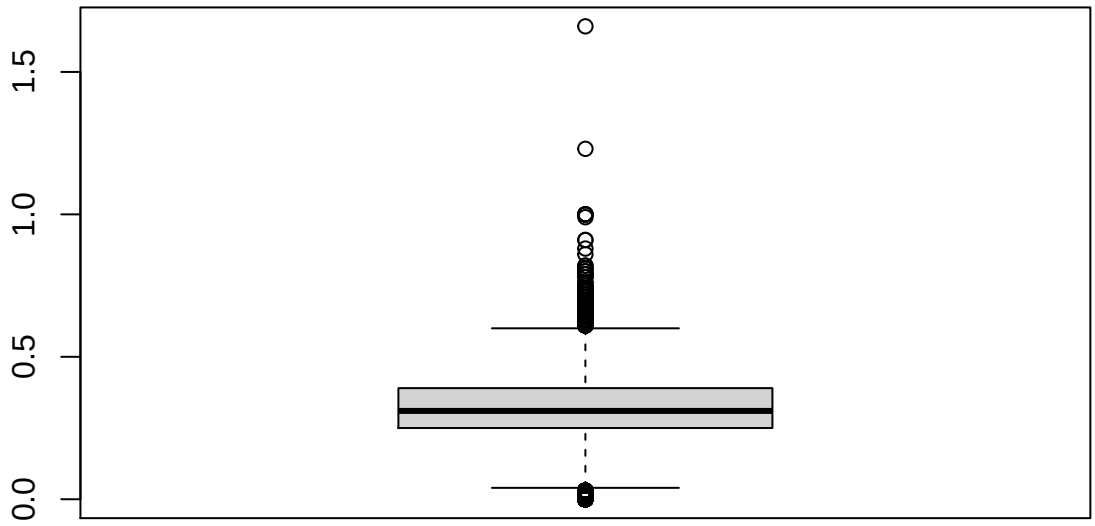


De los boxplot podemos ver que las variables citric.acid, residual.sugar,chlorides, free.sulfur.dioxide y density presentan claros valores extremos.

Al ser valores muy puntuales y muy extremos, no parece que puedan considerarse válidos en modo alguno y, por lo tanto, hay que identificarlos y suprimirlos.

Vamos a lozalizar los valores extremos de cada variable apra suprimirlo:

```
boxplot(wine$citric.acid)
```

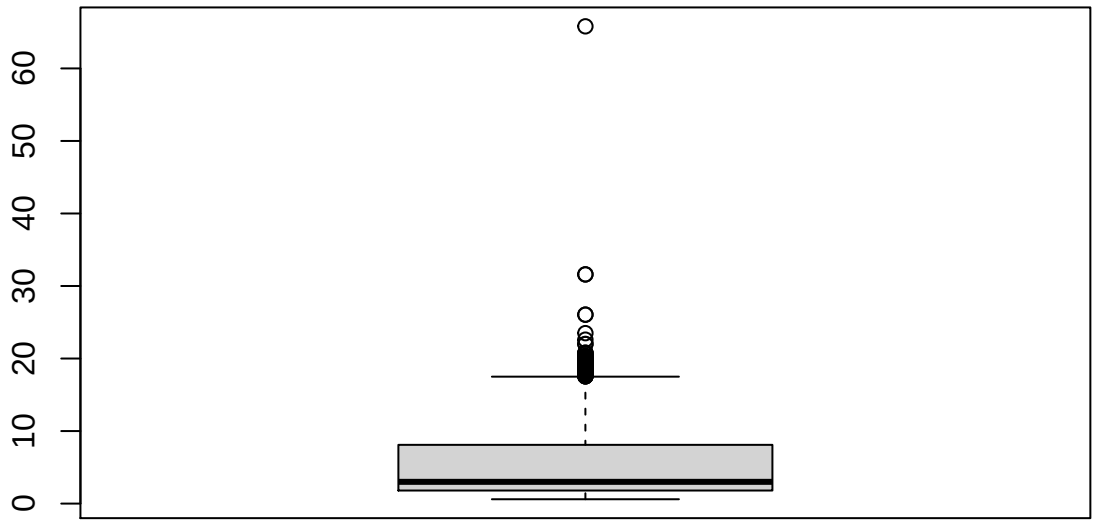



```
wine[which(wine$citric.acid > 1.2),]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 2345           7.4             0.20        1.66            2.1     0.022
## 4752           7.6             0.25        1.23            4.6     0.035
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 2345                  34                  113 0.99165 3.26      0.55    12.2
## 4752                  51                  294 0.99018 3.03      0.43    13.1
##      quality  type
## 2345       6 White
## 4752       6 White
```

Las lineas 2345 y 4572 contienen outlier

```
boxplot(wine$residual.sugar)
```

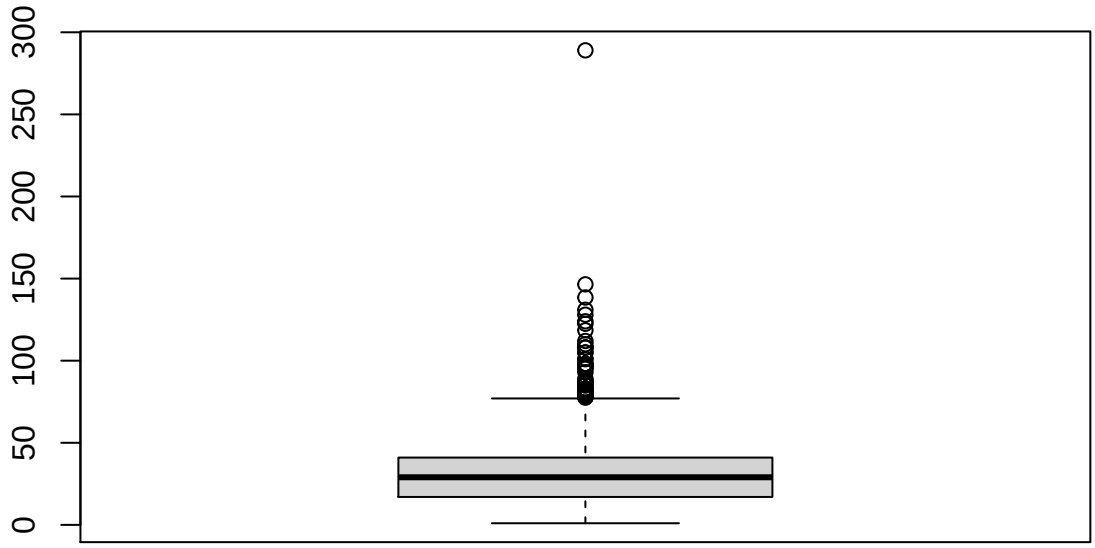



```
wine[which(wine$residual.sugar > 40),]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 4381           7.8            0.965         0.6           65.8     0.074
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 4381                   8                  160 1.03898 3.39      0.69    11.7
##      quality  type
## 4381       6 White
```

La linea 4381 contiene outlier

```
boxplot(wine$free.sulfur.dioxide)
```

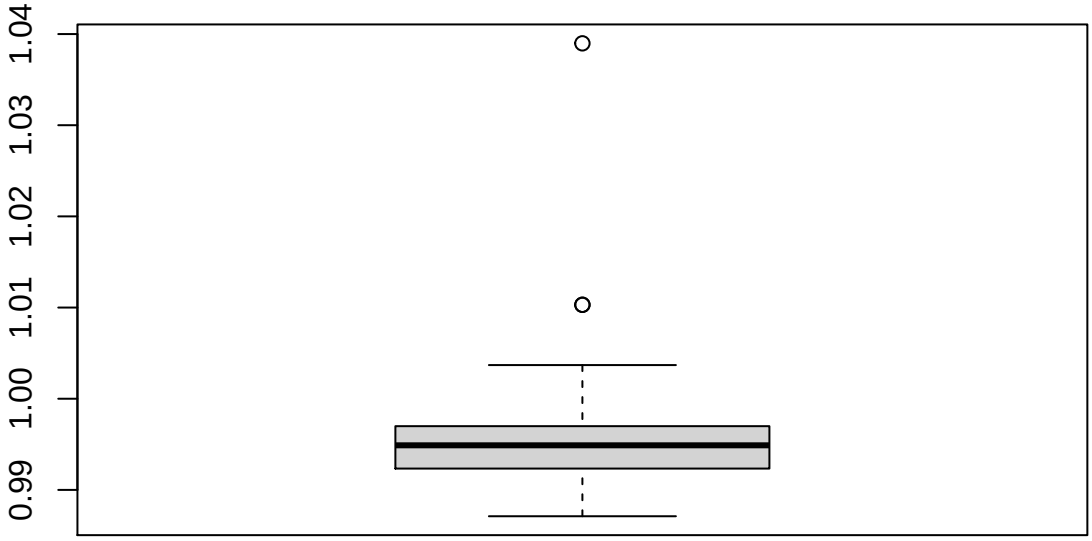



```
wine[which(wine$free.sulfur.dioxide > 200),]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 6345           6.1             0.26        0.25            2.9     0.047
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 6345                 289                  440 0.99314 3.44      0.64    10.5
##      quality  type
## 6345       3 White
```

6345 contiene outlier

```
boxplot(wine$density)
```




```
wine[which(wine$density > 1.02),]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 4381           7.8            0.965         0.6           65.8     0.074
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 4381                   8                  160 1.03898 3.39      0.69    11.7
##      quality  type
## 4381       6 White
```

La 4381 contiene outlier

Eliminamos los valores detectados como outlier

```
wine <- wine[-c(2345,4381, 4572, 6345 ),]
```

Exportamos el nuevo fichero una vez hechas las tareas de limpieza:

```
write.csv(wine, "Wine_clean.csv")
```

4. Análisis de los datos.

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

Hacemos una agrupación por tipo de vino (color)

```
wine.red <- wine[wine$type == "Red",]
wine.white <- wine[wine$type == "White",]
```

Realizamos una agrupación por calidad del vino, una nueva variable dicotómica que indique : 1 Buena calidad (quality>=7) y el resto 0 No buena calidad.

```
good_wine <-ifelse(test=wine$quality>=7,yes=1,no=0)
wine$good_wine=good_wine
quality<- as.factor(wine$quality)
```

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

Primero vemos el tamaño de la muestra que estamos manejando.

```
nrow(wine)
```

```
## [1] 6493
```

En primer lugar, por el Teorema Central del Límite, dado que tenemos una muestra con de un tamaño muy grande (n=6493), podemos asumir normalidad. No obstante vamos a aplicar también algún test de los diponibles en R, concretamente el test de Anderson-Darling disponible en el paquete "nortest".

```
library(nortest)
ad.test(wine[,1])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 1]
## A = 181.97, p-value < 2.2e-16
```

```
ad.test(wine[,2])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 2]
## A = 237.04, p-value < 2.2e-16
```

```
ad.test(wine[,3])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 3]
## A = 58.524, p-value < 2.2e-16
```

```
ad.test(wine[,4])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 4]
## A = 397.12, p-value < 2.2e-16
```

```
ad.test(wine[,5])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 5]
## A = 467.86, p-value < 2.2e-16
```

```
ad.test(wine[,6])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 6]
## A = 41.673, p-value < 2.2e-16
```

```
ad.test(wine[,7])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 7]
## A = 23.958, p-value < 2.2e-16
```

```
ad.test(wine[,8])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 8]
## A = 25.826, p-value < 2.2e-16
```

```
ad.test(wine[,9])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 9]
## A = 10.979, p-value < 2.2e-16
```

```
ad.test(wine[,10])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 10]
## A = 98.865, p-value < 2.2e-16
```

```
ad.test(wine[,11])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 11]
## A = 92.484, p-value < 2.2e-16
```

```
ad.test(wine[,12])
```

```
##
##  Anderson-Darling normality test
##
## data:  wine[, 12]
## A = 367.25, p-value < 2.2e-16
```

Como vemos, en todas las variables el p-valor $<$ 2.2e-16, por lo que no existe evidencia estadística para rechazar Ho y asumimos normalidad.

Pasando a la comprobación de la homocedasticidad (es decir, homogeneidad de la varianza.) realizando un test de Levene, dado que los datos se distribuyen normalmente, se comprueba la homogeneidad de la varianza de la variable calidad entre vinos tintos y blancos:

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
leveneTest(quality ~ as.factor(type), data = wine)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  2.3538  0.125
##       6491
```

El test no encuentra diferencias significativas entre las varianzas de los dos grupos y se concluye que la variable quality presenta varianzas estadísticamente homogéneas para vinos blancos y tintos.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

4.3.1. ¿Qué variables influyen más en la calidad?

Mediante un análisis de correlación, vamos a estudiar todas las variables para ver en qué grado influyen en la calidad del vino:

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable con respecto a "quality"
for (i in 1:(ncol(wine) - 2)) {
if (is.integer(wine[,i]) | is.numeric(wine[,i]))
    {
    spearman_test = cor.test(wine[,i],wine[,length(wine)-2],method = "spearman")
      corr_coef = spearman_test$estimate
      p_val = spearman_test$p.value
      # Añadimos resultado a la matriz
    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(wine)[i]
  }
}
```

```
## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(wine[, i], wine[, length(wine) - 2], method =
## "spearman"): Cannot compute exact p-value with ties
```

```r
a <- corr_matrix[,'p-value']
corr_matrix[order(a),]
```

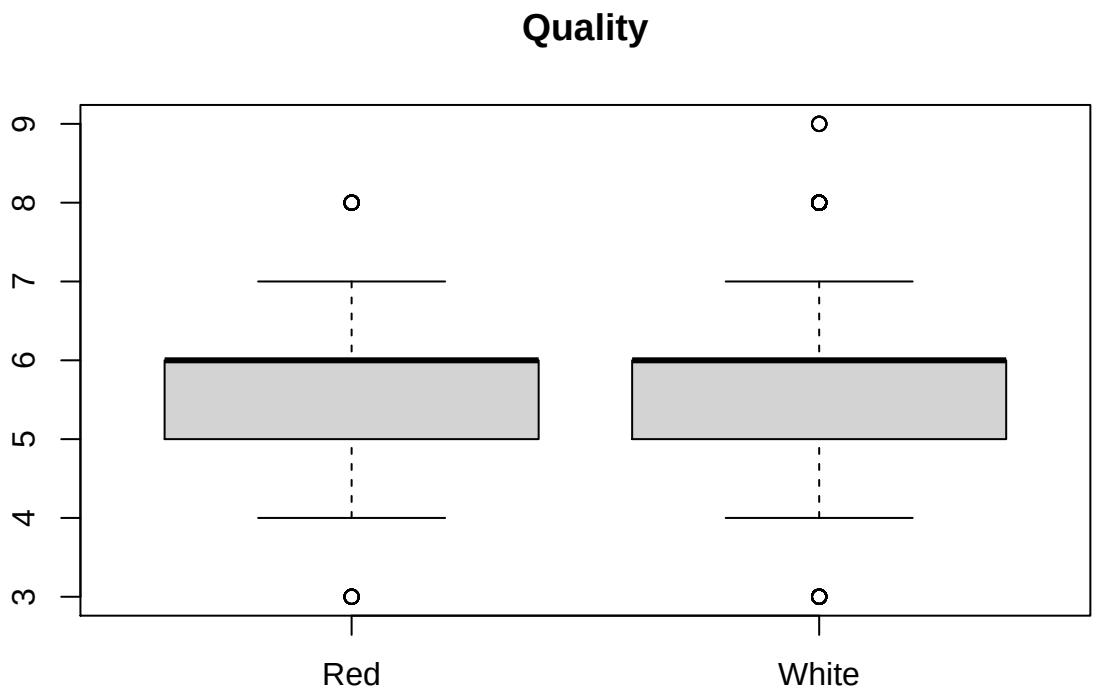
```
##                         estimate       p-value
## quality               1.00000000  0.000000e+00
## alcohol               0.44703974 1.251984e-316
## density              -0.32314771 1.080406e-157
## chlorides            -0.29504314 1.429544e-130
## volatile.acidity     -0.25809213  2.634262e-99
## citric.acid           0.10540675  1.656124e-17
## fixed.acidity        -0.09858185  1.699505e-15
## free.sulfur.dioxide   0.08746942  1.660362e-12
## total.sulfur.dioxide -0.05433877  1.181544e-05
## pH                    0.03292249  7.976211e-03
## sulphates             0.03007290  1.537893e-02
## residual.sugar       -0.01698474  1.711708e-01
```

La variable más correlacionada sería alcohol, por ser la próxima a 1 o -1. No obstante, los valores no son especialmente buenos por lo que no se podrían sacar conclusiones en base a esta parte del estudio.

4.3.2. ¿La calidad del vino blanco es superior a la del tinto?

Para responder a esta pregunta, quizá sea interesante ver como se distribuyen gráficamente en ambas variedades la calidad:

```
boxplot( wine.red$quality, wine.white$quality, names=c("Red","White"), main="Quality" )
```



Gráficamente no parece haber diferencia alguna pero vamos a comprobarlo con un método estadístico.

A través de un contraste de hipótesis vamos a determinar si la calidad del vino es superior dependiendo del tipo de vino del que se trate (tinto o blanco) con una confianza del 95%.

Hipótesis nula: la calidad de los vinos blancos es igual a la de los tintos.

Hipótesis alternativa: la calidad de los vinos blancos es superior a la de los tintos.

H0 : $\mu_{\text{white}} = \mu_{\text{red}}$

H1 : $\mu_{\text{white}} > \mu_{\text{red}}$

Realizamos un test de homoscedasticidad antes de aplicar el t.test:

```
var.test( wine.white$quality, wine.red$quality )
```

```
##
##  F test to compare two variances
##
## data:  wine.white$quality and wine.red$quality
## F = 1.2011, num df = 4893, denom df = 1598, p-value = 9.958e-06
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.107938 1.299878
## sample estimates:
## ratio of variances
##           1.201062
```

Interpretación del test de homoscedasticidad: El resultado nos da un p-valor muy pequeño. Por tanto, debemos rechazar la hipótesis nula de igualdad de varianzas. Debemos considerar que las varianzas son distintas.

Aplicamos el test:

```
t.test(wine.white$quality, wine.red$quality, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  wine.white$quality and wine.red$quality
## t = 10.172, df = 2949.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2031894       Inf
## sample estimates:
## mean of x mean of y
##  5.878423  5.636023
```

El p-valor obtenido (p-value < 2.2e-16) significa que podemos rechazar la hipótesis nula a favor de la hipótesis alternativa. Podemos concluir con un 95 % de nivel de confianza que los vinos blancos son significativamente mejores que los vinos tintos, en relación con su calidad.

4.3.3. Modelo de regresión lineal

```
library(rsample)
split.1 <- initial_split(wine, prop = 0.8, strata = "quality")
train.1 <- training(split.1)
test.1 <- testing(split.1)
```

```
mod.1 <- lm(quality~., train.1)
summary(mod.1)
```

```
##
## Call:
## lm(formula = quality ~ ., data = train.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.94893 -0.36595  0.00921  0.41581  2.00492
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.626e+01  1.282e+01   2.049   0.0405 *
## fixed.acidity         3.984e-03  1.360e-02   0.293   0.7695
## volatile.acidity     -1.067e+00  6.561e-02 -16.269  < 2e-16 ***
## citric.acid          -7.633e-02  6.485e-02  -1.177   0.2393
## residual.sugar        2.015e-02  5.118e-03   3.936 8.39e-05 ***
## chlorides            -4.588e-01  2.704e-01  -1.697   0.0898 .
## free.sulfur.dioxide   3.569e-03  6.240e-04   5.719 1.13e-08 ***
## total.sulfur.dioxide -6.336e-04  2.615e-04  -2.423   0.0154 *
## density              -2.207e+01  1.299e+01  -1.699   0.0893 .
## pH                    6.973e-02  7.576e-02   0.920   0.3574
## sulphates             2.902e-01  6.195e-02   4.684 2.89e-06 ***
## alcohol               1.225e-01  1.628e-02   7.521 6.37e-14 ***
```

```
## typeWhite            -2.588e-01  4.789e-02  -5.404 6.79e-08 ***
## good_wine             1.434e+00  2.065e-02  69.461  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5276 on 5179 degrees of freedom
## Multiple R-squared:  0.6377, Adjusted R-squared:  0.6368
## F-statistic: 701.2 on 13 and 5179 DF,  p-value: < 2.2e-16
```

Las variables con $\Pr(>|t|) > 0.05$ significa que no tienen significación en el modelo,por lo que nos fijamos en las que están señaladas con ***, que es una forma visual de ver el nivel de singificación. El valor de R-squared los indica qué porcentaje de varianza explica el modelo cerda de un 30%, que no es un resultado muy bueno.

Se puede concluir que no existe una relación lineal sólida entre las variables estudiadas y la calidad del vino, pues alrededor de un 70% de la varianza no está explicada por el modelo.

4.3.4. Modelo de regresión logística

Utilizando la variable dicotómica "good_wine" anteriormente creada y viendo las variables ue hasta ahora hemos encotnrado como significativas, vamos a construir un modlo e regresión logística

```
GLM.1 <- glm( wine$good_wine ~ alcohol + volatile.acidity + sulphates + citric.acid + fixed.acidity + cl
summary(GLM.1)
```

```
##
## Call:
## glm(formula = wine$good_wine ~ alcohol + volatile.acidity + sulphates +
##     citric.acid + fixed.acidity + chlorides + total.sulfur.dioxide +
##     density, family = binomial(logit), data = wine)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3533  -0.6361  -0.3839  -0.1782   3.1928
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.075e+02  2.315e+01  -4.644 3.42e-06 ***
## alcohol               9.831e-01  5.053e-02  19.456  < 2e-16 ***
## volatile.acidity     -4.120e+00  3.681e-01 -11.193  < 2e-16 ***
## sulphates             1.725e+00  2.544e-01   6.782 1.19e-11 ***
## citric.acid          -3.364e-01  3.490e-01  -0.964    0.335
## fixed.acidity         9.829e-03  3.859e-02   0.255    0.799
## chlorides            -1.254e+01  2.443e+00  -5.132 2.86e-07 ***
## total.sulfur.dioxide -1.005e-03  8.672e-04  -1.159    0.246
## density               9.716e+01  2.311e+01   4.205 2.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6437.9  on 6492  degrees of freedom
## Residual deviance: 5178.3  on 6484  degrees of freedom
## AIC: 5196.3
##
## Number of Fisher Scoring iterations: 6
```

```
wine$prob_qualityM=predict(GLM.1, wine, type="response")
newdatarisk=subset(wine, prob_qualityM>0.7)
Q3 <-quantile(wine$alcohol)[4]
alcohol <- which(newdatarisk$alcohol>Q3)
```

Calculamos el área bajo la curva ROC para comprobar la calidad del modelo:

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
g=roc(wine$good_wine,wine$prob_qualityM, data=wine)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(g)
```

```
## Area under the curve: 0.804
```

El área bajo la curva ROV es de 0,804. Cuanto mas se acerque a 1, mejor es el modelo, por lo que este model esbastante bueno.

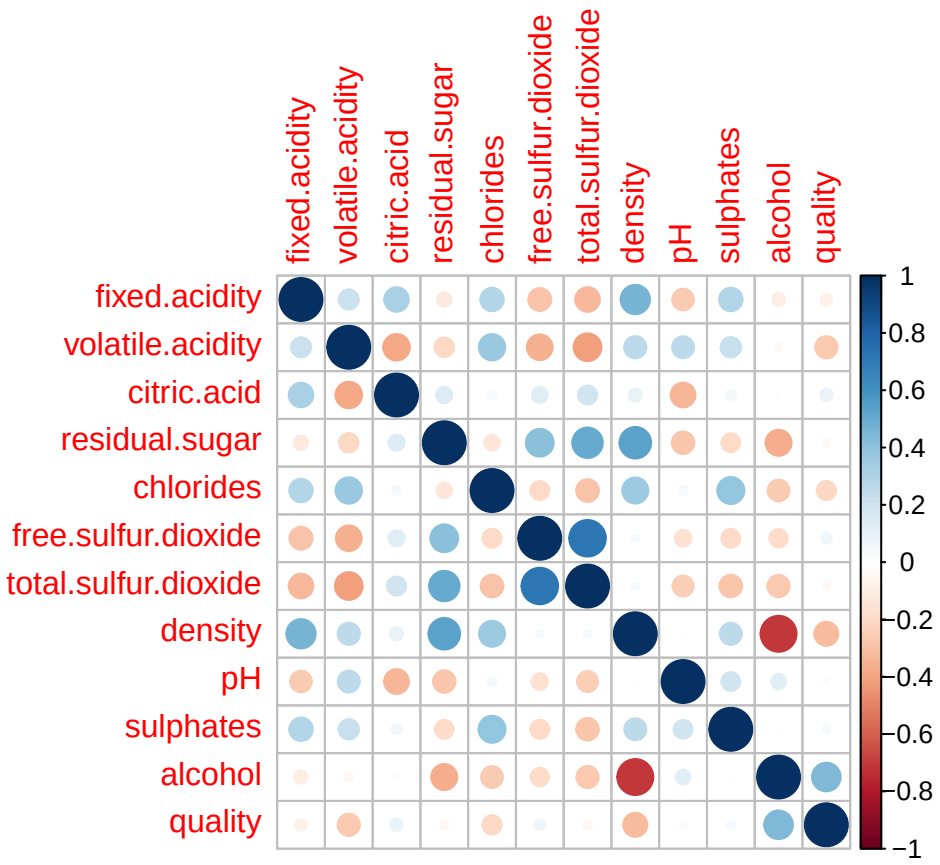
5. Representación de los resultados a partir de tablas y gráficas.

Representamos gráficamente las correlaciones entre variables al hilo de lo estudiado en el primer punto:

```
library(corrplot)
```

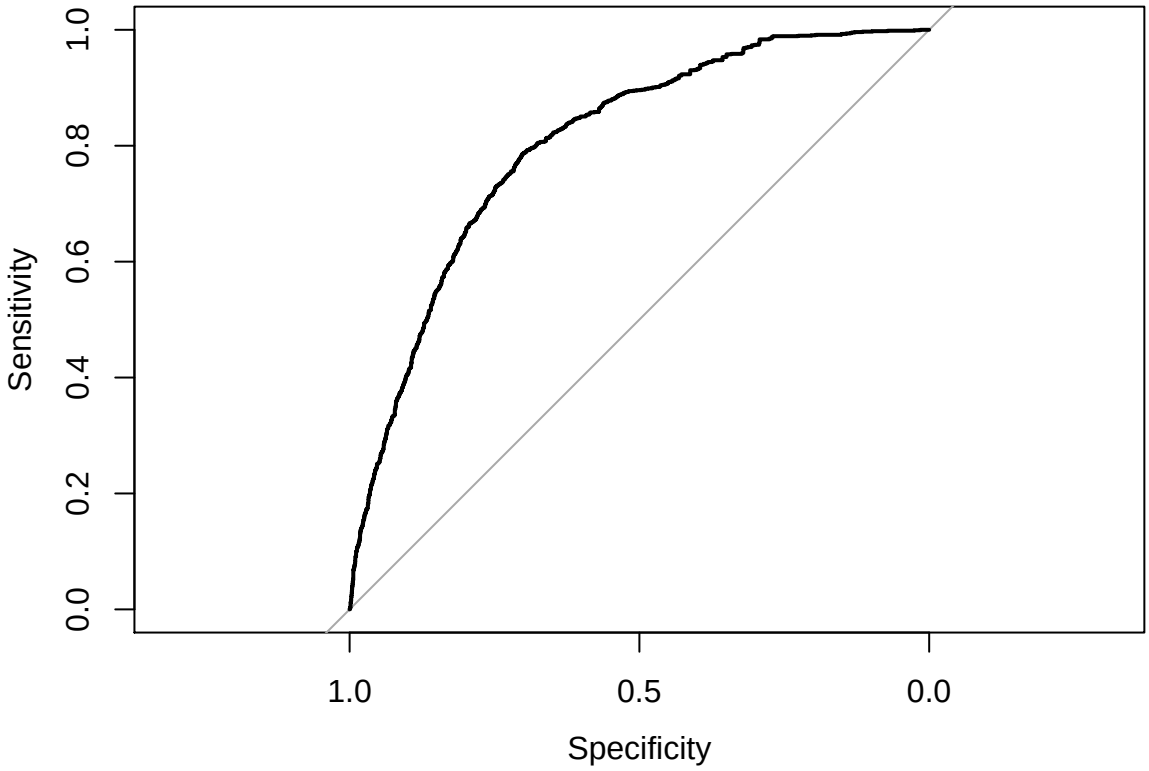
```
## corrplot 0.88 loaded
```

```
corrplot(cor(wine[1:12]))
```



Representamos gráficamente la curva ROC del modelo de regresión logística:

```
plot(g)
```




```
auc(g)
```

```
## Area under the curve: 0.804
```

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

Por una parte, hemos llegado a la conclusión, mediante un contraste de hipótesis, de que la calidad de los vinos blancos es superior a la de los tintos. También hemos llegado a la conclusión de que el alcohol es determinante en la calidad del vino mediante su correlación y, posteriormente a través del modelo de regresión logística hemos también encontrado que determina si un vino es de buna calidad o no, junto a otras variables que han resultado explicativas de dicha cualidad.