Práctica 2: Limpieza y validación de los datos

Tipología y ciclo de vida de los datos

Eva Mª Ambroa Rey

1. Descripción del dataset.

Se carga el fichero 'winequality-red'.

```
datos<-read.csv('winequality-red.csv',header=TRUE, sep=",")
attach(datos)
```

Al cargar los datos ya podemos observar que tenemos 1599 observacionesy 12 variables. Las diferentes variables que existen en este dataset hacen referencia a diferentes características o métricas del vino tinto. Además, según sus características el vino se clasifica según su calidad en una escala de va del 3 al 8, correspondiendo los valores más bajos a una mala calidad. Este dataset pretende dar repuesta a cuál es la calidad de un vino tinto según sus características. Para ello necesitamos evaluar cuáles son las variables que más influyen para determinar la calidad de un vino tinto. Lo primero que haremos será evaluar los tipos de datos que tenemos. Para ello hacemos una tabla con las diferentes variables y el tipo de dato.

```
library(knitr)
library(kableExtra)
clase<-sapply(datos,class)
kable(data.frame(Variables=names(clase), Tipo=as.vector(clase)))
```

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

Los datos se componen de diferentes variables referentes a diferentes medidas del vino a estudiar. Todos los valores son numéricos.

2. Integración y selección de los datos.

La calidad del vino se determina a partir de una suma de factores. Primero exploramos la variable 'quality'. Esta variable numérica tiene un rango de 3-8. Podemos clasificar la calidad de un vino como baja, intermedia y alta, para ello añadimos una columna que defina la calidad del vino según su índice:

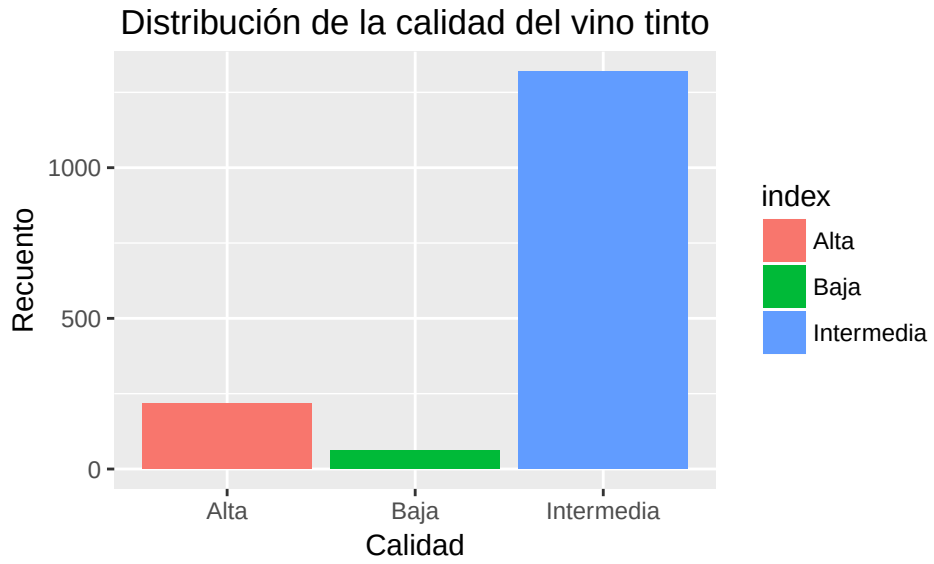
- calidad baja: 3-4
- calidad intermedia: 5-6
- calidad alta: 7-8

```r
library(plyr)
library(dplyr)
library(ggplot2)
library(pander)
library(kableExtra)
w = table(datos$quality)
t<-as.data.frame(w)
names(t)[1] = 'Calidad del vino tinto'
kable(t)
```

Calidad del vino tinto	Freq
3	10
4	53
5	681
6	638
7	199
8	18

Representamos esta nueva variable para ver su distribución.

```r
datos$index=''
datos$index[datos$quality<5] = 'Baja'
datos$index[5<=datos$quality & datos$quality <7] ='Intermedia'
datos$index[7<=datos$quality] ='Alta'
datos$index = as.factor(datos$index)
qplot(x=index,data = datos, fill = index,
      main='Distribución de la calidad del vino tinto', xlab='Calidad',
      ylab='Recuento')+theme(plot.title = element_text(hjust = 0.5))
```

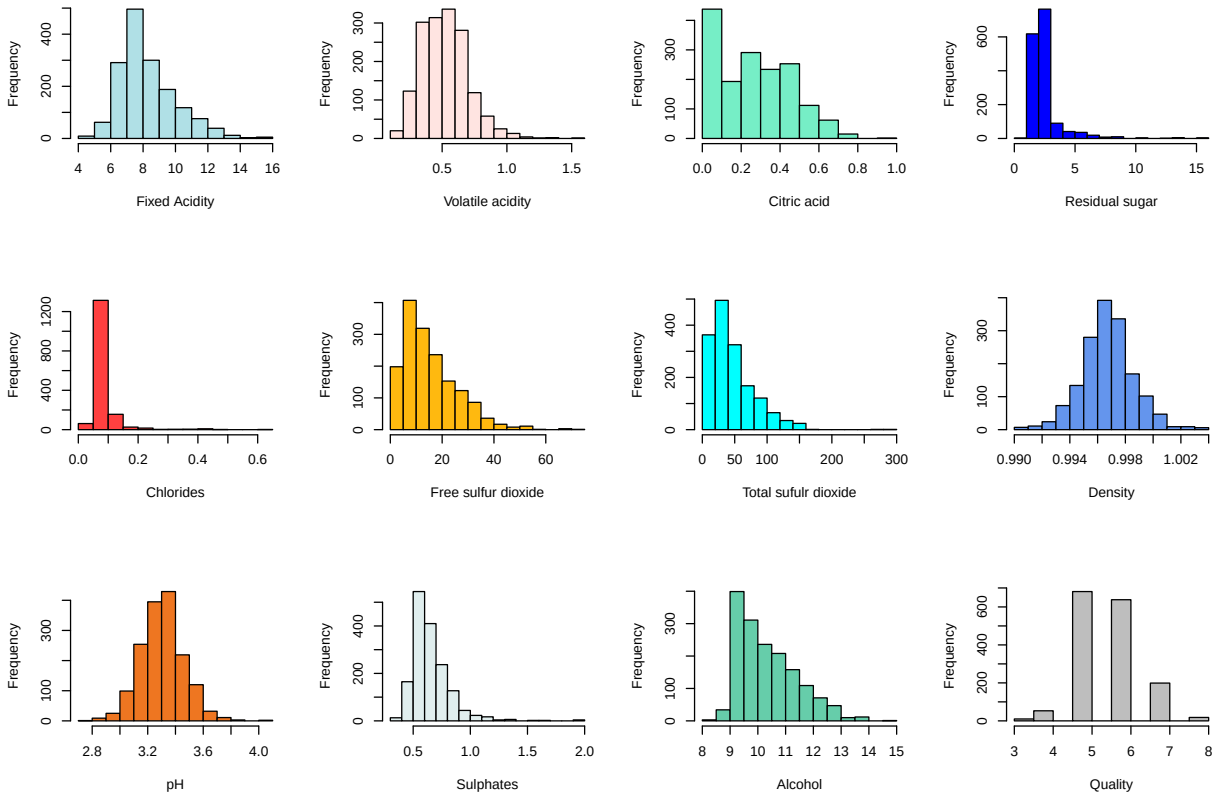


Como se puede ver en la figura anterior, la gran mayoría de los vinos tintos (82%) tiene un índice de calidad intermedio (5-6), un 4% tienen un índice bajo (3-4) y un 14% tienen un índice de calidad alto (7-8).

A continuación vamos a explorar el resto de variables y ver su distribución.

```r
par(mfrow=c(3,4))
hist(fixed.acidity, xlab="Fixed Acidity",col='powderblue', main=NULL)
hist(volatile.acidity,xlab="Volatile acidity", col='mistyrose',main=NULL)
```

```
hist(citric.acid,xlab="Citric acid", col='aquamarine2',main=NULL)
hist(residual.sugar,xlab="Residual sugar",col='blue',main=NULL)
hist(chlorides,xlab="Chlorides",col='brown1',main=NULL)
hist(free.sulfur.dioxide,xlab="Free sulfur dioxide",col='darkgoldenrod1',main=NULL)
hist(total.sulfur.dioxide,xlab="Total sufulr dioxide",col='cyan1',main=NULL)
hist(density,xlab="Density",col='cornflowerblue',main=NULL)
hist(pH,xlab="pH",col='chocolate2',main=NULL)
hist(sulphates,xlab="Sulphates",col='azure2',main=NULL)
hist(alcohol,xlab="Alcohol",col='aquamarine3',main=NULL)
hist(quality,xlab="Quality",col='grey',main=NULL)
```



Observando los histogramas podemos decir que tanto la variable 'density' como 'pH' siguen unas distribución normal. El resto de variables tienen una distribución asimétrica.

A continuación exploramos las principales métricas de los datos, donde podemos observar los valores mínimos y máximos, así como la media, la mediana y los rangos intercuartilícos.

```
library(pander)
library(Hmisc)
pander(head(summary(datos)), split.table = 80, style = 'rmarkdown')
```

Table 1: Table continues below

fixed.acidity	volatile.acidity	citric.acid	residual.sugar
Min. : 4.60	Min. :0.1200	Min. :0.000	Min. : 0.900
1st Qu.: 7.10	1st Qu.:0.3900	1st Qu.:0.090	1st Qu.: 1.900
Median : 7.90	Median :0.5200	Median :0.260	Median : 2.200

fixed.acidity	volatile.acidity	citric.acid	residual.sugar
Mean : 8.32	Mean :0.5278	Mean :0.271	Mean : 2.539
3rd Qu.: 9.20	3rd Qu.:0.6400	3rd Qu.:0.420	3rd Qu.: 2.600
Max. :15.90	Max. :1.5800	Max. :1.000	Max. :15.500

Table 2: Table continues below

chlorides	free.sulfur.dioxide	total.sulfur.dioxide
Min. :0.01200	Min. : 1.00	Min. : 6.00
1st Qu.:0.07000	1st Qu.: 7.00	1st Qu.: 22.00
Median :0.07900	Median :14.00	Median : 38.00
Mean :0.08747	Mean :15.87	Mean : 46.47
3rd Qu.:0.09000	3rd Qu.:21.00	3rd Qu.: 62.00
Max. :0.61100	Max. :72.00	Max. :289.00

Table 3: Table continues below

density	pH	sulphates	alcohol
Min. :0.9901	Min. :2.740	Min. :0.3300	Min. : 8.40
1st Qu.:0.9956	1st Qu.:3.210	1st Qu.:0.5500	1st Qu.: 9.50
Median :0.9968	Median :3.310	Median :0.6200	Median :10.20
Mean :0.9967	Mean :3.311	Mean :0.6581	Mean :10.42
3rd Qu.:0.9978	3rd Qu.:3.400	3rd Qu.:0.7300	3rd Qu.:11.10
Max. :1.0037	Max. :4.010	Max. :2.0000	Max. :14.90

quality	index
Min. :3.000	Alta : 217
1st Qu.:5.000	Baja : 63
Median :6.000	Intermedia:1319
Mean :5.636	NA
3rd Qu.:6.000	NA
Max. :8.000	NA

Aquí ya se observan algunos valores maximos bastante alejados del tercer quartil en algunas de las variables. Lo estudiaremos en profundidad en la siguiente sección.

3. Limpieza de los datos.

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

Vemos si existen campos nulos en el dataset.

```
a<-sapply(datos, function(x) sum(is.na(x)))
pander(head(a), split.table = 80, style = 'rmarkdown')
```

Table 5: Table continues below

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides
0	0	0	0	0

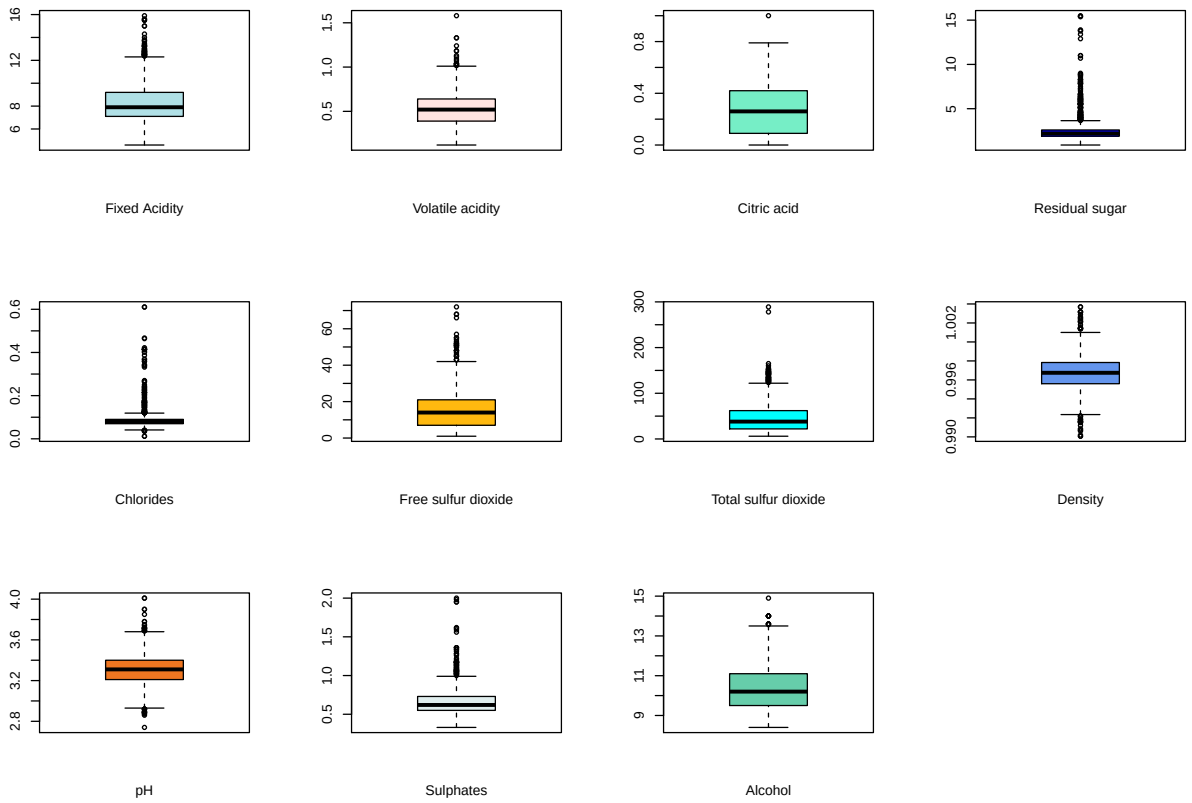
free.sulfur.dioxide
0

No existen campos nulos en ninguna columna. Si existiese algún dato nulo la manera de proceder podría ser o bien eliminar toda la fila del dataset o bien sustituir el valor nulo por la media o mediana de los valores de esa columna.

3.2. Identificación y tratamiento de valores extremos.

Para identificar los valores extremos primero representaremos los datos en un boxplot. Representaremos todas las variables excepto 'quality'.

```
par(mfrow=c(3,4))
attach(datos)
boxplot(fixed.acidity, xlab="Fixed Acidity",col='powderblue')
boxplot(volatile.acidity,xlab="Volatile acidity", col='mistyrose')
boxplot(citric.acid,xlab="Citric acid", col='aquamarine2')
boxplot(residual.sugar,xlab="Residual sugar",col='blue')
boxplot(chlorides,xlab="Chlorides",col='brown1')
boxplot(free.sulfur.dioxide,xlab="Free sulfur dioxide",col='darkgoldenrod1')
boxplot(total.sulfur.dioxide,xlab="Total sulfur dioxide",col='cyan1')
boxplot(density,xlab="Density",col='cornflowerblue')
boxplot(pH,xlab="pH",col='chocolate2')
boxplot(sulphates,xlab="Sulphates",col='azure2')
boxplot(alcohol,xlab="Alcohol",col='aquamarine3')
```



Visualmente ya se pueden identificar alguno de los valores extremos en estas variables. Por ejemplo en la variable 'Citric acid' hay claramente un valor que se aleja mucho del resto. Lo mismo ocurre para la variable 'Total sulfur dioxide' y 'Volatile acidity'. En general, todas las variables presentan valores extremos. Usaremos el percentill 99 como el umbral para marcar todos los valores extremos.

De la variable 'Fixed acidity' eliminamos tres outliers. Los datos sin los outliers los guardaremos en otro dataframe llamado 'datos_limpios', así seguiremos manteniendo el original.

```
quantile(datos$fixed.acidity, c(.999))
```

```
## 99.9%
##  15.6
```

```
datos_limpios<-filter(datos,fixed.acidity < 15.6)
```

Para la variable 'Volatile acidity':

```
quantile(datos$volatile.acidity, c(.999))
```

```
## 99.9%
##  1.33
```

```
datos_limpios<-filter(datos_limpios,volatile.acidity < 1.33)
```

La variable 'Citric acid' tiene un outlier.

```
quantile(datos$citric.acid, c(.999))
```

```
##   99.9%
## 0.78402
```

```
datos_limpios<-filter(datos_limpios,citric.acid < 0.78402)
```

La variable 'Residual sugar':

```
quantile(datos$residual.sugar, c(.999))
```

```
## 99.9%
##  15.4
```

```
datos_limpios<-filter(datos_limpios,residual.sugar < 15.4)
```

Para la variabe 'Chlorides'presenta un solo valor extremo.

```
quantile(datos$chlorides, c(.999))
```

```
##    99.9%
## 0.524486
```

```
datos_limpios<-filter(datos_limpios,chlorides < 0.524486)
```

La variable 'Free sulfur dioxide' también tiene 3 valores extremos.

```
quantile(datos$free.sulfur.dioxide, c(.999))
```

```
## 99.9%
##    68
```

```
datos_limpios<-filter(datos_limpios,free.sulfur.dioxide < 68)
```

De la variable Total sulfur dioxide, claramente se ven dos outliers. Eliminaremos estos valores.

```
quantile(datos$total.sulfur.dioxide, c(.999))
```

```
##   99.9%
## 210.426
```

```
datos_limpios<-filter(datos_limpios, total.sulfur.dioxide<210.426)
```

En el caso de la variable 'Density' tenemos valores extremos en las dos direcciones.

```
quantile(datos$density, c(.999))
```

```
##    99.9%
## 1.003397
```

```
datos_limpios<-filter(datos_limpios,density < 1.003397)
quantile(datos$density, c(.001))
```

```
##      0.1%
## 0.9901477
```

```
datos_limpios<-filter(datos_limpios,density > 0.9901477)
```

Para la variable 'pH':

```
quantile(datos$pH, c(.999))
```

```
##   99.9%
## 3.94422
```

```
datos_limpios<-filter(datos_limpios,pH < 3.94422)
quantile(datos$pH, c(.001))
```

```
##    0.1%
## 2.86598
```

```
datos_limpios<-filter(datos_limpios,pH > 2.86598)
```

Para la variable 'Sulphates':

```
quantile(datos$sulphates, c(.999))
```

```
##   99.9%
## 1.96206
```

```
datos_limpios<-filter(datos_limpios,sulphates < 1.96206)
```

Finalmente para la variable 'Alcohol':

```
quantile(datos$alcohol, c(.999))
```

```
## 99.9%
##    14
```

```
datos_limpios<-filter(datos_limpios,alcohol < 14)
```

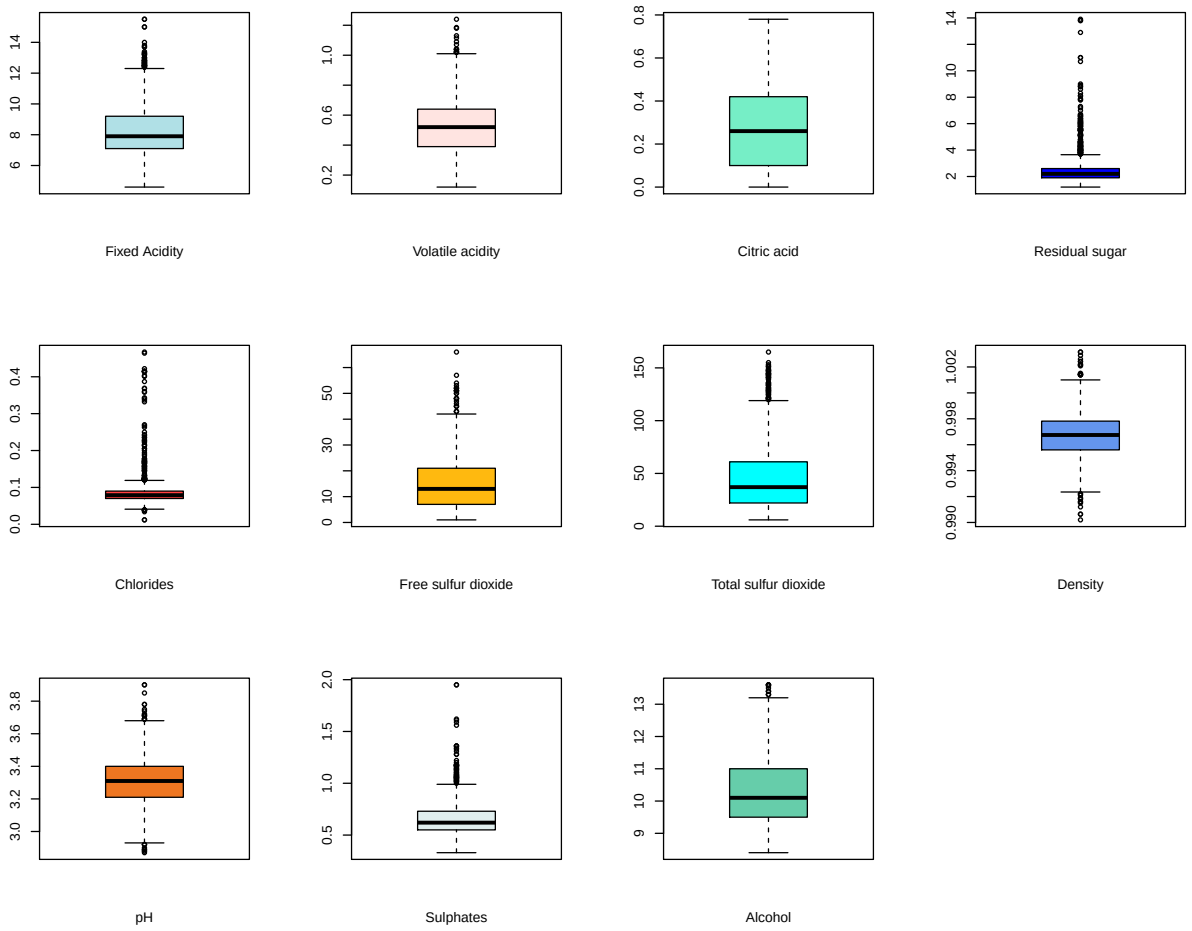
Generamos un nuevo dataset con los datos limpios.

```
write.csv(datos_limpios, file="winequality-red_clean.csv")
```

Finalmente, en el dataframe 'datos_limpios' tenemos 1569 observaciones, es decir, hemos eliminado 30 valores extremos.

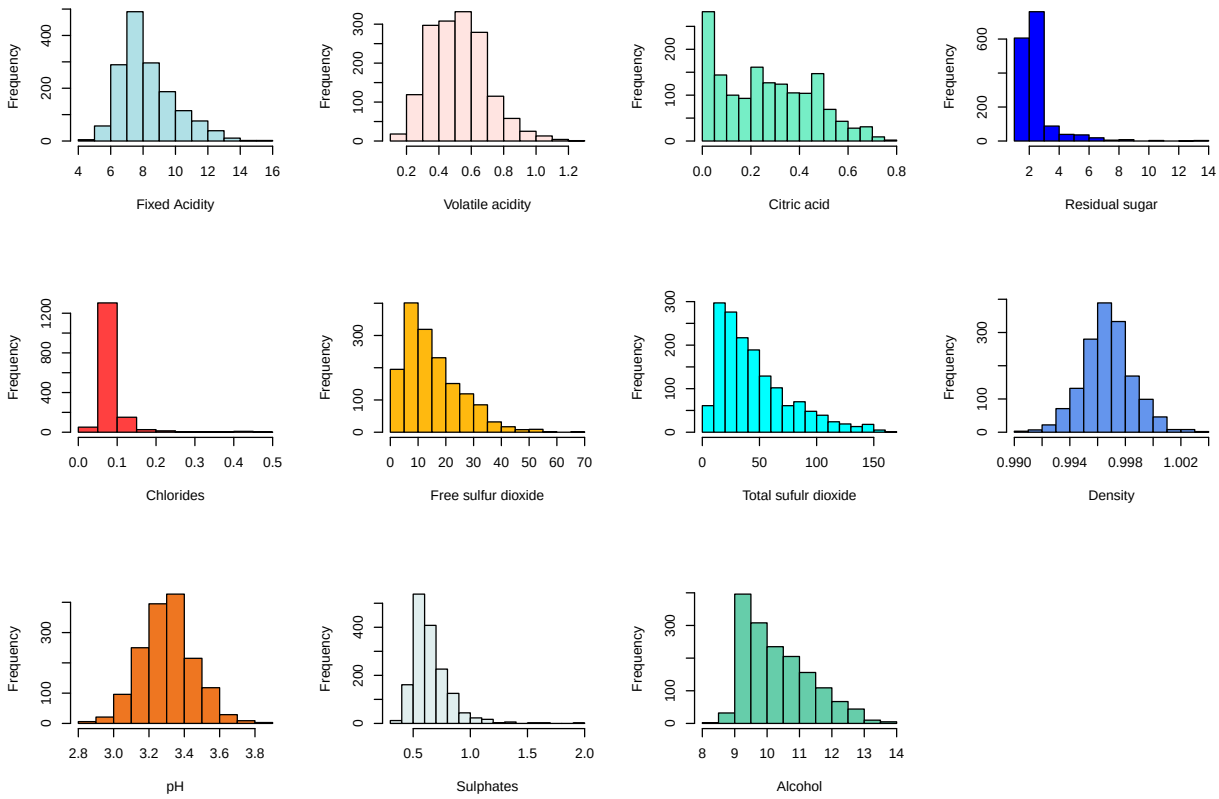
Volvemos a representar los datos y observamos si las distribuciones han cambiado.

```
par(mfrow=c(3,4))
attach(datos_limpios)
boxplot(fixed.acidity, xlab="Fixed Acidity",col='powderblue')
boxplot(volatile.acidity,xlab="Volatile acidity", col='mistyrose')
boxplot(citric.acid,xlab="Citric acid", col='aquamarine2')
boxplot(residual.sugar,xlab="Residual sugar",col='blue')
boxplot(chlorides,xlab="Chlorides",col='brown1')
boxplot(free.sulfur.dioxide,xlab="Free sulfur dioxide",col='darkgoldenrod1')
boxplot(total.sulfur.dioxide,xlab="Total sulfur dioxide",col='cyan1')
boxplot(density,xlab="Density",col='cornflowerblue')
boxplot(pH,xlab="pH",col='chocolate2')
boxplot(sulphates,xlab="Sulphates",col='azure2')
boxplot(alcohol,xlab="Alcohol",col='aquamarine3')
```



Ahora representamos las distribuciones de las variables.

```
par(mfrow=c(3,4))
hist(fixed.acidity, xlab="Fixed Acidity",col='powderblue', main=NULL)
hist(volatile.acidity,xlab="Volatile acidity", col='mistyrose',main=NULL)
hist(citric.acid,xlab="Citric acid", col='aquamarine2',main=NULL)
hist(residual.sugar,xlab="Residual sugar",col='blue',main=NULL)
hist(chlorides,xlab="Chlorides",col='brown1',main=NULL)
hist(free.sulfur.dioxide,xlab="Free sulfur dioxide",col='darkgoldenrod1',main=NULL)
hist(total.sulfur.dioxide,xlab="Total sufulr dioxide",col='cyan1',main=NULL)
hist(density,xlab="Density",col='cornflowerblue',main=NULL)
hist(pH,xlab="pH",col='chocolate2',main=NULL)
hist(sulphates,xlab="Sulphates",col='azure2',main=NULL)
hist(alcohol,xlab="Alcohol",col='aquamarine3',main=NULL)
```



Las distribuciones de algunas variables han cambiado y se asemejan más a una distribución normal. El cambio es notable para la variable 'Density'. Se puede observar algunas variables siguen teniendo algún outliers, pero como estos corresponden a las calidades que menos valores tienen, no los vamos a eliminar.

Examinamos de donde hemos eliminado los valores extremos.

```
w = table(datos_limpios$quality)
t<-as.data.frame(w)
names(t)[1] = 'Calidad del vino tinto'
kable(t)
```

Calidad del vino tinto	Freq
3	9
4	52
5	671
6	626
7	195
8	16

Como se ve en la tabla anterior, la calidad del vino 3 y 4 solo han perdido un dato, mientras que la calidad 5 ha perdido 10 valores, la 6 ha perdido 12, la 7 ha perdido 4 y la calidad 8 solamente dos valores.

4. Análisis de los datos.

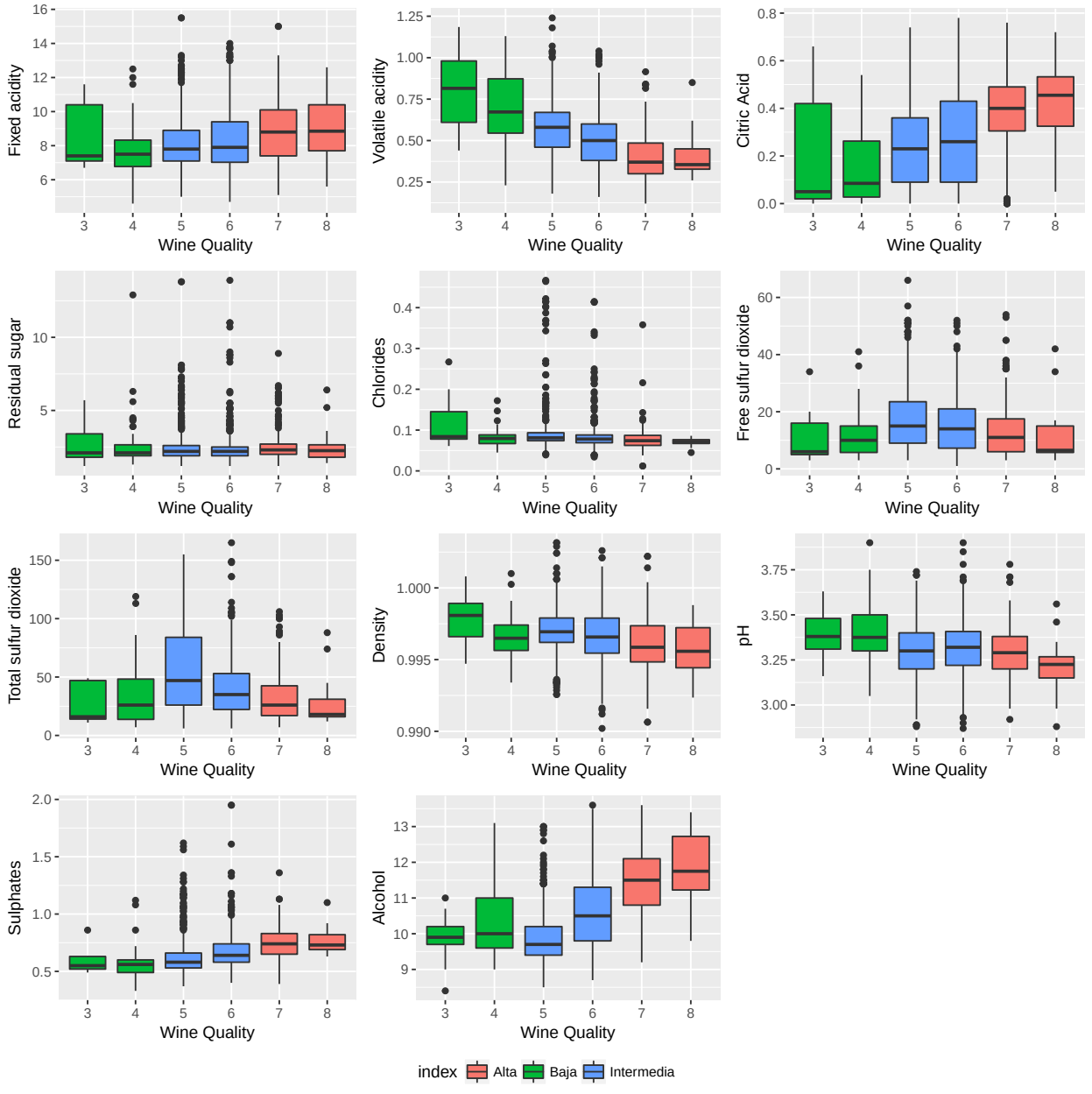
Ahora lo que trataremos de averiguar es cuales son las variables que más influyen en la determinación de la calidad del vino.

4.1. Selección de datos.

Graficamos todas las variales frente a la variable 'quality' para ver si visualmente detectamos algún patrón.

```
library(gridExtra)
library(grid)
library(lattice)
library(ggpubr)
p1<-ggplot(datos_limpios, aes(x=as.factor(quality), y=fixed.acidity, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Fixed acidity")

p2<-ggplot(datos_limpios, aes(x=as.factor(quality), y=volatile.acidity, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Volatile acidity")

p3<-ggplot(datos_limpios, aes(x=as.factor(quality), y=citric.acid, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Citric Acid")

p4<-ggplot(datos_limpios, aes(x=as.factor(quality), y=residual.sugar, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Residual sugar")

p5<-ggplot(datos_limpios, aes(x=as.factor(quality), y=chlorides, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Chlorides")

p6<-ggplot(datos_limpios, aes(x=as.factor(quality), y=free.sulfur.dioxide, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Free sulfur dioxide")

p7<-ggplot(datos_limpios, aes(x=as.factor(quality), y=total.sulfur.dioxide, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Total sulfur dioxide")

p8<-ggplot(datos_limpios, aes(x=as.factor(quality), y=density, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Density")

p9<-ggplot(datos_limpios, aes(x=as.factor(quality), y=pH, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("pH")

p10<-ggplot(datos_limpios, aes(x=as.factor(quality), y=sulphates, fill=index)) +
```

```
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Sulphates")

p11<-ggplot(datos_limpios, aes(x=as.factor(quality), y=alcohol, fill=index)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Alcohol")

ggarrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11, ncol=3,
          nrow=4, common.legend = TRUE, legend="bottom")
```



A simple vista podemos decir que las variables 'Fixed acidity','Alcohol', 'Volatile acidity', 'Citric Acid', 'Sulphates', 'pH' y 'Density' podrían determinar la calidad del vino. La calidad alta del vino parece estar

determinada por los siguientes factores:

- Aumenta con el valor de Fixed acidity.
- Aumenta si disminuye el valor Volatil acidity.
- Aumenta con el Citric Acid.
- Aumenta si disminuy la densidad.
- Aumenta si el valor de pH es bajo.
- Aumenta con el valor de los sulfatos.
- Aumenta con el valor del alcohol.

Es necesario llevar a cabo un análisis estadístico para comprobar si estas correlaciones son ciertas.

4.2. Normalidad y homogeneidad de la varianza.

Para determinar la existencia de normalidad podemos optar por realizar la prueba de Shapiro-Wilk. También graficaremos las variables en un QQ plot.

La hipótesis nula H0: la distribución de cada variable es normal.
La hipótesis alternativa H1: la distribución de las variables no es normal.

Si $p > 0.05$ aceptamos la hipótesis nula, existe normalidad.

Si $p < 0.05$ rechazamos la hipótesis nula, no existe normalidad.

```
library("dplyr")
shapiro.test(datos_limpios$fixed.acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$fixed.acidity
## W = 0.94458, p-value < 2.2e-16
```

```
shapiro.test(datos_limpios$volatile.acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$volatile.acidity
## W = 0.98258, p-value = 7.369e-13
```

```
shapiro.test(datos_limpios$citric.acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$citric.acid
## W = 0.95512, p-value < 2.2e-16
```

```
shapiro.test(datos_limpios$residual.sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$residual.sugar
## W = 0.6096, p-value < 2.2e-16
```

```r
shapiro.test(datos_limpios$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$chlorides
## W = 0.50835, p-value < 2.2e-16
```

```r
shapiro.test(datos_limpios$free.sulfur.dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$free.sulfur.dioxide
## W = 0.91213, p-value < 2.2e-16
```

```r
shapiro.test(datos_limpios$total.sulfur.dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$total.sulfur.dioxide
## W = 0.88929, p-value < 2.2e-16
```

```r
shapiro.test(datos_limpios$density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$density
## W = 0.99492, p-value = 3.626e-05
```

```r
shapiro.test(datos_limpios$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$pH
## W = 0.99628, p-value = 0.0007165
```

```r
shapiro.test(datos_limpios$sulphates)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$sulphates
## W = 0.85781, p-value < 2.2e-16
```

```r
shapiro.test(datos_limpios$alcohol)
```

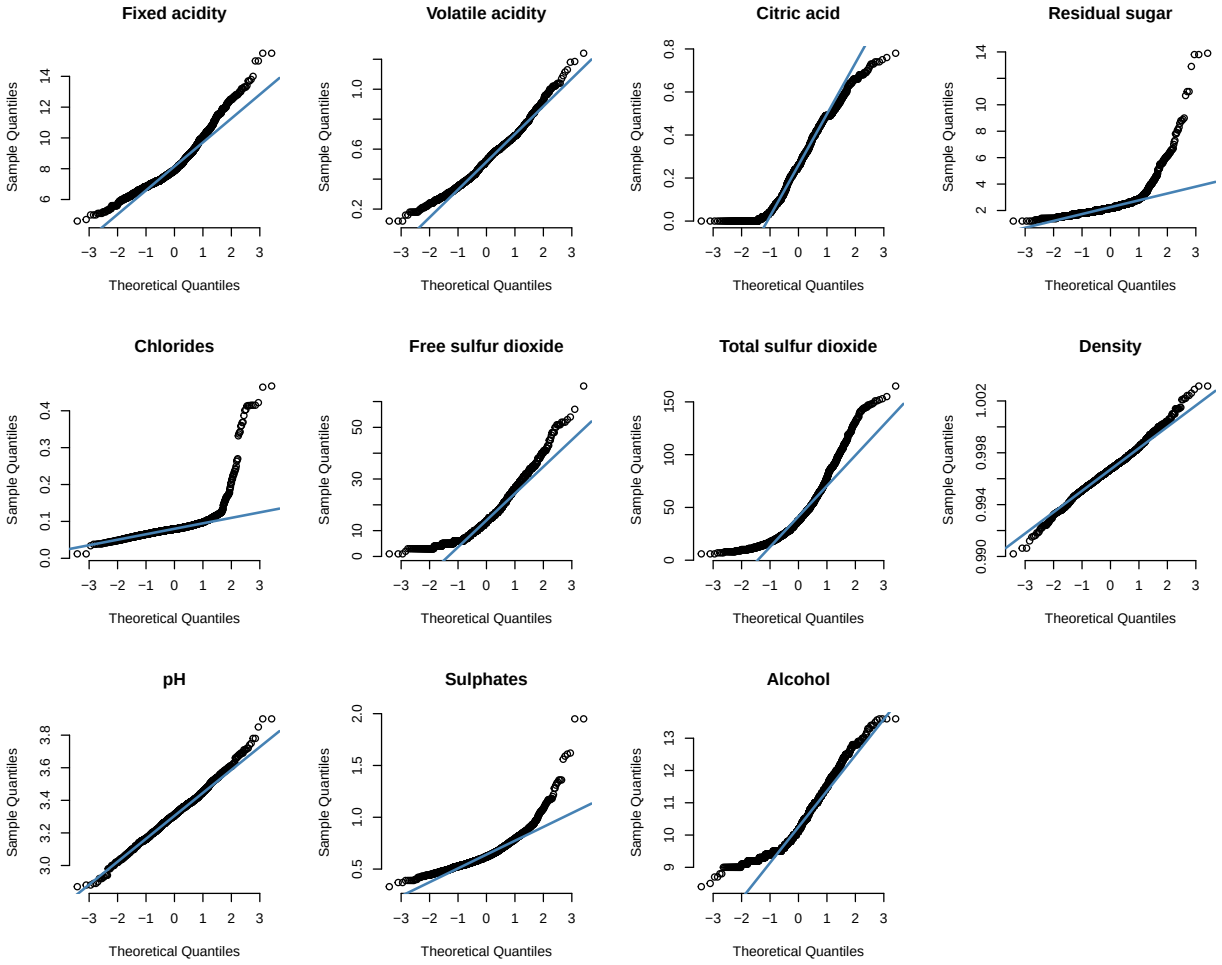
```
##
##  Shapiro-Wilk normality test
##
## data:  datos_limpios$alcohol
## W = 0.9314, p-value < 2.2e-16
```

En general podemos decir que los datos no siguen una distribución normal. Cuando se tiene la muestra de varios grupos es apropiado comprobar la normalidad por grupo, pero no lo haremos porque existe una gran diferencia de muestras en los diferentes grupos según la variable 'quality'

Para poblaciones con n>30 esto no representa un problema, pero en nuestro caso tenemos algunos grupos con muy pocos valores, por lo que los test que tenemos que aplicar deberán reflejar la no normalidad de estos grupos.

Representamos ahora el gráfico QQ:

```
library(ggpubr)
par(mfrow=c(3,4))
a<-qqnorm(datos_limpios$fixed.acidity, pch = 1, frame = FALSE, main='Fixed acidity')
a<-qqline(datos_limpios$fixed.acidity, col = "steelblue", lwd = 2)

b<-qqnorm(datos_limpios$volatile.acidity, pch = 1, frame = FALSE,main='Volatile acidity')
b<-qqline(datos_limpios$volatile.acidity, col = "steelblue", lwd = 2)

c<-qqnorm(datos_limpios$citric.acid, pch = 1, frame = FALSE,main='Citric acid')
c<-qqline(datos_limpios$citric.acid, col = "steelblue", lwd = 2)

d<-qqnorm(datos_limpios$residual.sugar, pch = 1, frame = FALSE,main='Residual sugar')
d<-qqline(datos_limpios$residual.sugar, col = "steelblue", lwd = 2)

e<-qqnorm(datos_limpios$chlorides, pch = 1, frame = FALSE,main='Chlorides')
e<-qqline(datos_limpios$chlorides, col = "steelblue", lwd = 2)

f<-qqnorm(datos_limpios$free.sulfur.dioxide, pch = 1, frame = FALSE,main='Free sulfur dioxide')
f<-qqline(datos_limpios$free.sulfur.dioxide, col = "steelblue", lwd = 2)

g<-qqnorm(datos_limpios$total.sulfur.dioxide, pch = 1, frame = FALSE,main='Total sulfur dioxide')
g<-qqline(datos_limpios$total.sulfur.dioxide, col = "steelblue", lwd = 2)

h<-qqnorm(datos_limpios$density, pch = 1, frame = FALSE,main='Density')
h<-qqline(datos_limpios$density, col = "steelblue", lwd = 2)

i<-qqnorm(datos_limpios$pH, pch = 1, frame = FALSE,main='pH')
i<-qqline(datos_limpios$pH, col = "steelblue", lwd = 2)

j<-qqnorm(datos_limpios$sulphates, pch = 1, frame = FALSE,main='Sulphates')
j<-qqline(datos_limpios$sulphates, col = "steelblue", lwd = 2)

k<-qqnorm(datos_limpios$alcohol, pch = 1, frame = FALSE,main='Alcohol')
k<-qqline(datos_limpios$alcohol, col = "steelblue", lwd = 2)
```



Observando los gráficos QQ, se puede decir que las variables 'Density' y 'pH' son las que más se acercan a una distribución normal.

El supuesto de homogeneidad de varianzas (homocedasticidad), considera que la varianza es constante entre diferentes grupos. Haremos un contraste de hipótesis para comprobar la homodedasticidad:

Hipótesis nula: igualdad de varianzas entre los diferentes grupos (H0=H1).
Hipótesis alternativa: no existe igual de varianzas entre los diferentes grupos.

Si se tiene seguridad de que las muestras a comparar proceden de poblaciones que siguen una distribución normal, son recomendables el F-test y el test de Bartlet, pareciendo ser el segundo más recomendable ya que el primero es muy potente pero extremadamente sensible a desviaciones de la normal. Si no se tiene la seguridad de que las poblaciones de origen son normales, se recomiendan el test de Levene utilizando la mediana.

```
library(car)
leveneTest(y = datos_limpios$fixed.acidity, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value    Pr(>F)
## group    5  5.9538 1.835e-05 ***
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Fixed acidity' no cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$volatile.acidity, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value    Pr(>F)
## group    5  4.9236 0.0001779 ***
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Volatile acidity' no cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$citric.acid, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value   Pr(>F)
## group    5  3.3881 0.004762 **
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Citric acid' no cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$residual.sugar, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value Pr(>F)
## group    5  1.8173 0.1063
##       1563
```

La variable 'Residual sugar' cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$chlorides, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value Pr(>F)
## group    5  1.6343 0.1477
##       1563
```

La variable 'Chlorides' cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$free.sulfur.dioxide, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value  Pr(>F)
## group    5  2.1371 0.05857 .
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Free sulfur dioxide' cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$total.sulfur.dioxide, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value    Pr(>F)
## group    5   28.55 < 2.2e-16 ***
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Total sulfur dioxide' no cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$density, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value    Pr(>F)
## group    5  8.4815 6.221e-08 ***
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La variable 'Density' no cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$pH, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value Pr(>F)
## group    5  0.3111 0.9065
##       1563
```

La variable 'pH' cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$sulphates, group = datos_limpios$quality, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value Pr(>F)
## group    5  0.5957 0.7033
##       1563
```

La variable 'Sulphates' cumple el supuesto de homocedasticidad.

```
leveneTest(y = datos_limpios$alcohol, group = datos_limpios$quality, center = "median")
```

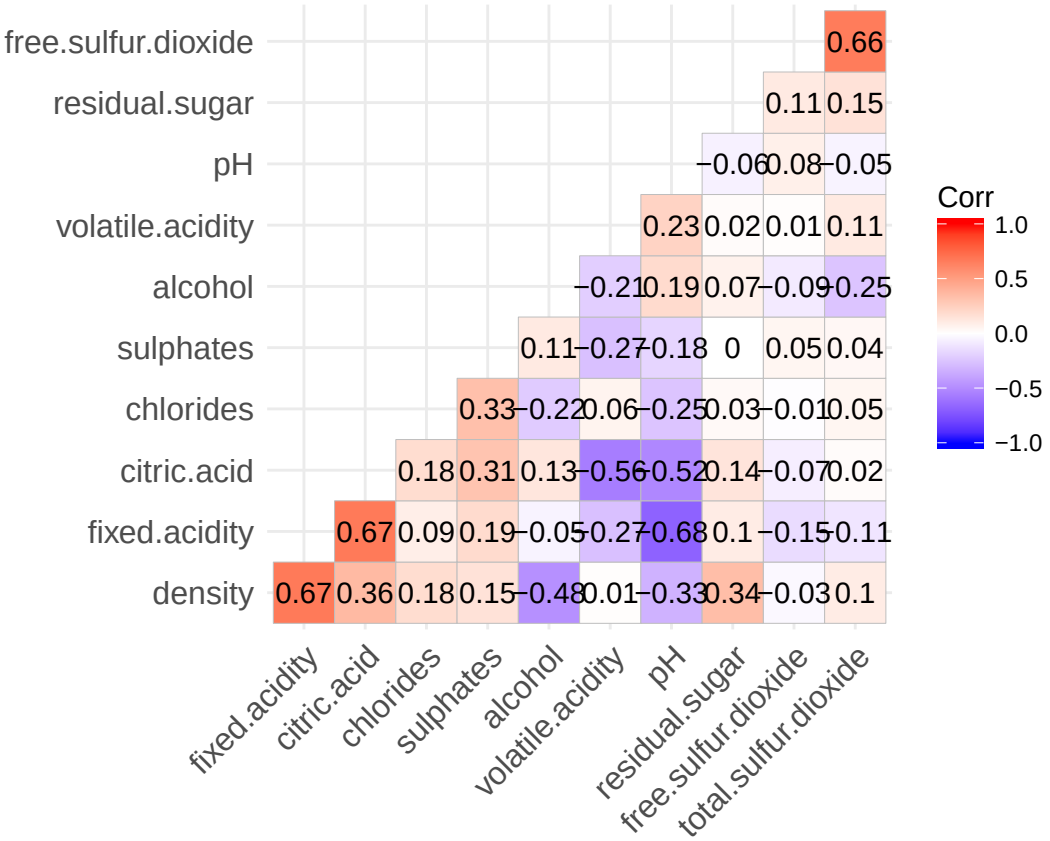
```
## Levene's Test for Homogeneity of Variance (center = "median")
##         Df F value    Pr(>F)
## group    5  24.613 < 2.2e-16 ***
##       1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En la variable 'Alcohol' la varianza no es constante entre los diferentes grupos.

4.3. Pruebas estadísticas.

Queremos evaluar que variables están relacionadas entre sí y cuáles determinan la calidad del vino. Para empezar veremos que correlaciones existen entre las diferentes variables.

```
library(ggcorrplot)
datos_corr<-select(datos_limpios, -index,-quality)
ggcorrplot(cor(datos_corr), hc.order = TRUE, type = "lower", lab = TRUE, insig = "blank")
```



De la matriz anterior podemos observar que existe un correlación positiva entre las siguientesvariables:

- 'Free sulfur dioxide' y 'Total sulfur dioxide'
- 'Density' y 'Fixed acidity'
- 'Fixed acidity y 'Citric acid'

La correlaciones negativas más destacadas son entre:

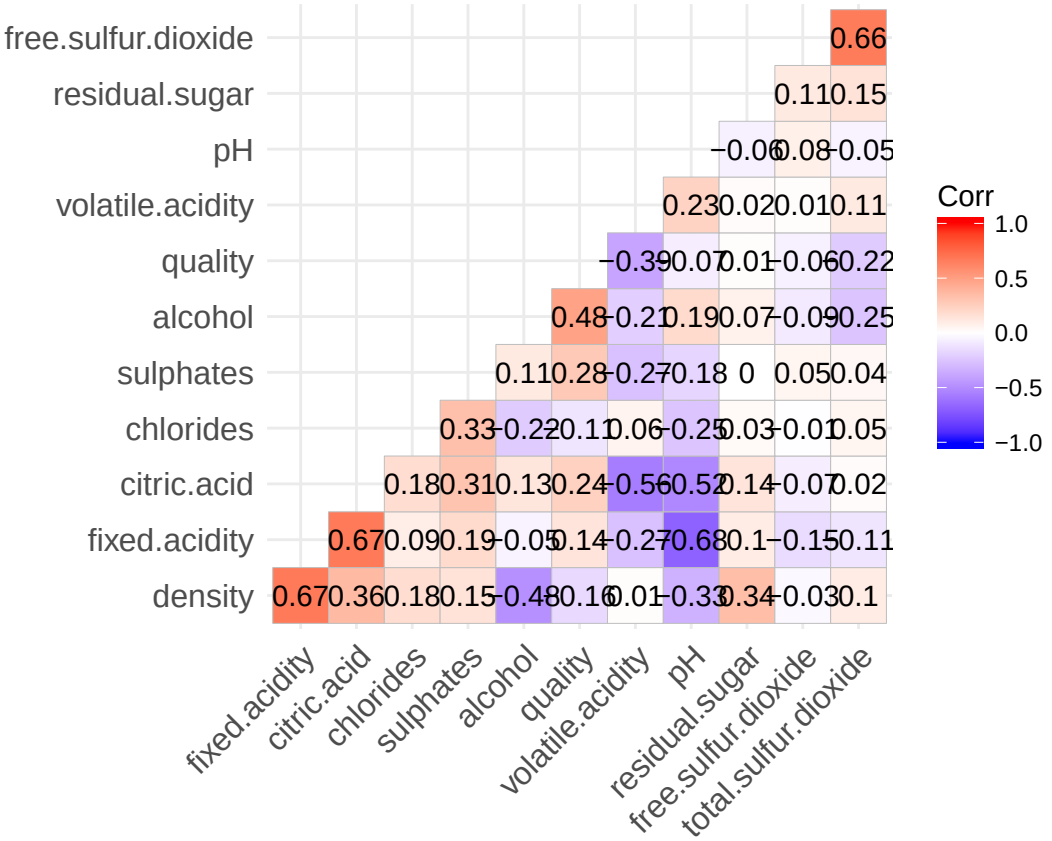
- 'pH' y 'Fixed acidity'
- 'Citric acid' y 'Volatile acidity'

Exploramos ahora la correlación existente entre la calidad del vino y el resto de variables.

```
cor(x=datos_limpios[,1:12], y=(datos_limpios$quality))
```

```
##                              [,1]
## fixed.acidity         0.14225114
## volatile.acidity     -0.38713762
## citric.acid           0.23706915
## residual.sugar        0.01306846
## chlorides            -0.11226700
## free.sulfur.dioxide  -0.06242428
## total.sulfur.dioxide -0.21847834
## density              -0.16342643
## pH                   -0.07262987
## sulphates             0.27974636
## alcohol               0.48183987
## quality               1.00000000
```

```r
ggcorrplot(cor(datos_limpios[,1:12]), hc.order = TRUE, type = "lower", lab = TRUE, insig = "blank")
```



La variables que presentan una correlación más alta son 'Alcohol', 'Volatile acidity', 'Sulphates' y 'Citric acid'. La variable 'Alcohol' es la que tiene un De los resultados, vemos que la variable 'Alcohol', destaca en términos de relación lineal positiva. Las variables 'Citric acid' y 'Sulphates' también muestran una relación positiva. La variable 'Volatile acidity' destaca como una variable correlacionada negativamente.

Veamos ahora si estas variables se ajustan a un modleo lineal.

- Alcohol:

```r
summary(lm(formula = quality ~ alcohol, data = datos_limpios))
```

```
##
## Call:
## lm(formula = quality ~ alcohol, data = datos_limpios)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8594 -0.4084 -0.1601  0.5165  2.5916
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.72506    0.18043   9.561   <2e-16 ***
## alcohol      0.37585    0.01727  21.767   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.7029 on 1567 degrees of freedom
## Multiple R-squared:  0.2322, Adjusted R-squared:  0.2317
## F-statistic: 473.8 on 1 and 1567 DF,  p-value: < 2.2e-16
```

- Volatile acidity:

```
summary(lm(formula = quality ~ volatile.acidity, data = datos_limpios))
```

```
##
## Call:
## lm(formula = quality ~ volatile.acidity, data = datos_limpios)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78856 -0.54003 -0.00745  0.46885  2.93929
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.56967    0.05934  110.72   <2e-16 ***
## volatile.acidity -1.77525    0.10681  -16.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7396 on 1567 degrees of freedom
## Multiple R-squared:  0.1499, Adjusted R-squared:  0.1493
## F-statistic: 276.3 on 1 and 1567 DF,  p-value: < 2.2e-16
```

- Sulphates:

```
summary(lm(formula = quality ~ sulphates, data = datos_limpios))
```

```
##
## Call:
## lm(formula = quality ~ sulphates, data = datos_limpios)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.91454 -0.52837  0.07167  0.45784  2.40267
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.72843    0.08084   58.49   <2e-16 ***
## sulphates    1.37919    0.11957   11.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7701 on 1567 degrees of freedom
## Multiple R-squared:  0.07826,    Adjusted R-squared:  0.07767
## F-statistic:   133 on 1 and 1567 DF,  p-value: < 2.2e-16
```

- Citric acid:

```
summary(lm(formula = quality ~ citric.acid, data = datos_limpios))
```

```
##
## Call:
## lm(formula = quality ~ citric.acid, data = datos_limpios)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02011 -0.59392  0.08892  0.51510  2.58448
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.36596    0.03397  157.94   <2e-16 ***
## citric.acid  0.99113    0.10260    9.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7793 on 1567 degrees of freedom
## Multiple R-squared:  0.0562, Adjusted R-squared:  0.0556
## F-statistic: 93.31 on 1 and 1567 DF,  p-value: < 2.2e-16
```

De los resultados anteriores, basandonos en el valor R-squared, la variable 'Alcohol' explica la calidad del vino en un 23.2%. La siguiente variable que más peso aporta a la calidad del vino es 'Volatile acidity' con un 14.9%

En resumen podemos decir que la variable que más peso aporta para la calidad del vino es el Alcohol, seguida de 'Volatile acidity' y en menos medida 'Sulphates' y 'Citric acid'.

Creamos ahora un modelo añadiendo cada variable, para ver si podemos obtener uno que se ajuste bien a nuestros datos.

```
m1 <- lm(as.numeric(quality) ~ alcohol, data = datos_limpios)
m2 <- update(m1, ~ . + volatile.acidity)
m3 <- update(m2, ~ . + sulphates)
m4 <- update(m3, ~ . + fixed.acidity)
m5 <- update(m4, ~ . + citric.acid)
summary(m5)
```

```
##
## Call:
## lm(formula = as.numeric(quality) ~ alcohol + volatile.acidity +
##     sulphates + fixed.acidity + citric.acid, data = datos_limpios)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80821 -0.37393 -0.06062  0.46397  2.09782
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.91836    0.23051   8.322  < 2e-16 ***
## alcohol            0.33276    0.01646  20.221  < 2e-16 ***
## volatile.acidity -1.29743    0.11646 -11.141  < 2e-16 ***
## sulphates          0.83044    0.10631   7.812 1.03e-14 ***
## fixed.acidity      0.06335    0.01335   4.745 2.28e-06 ***
## citric.acid       -0.49194    0.13802  -3.564 0.000376 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6459 on 1563 degrees of freedom
## Multiple R-squared:  0.3534, Adjusted R-squared:  0.3513
## F-statistic: 170.8 on 5 and 1563 DF,  p-value: < 2.2e-16
```

Este último modelo creado tiene un R2 de 35%.

Podemos crear un árbol de decisión para intentar obtener un modelo mejor que el anterior.

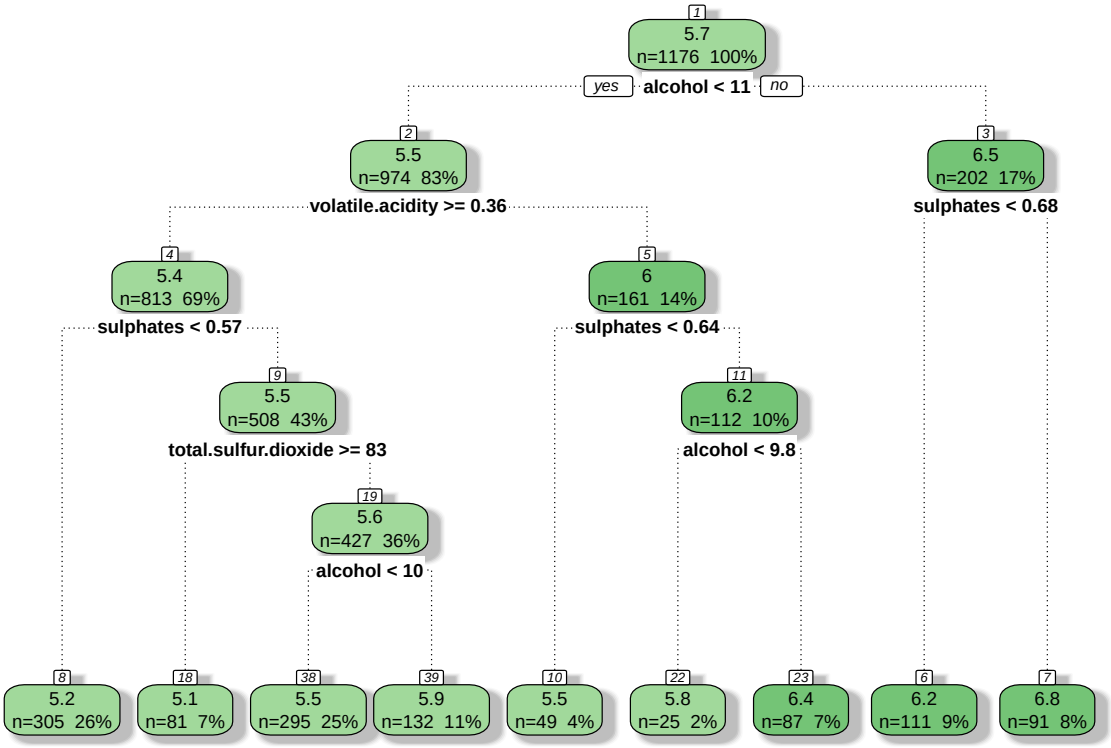
```
#training set
datos_tree<-select(datos_limpios, -index)
datos_train <- datos_tree[1:1176, ]

#test set
datos_test <- datos_tree[1177:1569, ]
```

```
library(rpart)
library(plotly)
library(rpart.plot)
library(rattle)
arbol <- rpart(quality ~. , data = datos_train)
arbol
```

```
## n= 1176
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1176 764.65310 5.663265
##    2) alcohol< 11.45 974 505.49590 5.497947
##      4) volatile.acidity>=0.3625 813 357.83270 5.392374
##        8) sulphates< 0.575 305 105.12790 5.160656 *
##        9) sulphates>=0.575 508 226.49610 5.531496
##         18) total.sulfur.dioxide>=83 81  12.39506 5.086420 *
##         19) total.sulfur.dioxide< 83 427 195.01170 5.615925
##           38) alcohol< 10.45 295 121.74240 5.494915 *
##           39) alcohol>=10.45 132  59.29545 5.886364 *
##      5) volatile.acidity< 0.3625 161  92.84472 6.031056
##       10) sulphates< 0.635 49  20.20408 5.530612 *
##       11) sulphates>=0.635 112  55.00000 6.250000
##         22) alcohol< 9.75 25   8.56000 5.760000 *
##         23) alcohol>=9.75 87  38.71264 6.390805 *
##    3) alcohol>=11.45 202 104.18320 6.460396
##      6) sulphates< 0.685 111  50.39640 6.180180 *
##      7) sulphates>=0.685 91  34.43956 6.802198 *
```

Visualizamos el árbol de decisión.

```
fancyRpartPlot(arbol)
```



Rattle 2019−Jan−06 12:48:02 evita

Si nos fijamos en el árbol, el principal método de clasificación es por la variable 'Alcohol', seguida de 'Volatile acidity' y 'Sulphates'. Esto es coherente con lo que hemos obtenido anteriormente.

Caculamos la predicción:

```
prediccion <- predict(arbol,datos_test)

summary(prediccion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.086   5.161   5.495   5.662   5.886   6.802
```

```
summary(datos_test$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.545   6.000   8.000
```

Por lo que podemos observar la predicción es bastante mala. Calculamos el error medio absoluto.

```
error <- function(actual, predicted){
  mean(abs(actual - predicted))
}

error(datos_test$quality, prediccion)
```

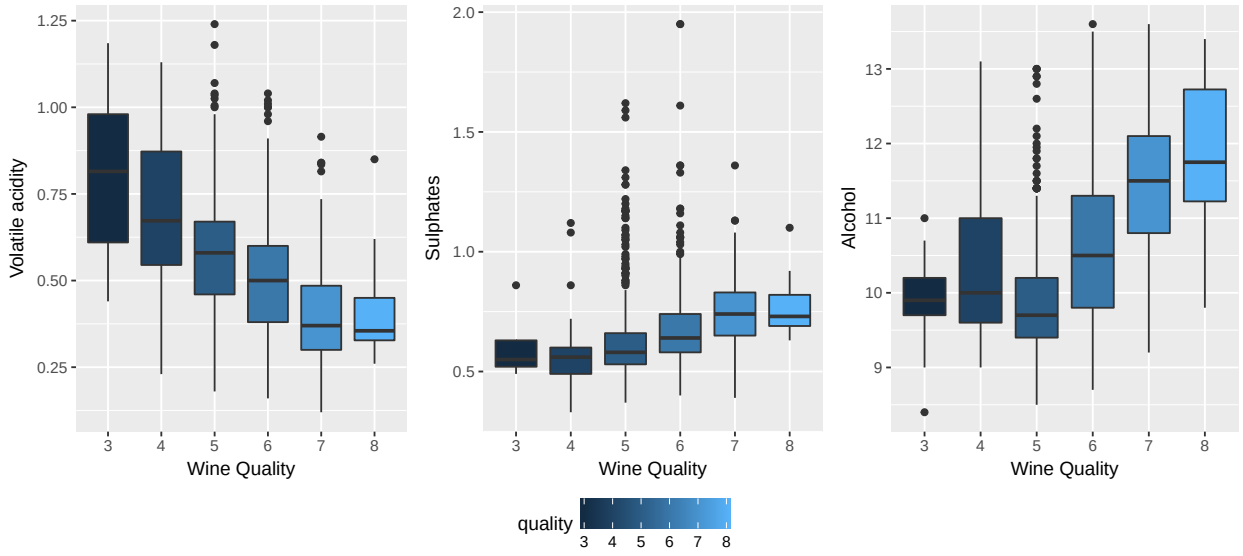
```
## [1] 0.5225932
```

Tenemos un error del 52%. Al igual qu el modelo anterior, no hemos encontrado un método que se ajuste del todo bien a los datos.

5. Representación de los datos.

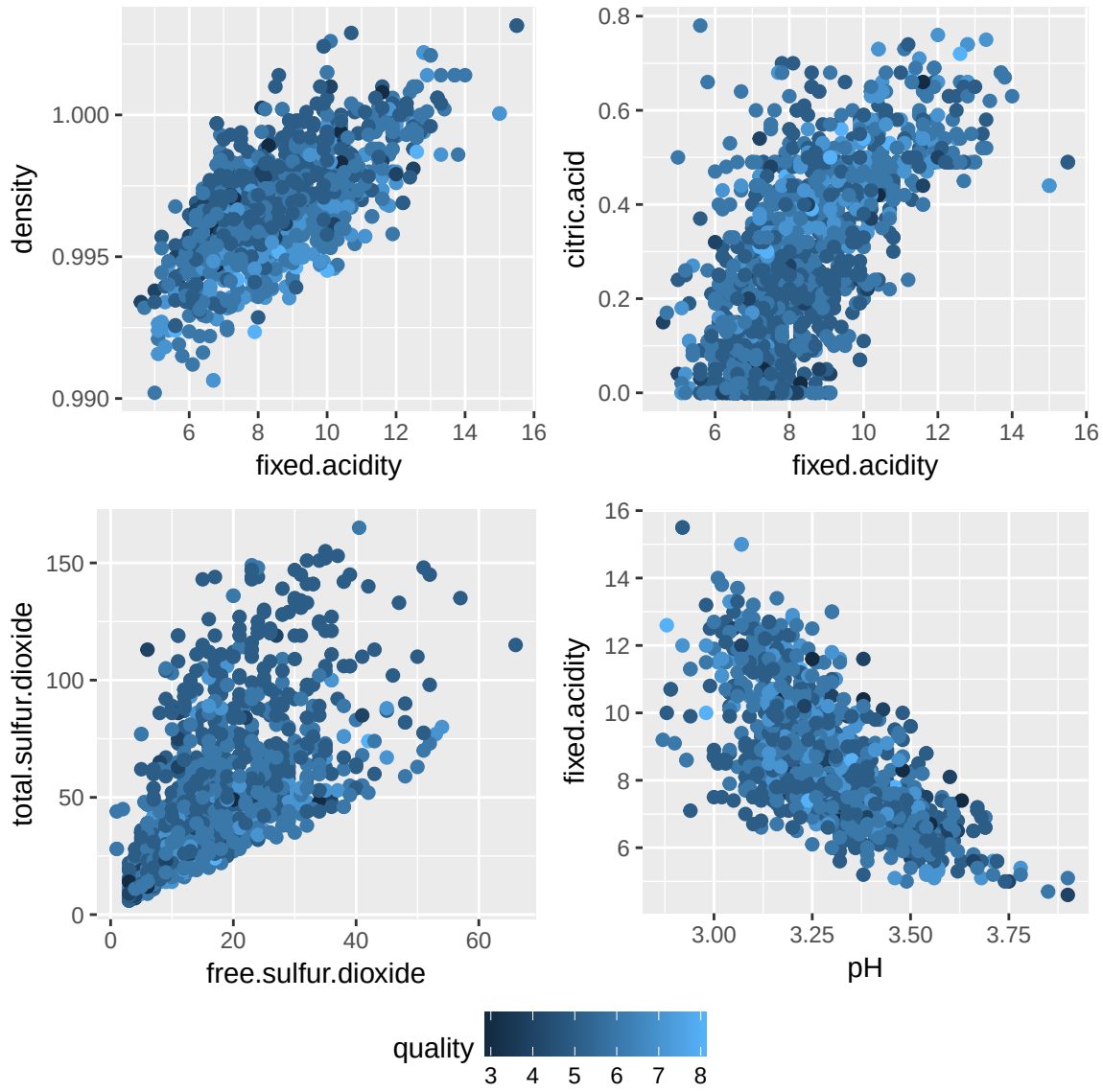
Hemos encontrado que hay varias variables correlacionadas entre si y otras que están correlaciones de manera positiva o negativa con la calidad del vino tinto. Vamos a representarlas:

```
p12<-ggplot(datos_limpios, aes(x=as.factor(quality), y=volatile.acidity, fill=quality)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Volatile acidity")


p13<-ggplot(datos_limpios, aes(x=as.factor(quality), y=sulphates, fill=quality)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Sulphates")

p14<-ggplot(datos_limpios, aes(x=as.factor(quality), y=alcohol, fill=quality)) +
    geom_boxplot()+
    xlab("Wine Quality") +
    ylab("Alcohol")

ggarrange(p12,p13,p14, ncol=3,
          nrow=1, common.legend = TRUE, legend="bottom")
```




```
p15<-ggplot(aes(x = fixed.acidity, y = density, colour = quality),
       data = datos_limpios) +
  geom_point(size = 2)

p16<-ggplot(aes(x = fixed.acidity, y = citric.acid, colour = quality),
       data = datos_limpios) +
  geom_point(size = 2)

p17<-ggplot(aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide, colour = quality),
       data = datos_limpios) +
  geom_point(size = 2)
```

```
p18<-ggplot(aes(x = pH, y = fixed.acidity, colour = quality),
       data = datos_limpios) +
  geom_point(size = 2)

ggarrange(p15,p16,p17,p18, ncol=2,
          nrow=2, common.legend = TRUE, legend="bottom")
```



En los gráficos anteriores se pueden ver claramente las relaciones entre las variables estudiadas.

6. Resolución del problema.

Después de analizar los datos, podemos resaltar las siguientes conlcusiones:

AUnque a priori el conjunto de datos a analizar parece sencillo, la manera de catar un vino es muy personal y aunque las personas que se dedican a ellos son expertas, siempre habrá un pequeño grado de subjetividad que será imposible de medir o predecir con nuestros modelos. Es por ello, que no hemos encontrado un modelo que se ajuste de manera fiable a los datos. La distribución de las calidades del vino también influye de manera

negativa en nuestro análisis, ya que la cantidad de vinos con calidad 3 es muy baja en comparación con las calidad medias (5-6). Sería más adecuado tener un dataset un poco más equilibrado en cuanto a la calidad del vino, pero por otro lado eso también nos viene a decir, que la gran mayoría de vino tintos tienen una calidad aceptable y que solo unos pocos tienen calidad deficiente. Lo mismo para para la calidades altas (8).

7. Código.

El codigo R usado se adjunta en este repositorio con el nombre 'PRA2.Rmd'.