PRÁCTICA 2 - TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

Manuel Cubertorer Gumbau y Francisco Javier Corrales Estrella

11/12/2021

Contents

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

Los conjuntos de datos corresponden a una serie de registros de tipos de vino, obtenidos a partir de:

https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/

El dataset está formado por un total de 1599 registros de vino rojo y 4898 registros de vino blanco, y por 12 variables fisicoquímicas. Se definen una serie atributos como la acidez o la graduación, y una variable target con la calidad del vino. Extraeremos los dos dataset disponibles, uno para vinos blancos y otros para vinos tintos, y los fusionaremos en uno solo creando una variable categórica para el tipo de vino, el resto de variables son numéricas.

Los campos de los que se compone el dataset son los siguientes:

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

- **volatile acidity**: Cantidad de ácido acético en el vino, que en niveles demasiado altos puede llevar a un sabor desagradable a vinagre.
- **citric acid**: En pequeñas cantidades, el ácido cítrico puede agregar "frescura" y sabor a los vinos.
- **residual sugar**: Cantidad de azúcar residual después de la fermentación. Es raro encontrar vinos con menos de 1 g/l y los vinos con más de 45 g/l se consideran dulces.
- **chlorides**: Cantidad de sal en el vino.
- **free sulfur dioxide**: En estado natural, el SO2 presenta un equilibrio entre el SO2 molecular (como un gas disuelto) y el ion bisulfito. Previene el crecimiento microbiano y la oxidación del vino.
- **total sulfur dioxide**: Cantidad de formas libres y ligadas de SO2. En bajas concentraciones, el SO2 es mayormente indetectable en el vino, pero a concentraciones de SO2 libres superiores a 50 ppm, el SO2 se hace evidente en el olfato y también en el sabor del vino.
- **density**: Densidad del agua según el porcentaje de alcohol y contenido en azúcar.
- **pH**: Describe el grado de acidez o basicidad del vino en una escala de 0 (muy ácido) a 14 (muy básico). La mayoría de los vinos están entre 3 y 4 en la escala de pH.
- **sulphates**: Aditivo para vinos que puede contribuir a los niveles de gas de SO2, que actúa como antimicrobiano y antioxidante.
- **alcohol**: Porcentaje de alcohol en el vino.
- **quality**: Indica la calidad del vino en una escala del 1 al 10.
- **tipo_vino**: Variable categórica que distingue entre vinos blancos y vinos tintos.

Nuestro análisis, tratará de determinar que variable/s son más determinantes en la calidad del vino, y compararemos cómo influye en algunas de ellas el tipo de vino (blanco o tinto).

Este tipo de análisis son muy relevantes en el mundo de las bodegas y los vinos donde se utilizan estos datos para realizar investigaciones sobre la calidad de los vinos, las uvas y sus cualidades fisicoquímicas.

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

Primero cargamos los datos desde el repositorio de datatsets UCI Machine Learning. Luego creamos la variable "tipo" que nos indique el tipo de vino (blanco o tinto) y juntamos los dos datasets en uno.

Una **consideración importante**, es que no fusionaremos realmente los dos datasets hasta que hayamos completado las tareas de limpieza y preparación de datos pues no queremos que las distribuciones de los datos se mezclen, por ejemplo, los valores extremos los queremos tratar separados por cada tipo de vino.

```
red_wine_data <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequa
white_wine_data <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winec

colnames(red_wine_data) <- c("fixed_acidity","volatile_acidity", "citric_acid", "residual_sugar", "chlo

colnames(white_wine_data) <- c("fixed_acidity","volatile_acidity", "citric_acid", "residual_sugar", "ch

red_wine_data$tipo <- 'tinto'
white_wine_data$tipo <- 'blanco'

wine_data = rbind(white_wine_data, red_wine_data)
```

Ahora vamos a mostrar las primeras líneas del dataset para comprobar que se ha cargado correctamente.

```
head(wine_data, 10)
```

```
##    fixed_acidity volatile_acidity citric_acid residual_sugar chlorides
## 1            7.0             0.27        0.36           20.7     0.045
## 2            6.3             0.30        0.34            1.6     0.049
## 3            8.1             0.28        0.40            6.9     0.050
## 4            7.2             0.23        0.32            8.5     0.058
```

```
## 5            7.2             0.23        0.32            8.5     0.058
## 6            8.1             0.28        0.40            6.9     0.050
## 7            6.2             0.32        0.16            7.0     0.045
## 8            7.0             0.27        0.36           20.7     0.045
## 9            6.3             0.30        0.34            1.6     0.049
## 10           8.1             0.22        0.43            1.5     0.044
##    free_sulfur_dioxide total_sulfur_dioxide density   ph sulphates alcohol
## 1                   45                  170  1.0010 3.00      0.45     8.8
## 2                   14                  132  0.9940 3.30      0.49     9.5
## 3                   30                   97  0.9951 3.26      0.44    10.1
## 4                   47                  186  0.9956 3.19      0.40     9.9
## 5                   47                  186  0.9956 3.19      0.40     9.9
## 6                   30                   97  0.9951 3.26      0.44    10.1
## 7                   30                  136  0.9949 3.18      0.47     9.6
## 8                   45                  170  1.0010 3.00      0.45     8.8
## 9                   14                  132  0.9940 3.30      0.49     9.5
## 10                  28                  129  0.9938 3.22      0.45    11.0
##    quality   tipo
## 1        6 blanco
## 2        6 blanco
## 3        6 blanco
## 4        6 blanco
## 5        6 blanco
## 6        6 blanco
## 7        6 blanco
## 8        6 blanco
## 9        6 blanco
## 10       6 blanco
```

A continuación mostraremos la estructura de los datos. Donde comprobamos que todas las variables son numéricas, excepto la variable categórica **tipo** que hemos creado.

```
str(wine_data)
```

```
## 'data.frame':    6497 obs. of  13 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 ...
##  $ tipo                : chr  "blanco" "blanco" "blanco" "blanco" ...
```

También vemos que la variable **quality** es del tipo integer, así que la transformaremos a tipo numeric para que sea completamente compatible con el resto de variables y evitar posibles conflictos.

```
# Convertimos la columna "quality" a tipo numeric
wine_data$quality<-as.numeric(wine_data$quality)
class(wine_data$quality)
```

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

Por otra parte, también vemos que existen dos variables para definir la acidez (fixed_acidity y volatile_acidity). Así pues podemos crear una variable nueva llamada **acidity** que recoja la suma de estas dos y por tanto indique la acidez total del vino.

```
# Creamos la nueva columna acidity
wine_data$acidity<-wine_data$fixed_acidity + wine_data$volatile_acidity
# Eliminamos las columnas fixed_acidity y volatile_acidity
wine_data <- wine_data[, -(1:2)]
wine_data <- subset(wine_data, select=c(12,1:11))
head(wine_data)
```

```
##   acidity citric_acid residual_sugar chlorides free_sulfur_dioxide
## 1    7.27        0.36           20.7     0.045                  45
## 2    6.60        0.34            1.6     0.049                  14
## 3    8.38        0.40            6.9     0.050                  30
## 4    7.43        0.32            8.5     0.058                  47
## 5    7.43        0.32            8.5     0.058                  47
## 6    8.38        0.40            6.9     0.050                  30
##   total_sulfur_dioxide density   ph sulphates alcohol quality   tipo
## 1                  170  1.0010 3.00      0.45     8.8       6 blanco
## 2                  132  0.9940 3.30      0.49     9.5       6 blanco
## 3                   97  0.9951 3.26      0.44    10.1       6 blanco
## 4                  186  0.9956 3.19      0.40     9.9       6 blanco
## 5                  186  0.9956 3.19      0.40     9.9       6 blanco
## 6                   97  0.9951 3.26      0.44    10.1       6 blanco
```

Estadísticas principales de los datos:

```
summary(wine_data)
```

```
##     acidity        citric_acid     residual_sugar     chlorides
##  Min.   : 4.110   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900
##  1st Qu.: 6.710   1st Qu.:0.2500   1st Qu.: 1.800   1st Qu.:0.03800
##  Median : 7.300   Median :0.3100   Median : 3.000   Median :0.04700
##  Mean   : 7.555   Mean   :0.3186   Mean   : 5.443   Mean   :0.05603
##  3rd Qu.: 8.050   3rd Qu.:0.3900   3rd Qu.: 8.100   3rd Qu.:0.06500
##  Max.   :16.285   Max.   :1.6600   Max.   :65.800   Max.   :0.61100
##  free_sulfur_dioxide total_sulfur_dioxide    density             ph
##  Min.   :  1.00      Min.   :  6.0        Min.   :0.9871   Min.   :2.720
##  1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923   1st Qu.:3.110
##  Median : 29.00      Median :118.0        Median :0.9949   Median :3.210
##  Mean   : 30.53      Mean   :115.7        Mean   :0.9947   Mean   :3.219
##  3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970   3rd Qu.:3.320
##  Max.   :289.00      Max.   :440.0        Max.   :1.0390   Max.   :4.010
##    sulphates         alcohol         quality          tipo
##  Min.   :0.2200   Min.   : 8.00   Min.   :3.000   Length:6497
##  1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000   Class :character
##  Median :0.5100   Median :10.30   Median :6.000   Mode  :character
##  Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :2.0000   Max.   :14.90   Max.   :9.000
```

3. Limpieza de los datos

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

Mostramos las estadísticas de valores vacíos o nulos.

```
# Estadísticas de valores vacíos
colSums(is.na(wine_data))
```

```
##              acidity          citric_acid       residual_sugar
##                    0                    0                    0
##            chlorides  free_sulfur_dioxide total_sulfur_dioxide
##                    0                    0                    0
##              density                   ph            sulphates
##                    0                    0                    0
##              alcohol              quality                 tipo
##                    0                    0                    0
```

```
colSums(wine_data=="")
```

```
##              acidity          citric_acid       residual_sugar
##                    0                    0                    0
##            chlorides  free_sulfur_dioxide total_sulfur_dioxide
##                    0                    0                    0
##              density                   ph            sulphates
##                    0                    0                    0
##              alcohol              quality                 tipo
##                    0                    0                    0
```

En nuestro caso no existen valores nulos, si los hubiese la alternativa mas sencilla es setear el valor de la media para todo el conjunto de datos. Esto lo podemos mejorar tomando alguna medida de tendencia central dependiendo de la distribución de los datos, esto se puede hacer para toda la muestra o en función de alguna variable categórica, en nuestro caso el tipo de vino.

Existen otros métodos, como kNN que se basa en la similitud, básicamente se fija en que valores tiene esa variable en los "vecinos" mas cercanos, donde definimos cuantos vecinos queremos tomar.´

Otro análisis que vamos a realizar para la limpieza de los datos es detectar si existen valores duplicados. Los valores duplicados no aportan ninguna información adicional y se deberían eliminar para mayor integridad de los datos.

```
library("dplyr")
# Detección y eliminación de valores duplicados
sum(duplicated(wine_data))
```

```
## [1] 1177
```

```
wine_data <- distinct(wine_data)
```

Con estos cambios el total de registros que tenemos ahora es de:

```
dim(wine_data)
```

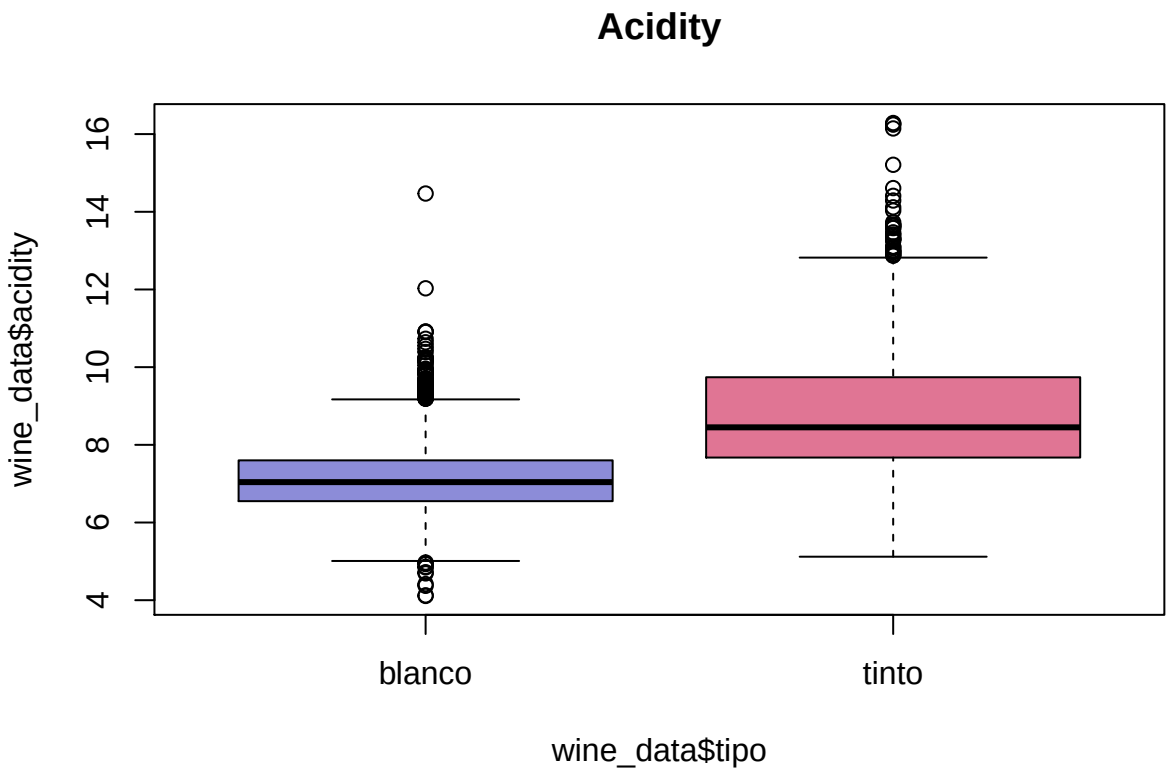
```
## [1] 5320   12
```

3.2 Identificación y tratamiento de valores extremos

Consideramos valores extremos como aquellos valores que son sospechosos por alejarse demasiado del resto de datos, esto en términos numéricos quiere decir, que están demasiado alejados de la media teniendo en cuenta

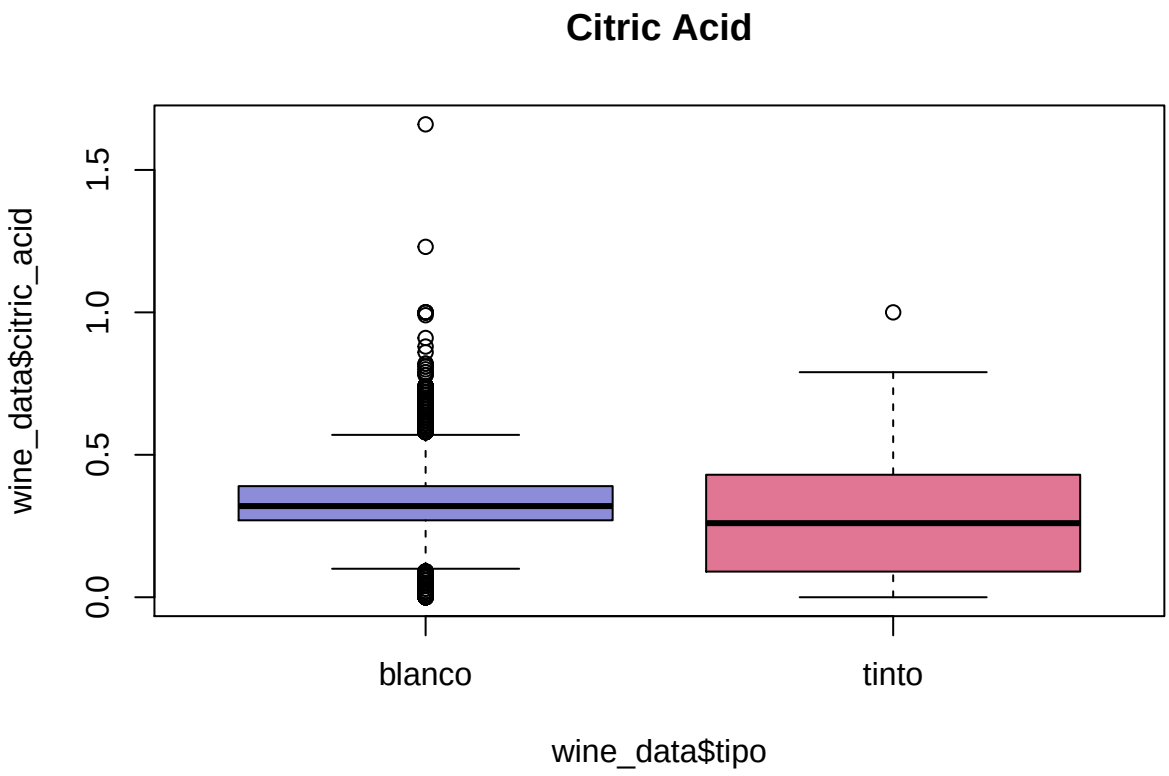
la desviación típica. Podemos hacer esta aproximación de una forma visual mediante un gráfico de caja o calculando los valores fuera del rango intercuartílico, podemos usar la función boxplots.stats() para esto.

```
myColors <- c(rgb(0.1,0.1,0.7,0.5) ,rgb(0.8,0.1,0.3,0.6))

tintos <- subset(wine_data, wine_data$tipo == "tinto")
blancos <- subset(wine_data, wine_data$tipo == "blanco")

boxplot(wine_data$acidity ~ wine_data$tipo, main="Acidity", col =myColors )
```




```
out_ac_tinto <- boxplot(tintos$acidity, plot=FALSE)$out
out_ac_blanco <- boxplot(blancos$acidity, plot=FALSE)$out
```

```
boxplot(wine_data$citric_acid ~ wine_data$tipo, main="Citric Acid", col =myColors )
```

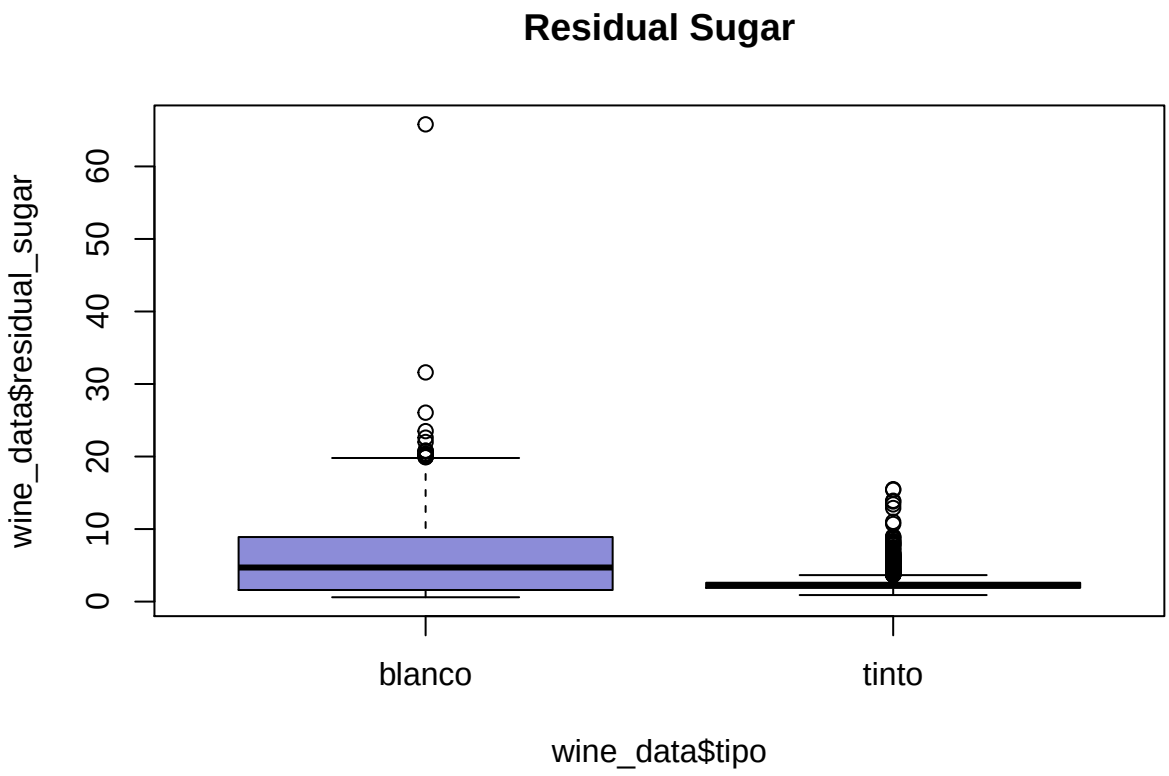



```
out_ca_tinto <- boxplot(tintos$citric_acid, plot=FALSE)$out
out_ca_blanco <- boxplot(blancos$citric_acid, plot=FALSE)$out
```

Eliminamos solo el valor que es de tipo blanco y están por encima de 1.5

```
wine_data <- wine_data[-which(wine_data$citric_acid > 1.5 & wine_data$tipo =='blanco'),]
```

```
boxplot(wine_data$residual_sugar ~ wine_data$tipo, main="Residual Sugar", col =myColors )
```

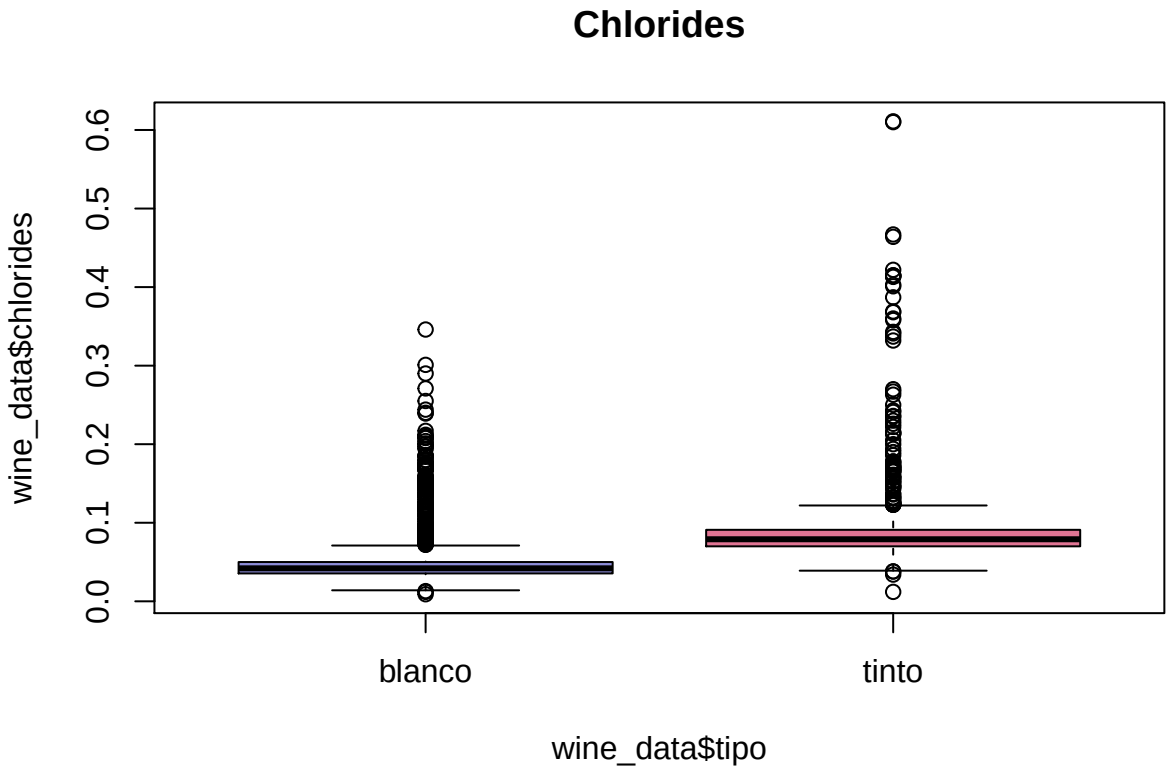



```
out_rs_tinto <- boxplot(tintos$residual_sugar, plot=FALSE)$out
out_rs_blanco <- boxplot(blancos$residual_sugar, plot=FALSE)$out
```

Eliminamos solo de sugar el valor que es de tipo blanco y están por encima de 60

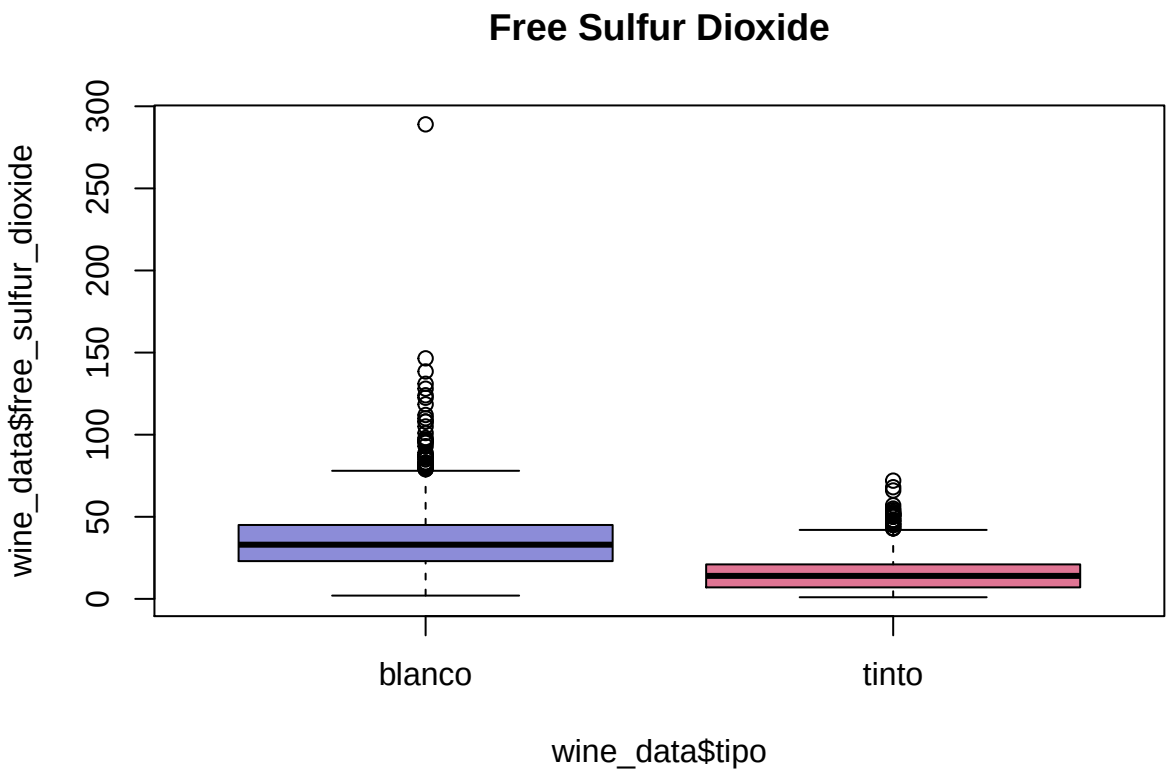
```
wine_data <- wine_data[-which(wine_data$residual_sugar > 60 & wine_data$tipo =='blanco'),]
```

```
boxplot(wine_data$chlorides ~ wine_data$tipo, main="Chlorides", col =myColors )
```




```
out_ch_tinto <- boxplot(tintos$chlorides, plot=FALSE)$out
out_ch_blanco <- boxplot(blancos$chlorides, plot=FALSE)$out
```

```
boxplot(wine_data$free_sulfur_dioxide ~ wine_data$tipo, main="Free Sulfur Dioxide", col =myColors )
```

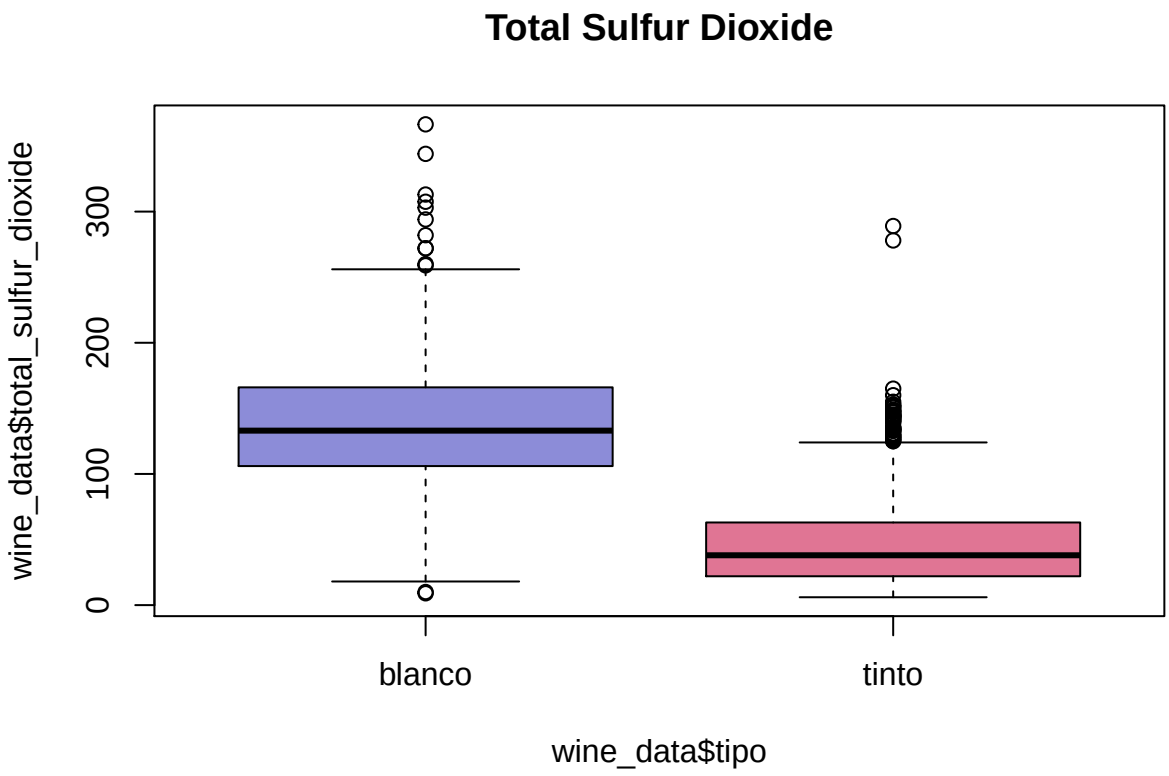



```
out_fsd_tinto <- boxplot(tintos$free_sulfur_dioxide, plot=FALSE)$out
out_fsd_blanco <- boxplot(blancos$free_sulfur_dioxide, plot=FALSE)$out
```

Eliminamos solo de free_sulfur_dioxide el valor que es de tipo blanco y están por encima de 250

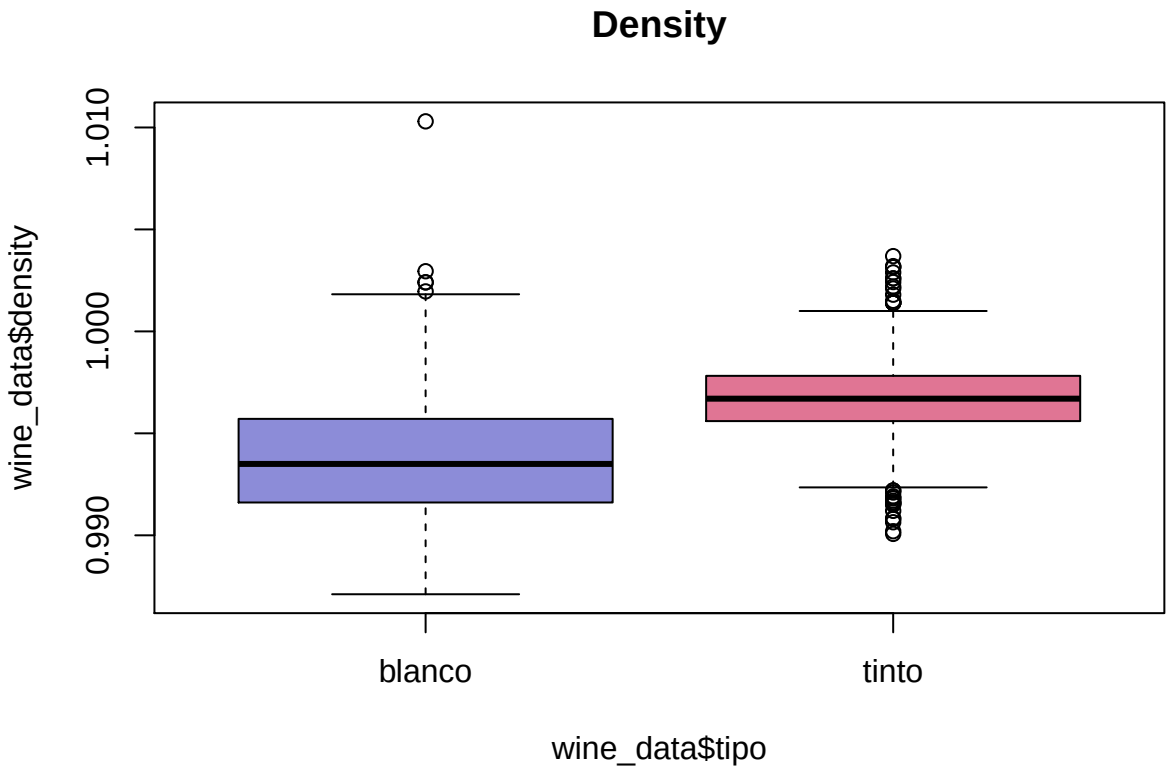
```
wine_data <- wine_data[-which(wine_data$free_sulfur_dioxide > 250 & wine_data$tipo =='blanco'),]
```

```
boxplot(wine_data$total_sulfur_dioxide ~ wine_data$tipo, main="Total Sulfur Dioxide", col =myColors )
```




```
out_tsd_tinto <- boxplot(tintos$total_sulfur_dioxide, plot=FALSE)$out
out_tsd_blanco <- boxplot(blancos$total_sulfur_dioxide, plot=FALSE)$out
```

```
boxplot(wine_data$density ~ wine_data$tipo, main="Density", col =myColors )
```

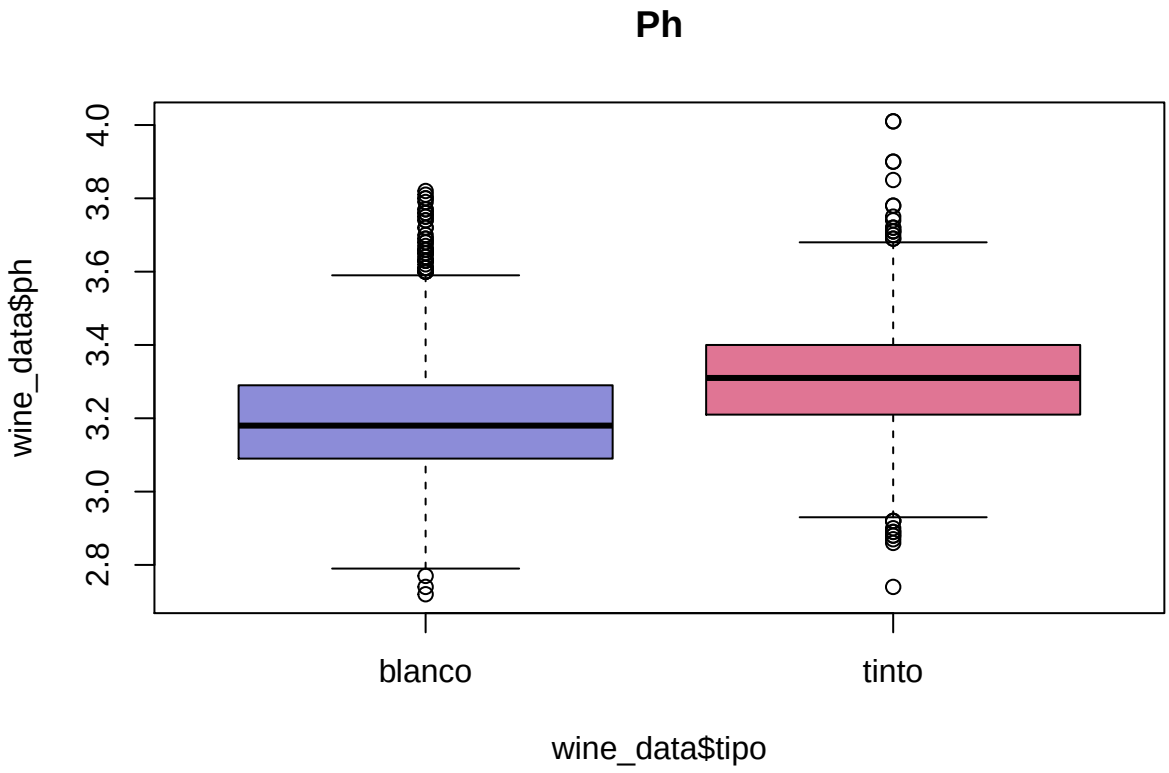



```
out_de_tinto <- boxplot(tintos$density, plot=FALSE)$out
out_de_blanco <- boxplot(blancos$density, plot=FALSE)$out
```

Eliminamos solo de density el valor que es de tipo blanco y están por encima de 250

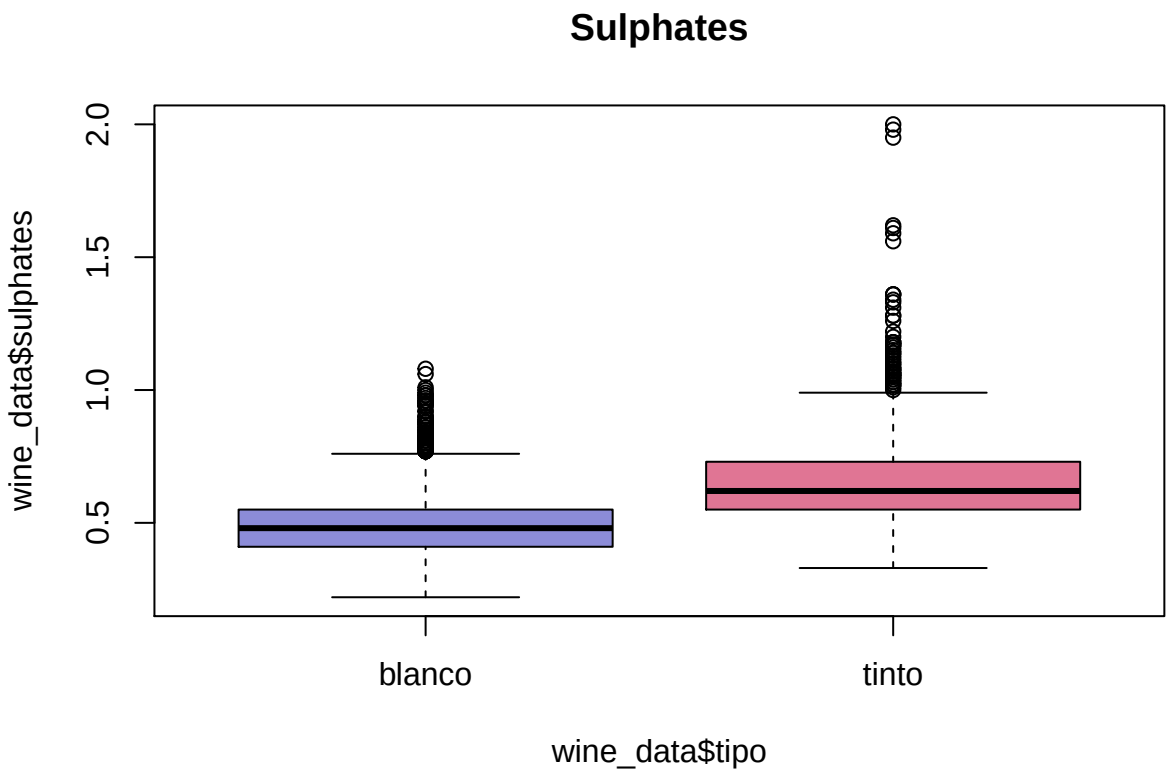
```
wine_data <- wine_data[-which(wine_data$density > 1.005 & wine_data$tipo =='blanco'),]
```

```
boxplot(wine_data$ph ~ wine_data$tipo, main="Ph", col =myColors )
```




```
out_ph_tinto <- boxplot(tintos$ph, plot=FALSE)$out
out_ph_blanco <- boxplot(blancos$ph, plot=FALSE)$out
```

```
boxplot(wine_data$sulphates ~ wine_data$tipo, main="Sulphates", col =myColors )
```

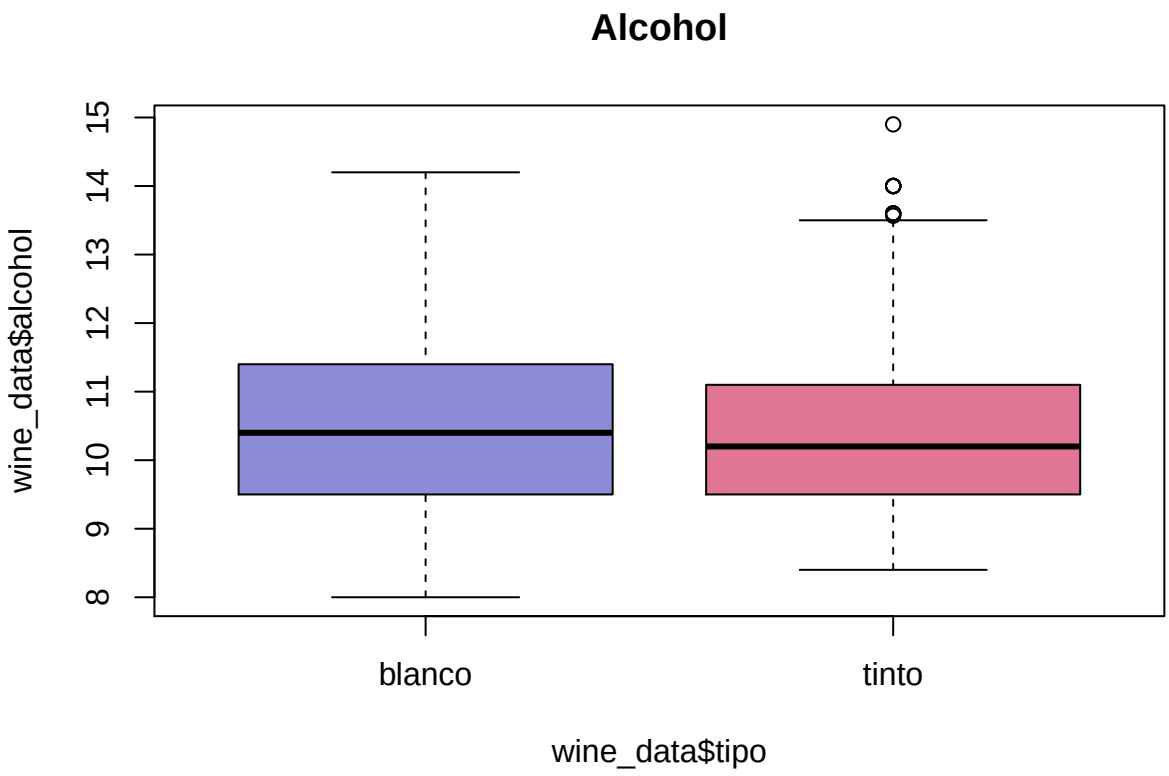



```
out_su_tinto <- boxplot(tintos$sulphates, plot=FALSE)$out
out_su_blanco <- boxplot(blancos$sulphates, plot=FALSE)$out
```

Eliminamos solo de free_sulfur_dioxide el valor que es de tipo tinto y están por encima de 1.7

```
wine_data <- wine_data[-which(wine_data$sulphates > 1.7 & wine_data$tipo =='tinto'),]
```

```
boxplot(wine_data$alcohol ~ wine_data$tipo, main="Alcohol", col =myColors )
```




```
out_al_tinto <- boxplot(tintos$alcohol, plot=FALSE)$out
out_al_blanco <- boxplot(blancos$alcohol, plot=FALSE)$out
```

Como hemos podido observar en los resultados anteriores, en todas las variables hay algunos valores atípicos, dado que algunos de estos valores seguramente sean válidos y se correspondan con la realidad, y como no tenemos el conocimiento suficiente para saberlo al cien por cien, solo hemos decidido eliminar aquellos que son excepcionalmente altos.

4. Análisis de los datos

Nuestro análisis de datos está orientado a saber si hay diferencias en la calidad entre los vinos blancos y los vinos tintos, conocer que variables influyen mas en la calidad del vino y saber si podemos predecir y con que garantia que calidad tendrá un vino en función de sus atributos.

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

Los grupos de datos que vamos a seleccionar para su análisis son la calidad con el tipo de vino, la calidad con el alcohol, el tipo de vino con el alcohol.

Para poder comparar dos grupos en primer lugar comprobaremos si se cumplen los criterios de normalizacion y homogeneidad de la varianzas, en caso afirmativo podremos aplicar una prueba de tipo parámetrico como t-student, en caso contrario aplicaremos una prueba de Mann-Whitney, ya que los dos grupos son independientes.

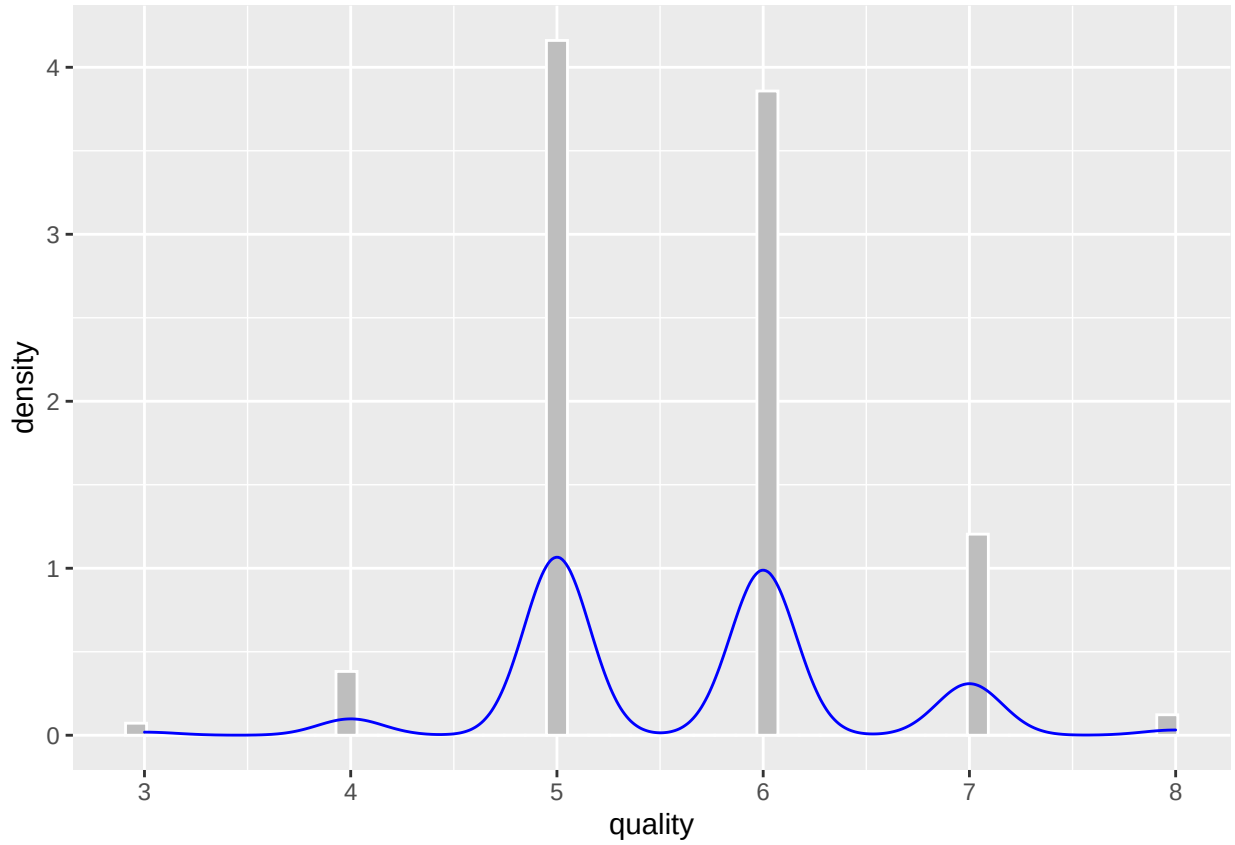
```
#Calidad de lso vinos por tipo
calidad_tintos = tintos$quality
```

```
calidad_blancos = blancos$quality

#Creamos una nueva variable con el vino categorizado
wine_data$calidad_vino <- factor(wine_data$quality)
```

4.2 Comprobación de la normalidad y homogenidad de la varianza

Queremos aplicar un contraste de hipótesis sobre la calidad de los vinos blancos y tintos, dependiendo de si los grupos cumplen los criterios de normalidad y homocedasticidad podremos aplicar un tipo de prueba o no. Para comprobar la **normalidad** de una serie a veces es suficiente una inspección visual. Hay que tener en cuenta en este caso que la calidad solo toma valores enteros.

```
library(ggplot2)
ggplot(tintos, aes(x = quality)) +
geom_histogram(aes(y = ..density..), bins = 50, color = "white", fill = "grey") +
geom_density(color = "blue")
```



Otra forma de comprobarlo es mediante un test de Kolmogorov-Smirnov o Shapiro-Wilk

```
#Kolmogorov-Smirno
ks.test(calidad_tintos, pnorm, mean(calidad_tintos), sd(calidad_tintos))
```

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

```
##
##  One-sample Kolmogorov-Smirnov test
##
```

```
## data:  calidad_tintos
## D = 0.24634, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

O con Shapiro-Wilk

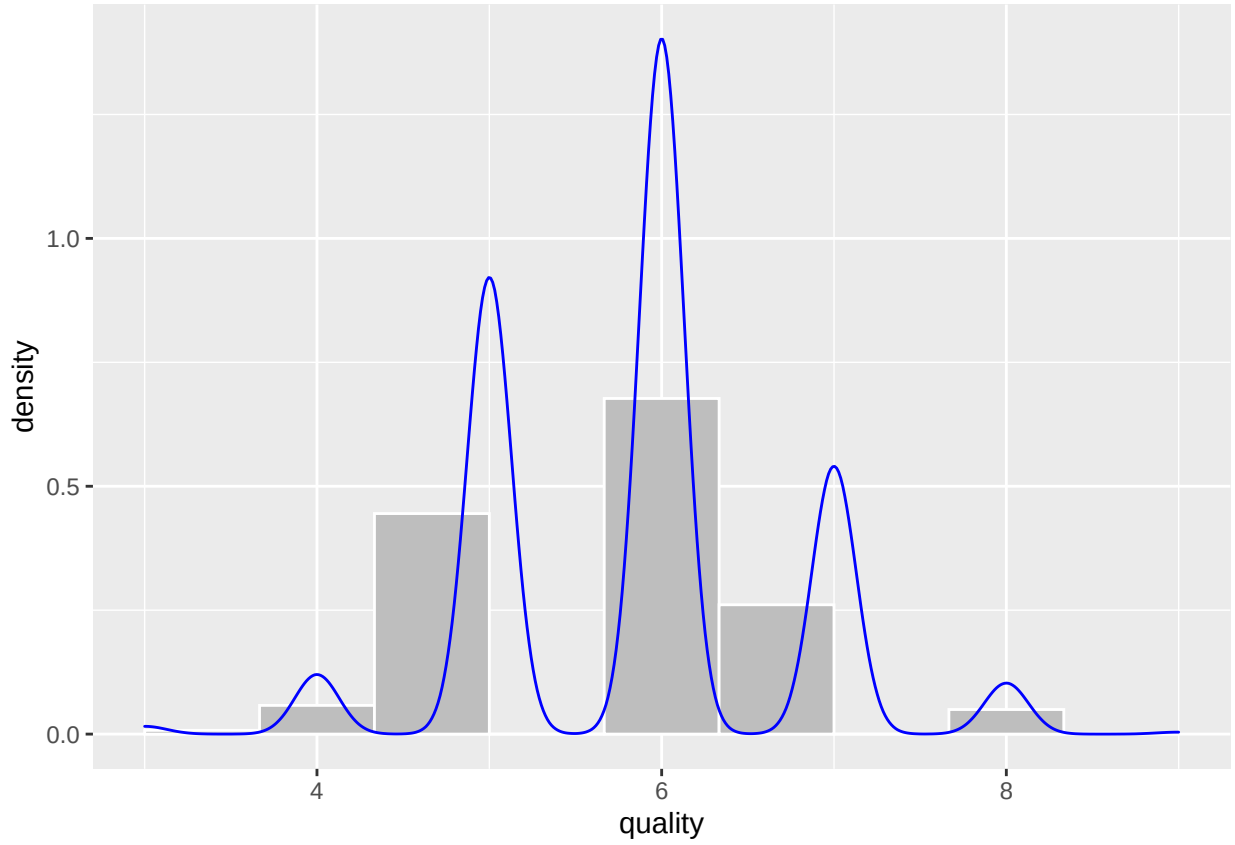
```
#Shapiro-Wilk
shapiro.test(calidad_tintos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  calidad_tintos
## W = 0.86398, p-value < 2.2e-16
```

Con ambas pruebas, claramente rechazaríamos la hipotesis nula ya que el p-valores es menor que 0.05, y no podemos asumir la normalidad de la serie. Pero por el teorema del limite central dado que existe un numero suficiente de registros si podemos asumir la normalidad de la media muestral.

La calidad de los vinos blancos sigue la siguiente distribución:

```
library(ggplot2)
ggplot(blancos, aes(x = quality)) +
geom_histogram(aes(y = ..density..), bins = 10, color = "white", fill = "grey") +
geom_density(color = "blue")
```



Y según el test de Shapiro-Wilk tampoco podemos asumir la normalidad

```
#Shapiro-Wilk
shapiro.test(calidad_blancos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  calidad_blancos
## W = 0.89113, p-value < 2.2e-16
```

Como nuestro conjunto de datos no cumple los criterios de normalidad para comprobar la homogeneidad de la varianza usaremos un test de Fligner-Killeen

```
fligner.test(quality ~ tipo, data = wine_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by tipo
## Fligner-Killeen:med chi-squared = 1.7566, df = 1, p-value = 0.185
```

El p-valor es mayor al nivel de significancia, por que rechazamos la hipotesis alternativa de la que las varianzas son distintas, por lo que aceptamos la igualdad de las varianzas.

La otra serie sobre la que queremos comprobar la normalidad es el alcohol

```
#Aplicamos test de Kolmogorox-Smirnof
ks.test(wine_data$alcohol, pnorm, mean(wine_data$alcohol), sd(wine_data$alcohol))
```

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

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  wine_data$alcohol
## D = 0.092714, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Aunque el p-valor es inferior al nivel de significancia y rechazariamos la hipotesis nula de que se cumple la normalidad, por el teorema del limite centrar asumiremos que la media muestral sigue una distribución normal para aplicar las pruebas paramétricas correspondientes.

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.

Vamos a aplicar las siguientes pruebas estadísticas.

4.3.1 Comparación de dos grupos. Queremos hacer un contraste de hipótesis sobre si los vinos blancos tienen la misma caildad que los vinos tintos.

Definimos una función para ver los intervalos de confianza de cada serie.

```
miIC.tstudent <- function(col, NC){
  alfa <- 1 - NC/100
  n <- length(col)
  #Desviacion típica muestral
  sd <- sd(col)
  #Error estandat
```

```
  SE <- sd / sqrt(n)

  #Buscamos t en la distribucion t-student
  t <- qt( alfa/2, df=n-1, lower.tail=FALSE )
  #Definimos el intervalo
  L <- mean(col) - t*SE
  U <- mean(col) + t*SE

  round(c(L,U),2)
}
```

Calculamos los intervalos de confianza de cada serie

```
#boxplot(wine_data$quality ~ wine_data$tipo, main="quality", col =myColors )
mean(calidad_blancos)
```

```
## [1] 5.854835
```

```
mean(calidad_tintos)
```

```
## [1] 5.623252
```

```
sd(calidad_blancos)
```

```
## [1] 0.8906827
```

```
sd(calidad_tintos)
```

```
## [1] 0.823578
```

```
# Aplicamos la funcion creada para calcular el intervalo de confianza
IC.blancos <- miIC.tstudent(calidad_blancos, 95)
IC.tintos <- miIC.tstudent(calidad_tintos, 95)
```

Podemos intuir que los intervalos son disjuntos por lo que la calidad de los vinos en función del tipo no es igual.

Todo esto lo podemos hacer directamente con funciones de R, aplicando un test de t-student, ya que asumimos la normalidad de la media muestrar por el teorema del limite central y hemos comprobado la homogeneidad de las varianzas.

"Sea:

μ_1 La media de la calidad de los vinos tintos

μ_2 La media de la calidad de los vinos blancos

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

```
#wilcox.test(quality ~ tipo, data = wine_data)
t.test(calidad_blancos,calidad_tintos)
```

```
##
##  Welch Two Sample t-test
##
## data:  calidad_blancos and calidad_tintos
## t = 8.7568, df = 2527, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1797244 0.2834401
## sample estimates:
## mean of x mean of y
##  5.854835  5.623252
```

Efectivamente, como habíamos intuido el t-test nos indica que rechamos la hipotesis nula de que la calidad para los dos tipos de vino.

4.3.2 Comparación de mas de dos grupos (ANOVA). Tomando la calidad del vino como una variable categórica, comprobaremos si el grado de alcohol es igual o distintos

```
res.aov <- aov(alcohol ~ calidad_vino, data = wine_data)
summary(res.aov)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## calidad_vino    6   1931   321.9   308.3 <2e-16 ***
## Residuals    5306   5540     1.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Podemos concluir que con ese p-valor, los valores del alcohol son significativamente distintos estadisticamente en función de la calidad del vino.

4.3.3 Correlación

En este punto vamos a visualizar cuál es la correlación entre todas las variables.

```
cols=c('acidity','citric_acid','residual_sugar','chlorides','free_sulfur_dioxide','total_sulfur_dioxide
data_corr <- cor(wine_data[cols])
data_corr
```

```
##                         acidity  citric_acid residual_sugar   chlorides
## acidity               1.0000000  0.273131182     -0.1262507  0.32664831
## citric_acid           0.2731312  1.000000000      0.1475058  0.04510371
## residual_sugar       -0.1262507  0.147505792      1.0000000 -0.12834034
## chlorides             0.3266483  0.045103706     -0.1283403  1.00000000
## free_sulfur_dioxide  -0.3189802  0.136182781      0.4198086 -0.19384169
## total_sulfur_dioxide -0.3664627  0.198551173      0.4967668 -0.27446589
## alcohol              -0.1068110 -0.007594672     -0.3128151 -0.27255873
## ph                   -0.2304407 -0.347090359     -0.2426242  0.03621609
## density               0.5089619  0.093056630      0.4996002  0.38273512
## sulphates             0.3246280  0.052848543     -0.1815712  0.39097139
##                      free_sulfur_dioxide total_sulfur_dioxide      alcohol
## acidity                      -0.31898023         -0.366462688 -0.106811010
## citric_acid                   0.13618278          0.198551173 -0.007594672
## residual_sugar                0.41980856          0.496766833 -0.312815080
## chlorides                    -0.19384169         -0.274465885 -0.272558733
## free_sulfur_dioxide           1.00000000          0.722065091 -0.173394476
## total_sulfur_dioxide          0.72206509          1.000000000 -0.249860909
## alcohol                      -0.17339448         -0.249860909  1.000000000
## ph                           -0.14879169         -0.225829687  0.096224025
## density                       0.01136162          0.004090262 -0.685608862
## sulphates                    -0.20766438         -0.284966512 -0.014213097
##                                   ph      density   sulphates
## acidity              -0.23044067  0.508961884  0.32462802
## citric_acid          -0.34709036  0.093056630  0.05284854
```

```
## residual_sugar       -0.24262423  0.499600194 -0.18157122
## chlorides             0.03621609  0.382735120  0.39097139
## free_sulfur_dioxide  -0.14879169  0.011361620 -0.20766438
## total_sulfur_dioxide -0.22582969  0.004090262 -0.28496651
## alcohol               0.09622403 -0.685608862 -0.01421310
## ph                    1.00000000  0.034496279  0.18513778
## density               0.03449628  1.000000000  0.28913586
## sulphates             0.18513778  0.289135861  1.00000000
```

Dado que nuestros atributos no tienen una distribución normal para comprobar el nivel de significancia con el que las variables están correlacionadas efectuaremos un test de Spearman

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
cols=c('acidity','citric_acid','residual_sugar','chlorides','free_sulfur_dioxide','total_sulfur_dioxide
# Calcular el coeficiente de correlación para cada variable cuantitativa con respecto al campo "quality

for (c in cols) {
  spearman_test = cor.test(wine_data[[c]], wine_data$quality, method = "spearman",  exact = FALSE)
  corr_coef = spearman_test$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)] <- c
}

print(corr_matrix)
```

```
##                         estimate       p-value
## acidity              -0.14158476  3.394923e-25
## citric_acid           0.11665826  1.458457e-17
## residual_sugar       -0.02843460  3.821525e-02
## chlorides            -0.30367186 9.505159e-114
## free_sulfur_dioxide   0.09071930  3.480320e-11
## total_sulfur_dioxide -0.05804573  2.300346e-05
## alcohol               0.48003473 1.926104e-304
## ph                    0.05252551  1.279639e-04
## density              -0.34935043 2.427215e-152
## sulphates             0.03697137  7.035853e-03
```

4.3.4 Regresión logistinca

Aplicaremos un modelo de regresion logistica multinominal sobre la variable categorica calidad_vino Vamos a dividir los datos en conjunto de test y de entrenamiento, aplicaremos el modelo y calcularemos la precisón del modelo

```
require(nnet)

#Construimos el conjunto de test y entrenamiento
train <- sample_frac(wine_data, 0.7)
sample_id <- as.numeric(rownames(train))
test <- wine_data[-sample_id,]
```

```
multinom.fit <- multinom(calidad_vino ~ acidity + citric_acid + residual_sugar +   chlorides + free_sul
```

```
## # weights:  84 (66 variable)
## initial  value 7236.839844
## iter  10 value 4889.603833
## iter  20 value 4641.799302
## iter  30 value 4487.752113
## iter  40 value 4052.710468
## iter  50 value 3984.248478
## iter  60 value 3975.565550
## iter  70 value 3970.273956
## iter  80 value 3967.451070
## iter  90 value 3962.254105
## iter 100 value 3956.419200
## final  value 3956.419200
## stopped after 100 iterations
```

```
# comprobamos el modelo
summary(multinom.fit)
```

```
## Call:
## multinom(formula = calidad_vino ~ acidity + citric_acid + residual_sugar +
##     chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     alcohol + ph + density + sulphates, data = train)
##
## Coefficients:
##   (Intercept)    acidity citric_acid residual_sugar  chlorides
## 4    12.60783 -0.7755497    1.400253     -0.1058444  -25.94118
## 5   -68.05495 -0.8694653    2.883114     -0.1125890  -20.28499
## 6   -24.35598 -1.0267807    4.523621     -0.0687374  -26.59522
## 7    73.37430 -0.8167854    6.069319     -0.0326853  -41.80426
## 8    14.08600 -0.8697812    7.085946      0.0170303  -64.46281
## 9   -12.04198  0.5040062    2.858110      0.0164863 -143.26629
##   free_sulfur_dioxide total_sulfur_dioxide    alcohol         ph    density
## 4        -0.009275505          -0.01116465 -0.2180235 -2.1766558   4.548462
## 5         0.057920121          -0.01721242 -0.4228719 -2.9439247  90.300581
## 6         0.071900995          -0.02324508  0.2983016 -2.8298511  38.969500
## 7         0.090051593          -0.02788213  0.8365324 -1.1838303 -74.226917
## 8         0.108250953          -0.03151988  1.3136396 -0.6979437 -22.908725
## 9         0.087333242          -0.01602998  0.8428614  5.8202301 -20.915846
##   sulphates
## 4  8.098995
## 5 10.135696
## 6 12.106968
## 7 13.689678
## 8 13.733989
## 9 10.683274
##
## Std. Errors:
##   (Intercept)   acidity citric_acid residual_sugar  chlorides
## 4   0.8058554 0.1630346  0.61991029     0.07304262 0.04891816
## 5   0.5923711 0.1497501  0.33782544     0.06885831 0.67374651
## 6   0.5306193 0.1500923  0.30657605     0.06891515 0.66628334
## 7   0.7067373 0.1541841  0.39711828     0.07016572 0.05485787
```

```
## 8   0.3629687 0.1730676  0.75587100     0.07490439 0.02240050
## 9   0.1441216 0.2748039  0.05632774     0.13655540 0.01318833
##   free_sulfur_dioxide total_sulfur_dioxide   alcohol        ph   density
## 4          0.02740543          0.006197916 0.1736228 0.5287830 0.7877065
## 5          0.02585937          0.005853555 0.1609799 0.3706737 0.5790267
## 6          0.02589400          0.005872471 0.1582025 0.3443840 0.5190966
## 7          0.02615837          0.006038841 0.1609023 0.3988425 0.6908272
## 8          0.02724189          0.006897342 0.1738150 0.4326538 0.3554459
## 9          0.04294629          0.014167057 0.3089487 0.8585527 0.1458852
##   sulphates
## 4 0.6967811
## 5 0.3420860
## 6 0.2927954
## 7 0.3521277
## 8 0.6136727
## 9 0.1653765
##
## Residual Deviance: 7912.838
## AIC: 8044.838
```

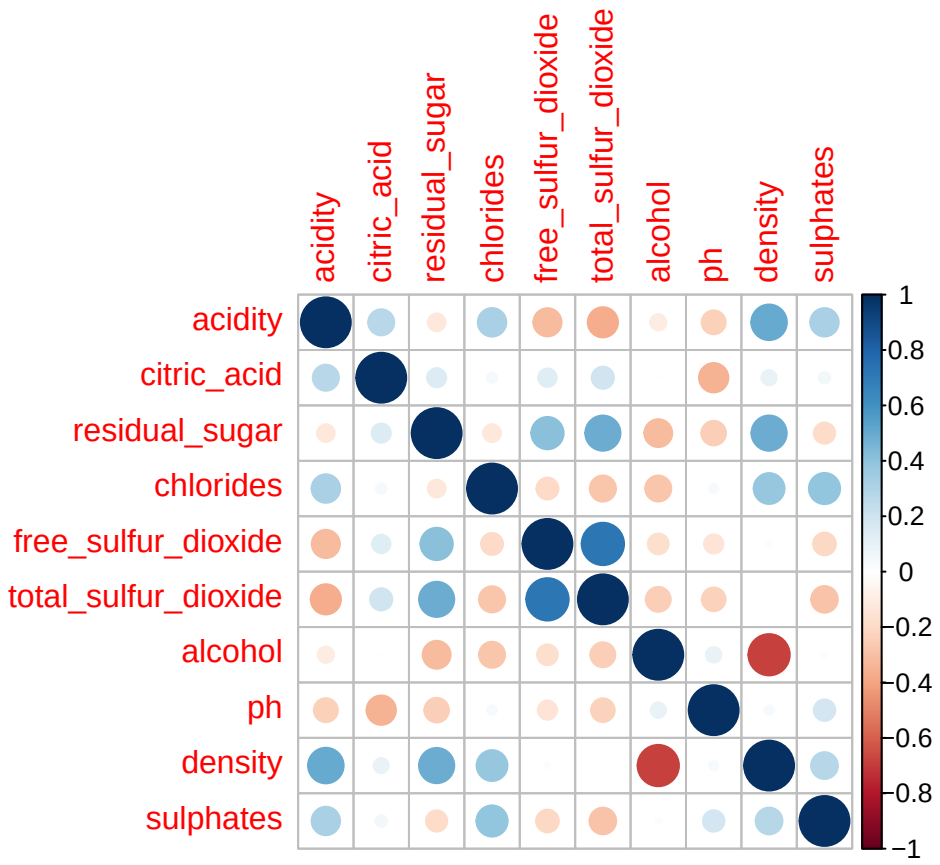
```
# Predicting the values for train dataset
train$precticed <- predict(multinom.fit, newdata = train, "class")

# Building classification table
ctable <- table(train$calidad_vino, train$precticed)

# Predicting the values for train dataset
test$precticedt <- predict(multinom.fit, newdata = test, "class")

# Building classification table
ctable1 <- table(test$calidad_vino, test$precticedt)
```

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

```
#Pintamos los dos intervalos
rbind(IC.blancos, IC.tintos)
```

```
##            [,1] [,2]
## IC.blancos 5.83 5.88
## IC.tintos  5.58 5.67
```

```
library(corrplot)
corrplot(data_corr)
```




```
# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(ctable))/sum(ctable))*100,2)
```

```
## [1] 54.48
```

```
# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(ctable1))/sum(ctable1))*100,2)
```

```
## [1] 56.84
```

```
ctable1
```

```
##
##       3   4   5   6   7   8   9
##   3   0   0  10   0   0   0   0
##   4   0   0  29  28   1   0   0
##   5   1   0 437 184   3   0   0
##   6   0   0 201 424  37   0   0
##   7   0   0  16 156  45   0   0
##   8   0   0   0  10  12   0   0
##   9   0   0   0   0   0   0   0
```

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

Después de ejecutar los análisis hemos concluido que la calidad del vino no es igual en el grupo de los vinos tintos que al de los vinos blancos, con un nivel de confianza del 95%. El análisis de las varianzas, muestra que el alcohol es estadisticamente distinto por la calidad del vino. No existe ninguna correlación significativa entre ninguna variable, salvo entre el grado del alcohol y la densidad. Aplicando un módelo logistico multinominal tomando la calidad del vino como una variable categórica, se obtiene resultados ligeramente superiores al 50%.