

PRA2 - Limpieza y validación de datos

Antonio Caparrini Lopez

26/12/2019

Contents

Detalles de la actividad	1
Descripción	1
Objetivos	1
Competencias	2
Resolución	2
Descripción del dataset	2
Importancia y objetivos de los análisis	2
Limpieza de los datos	2
Análisis de los datos	7
Pruebas estadísticas	9
Conclusiones	21

Detalles de la actividad

Descripción

Como parte de la asignatura *tipología de datos y ciclo de vida de los datos* dentro del máster en ciencia de datos de la UOC este documento elabora un caso práctico que consiste en el tratamiento de un conjunto de datos utilizando las herramientas disponibles para su limpieza, validación y análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos. Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Competencias

En esta práctica se desarrollan las siguientes competencias del **Master de Data Science**:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Resolución

Descripción del dataset

```
winedataset <- read.csv("data/winequality-red.csv")
```

El conjunto de datos elegido para el análisis es el *Red Wine Quality* disponible en **kaggle**. Este conjunto de datos tiene un total de 1599 registros y un total de 12 columnas que pasamos a comentar a continuación:

1. **fixed acidity**: La no-volatilidad de los ácidos presentes en el vino (que no se evaporan con facilidad).
2. **volatile acidity**: Cantidad de ácido acético en el vino. Niveles muy altos llevan a un sabor desagradable a vinagre.
3. **citric acid**: En pequeñas cantidades puede dar frescura a un vino.
4. **residual sugar**: El nivel de azúcar después de que pare la fermentación del vino. Es raro encontrar vinos con una menor a 1g/litro y los vinos con más de 45g/litro se consideran dulces.
5. **chlorides**: Cantidad de sal en el vino.
6. **free sulfur dioxide**: Cantidad de dióxido de azufre libre.
7. **total sulfur dioxide**: Cantidad de dióxido de azufre libre y ligado.
8. **density**: Suele ser cercana a la densidad del agua dependiendo de los niveles de alcohol o azúcar.
9. **pH**: Describe lo ácido o básico que es el vino. Desde 0 (muy ácido) a 14 (muy básico). Suele encontrarse en torno a 3-4.
10. **sulphates**: Aditivos al vino que pueden contribuir al gas dióxido de azufre que tiene efectos antimicrobianos y antioxidantes.
11. **alcohol**: Porcentaje de alcohol.
12. **quality**: Variable de salida, medida entre 0 y 10.

Para más detalles sobre los datos referimos al trabajo original Cortez et al., 2009.

Importancia y objetivos de los análisis

A partir del dataset se pretender determinar las variables más relevantes a la hora de valorar la calidad de un vino. También podremos generar un modelo de regresión que partiendo de las variables del vino estime la calidad.

Este análisis es relevante para el campo de la enología (estudio del vino). Un sommelier (por ejemplo) que deba hacer una cata de vinos necesitaría una cantidad de tiempo elevada para catar una gran cantidad de vinos mientras que con un modelo previo que estime los mejores podría dedicar su esfuerzo y criterio experto en los que previamente van a ser mejores debido a sus características.

Limpieza de los datos

Los datos los habíamos leído previamente en el apartado anterior, y mostramos un resumen de su contenido.

```
summary(winedataset)
```

```
## 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
## 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
## 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
## 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
## Min.      :3.000
## 1st Qu.:5.000
## Median :6.000
## Mean   :5.636
## 3rd Qu.:6.000
## Max.   :8.000
```

No vamos a descartar ninguna de las variables por el momento. En primera instancia vamos a considerarlas todas como potencialmente relevantes.

Es importante ver el tipo de variables que tenemos.

```
str(winedataset)
```

```
## '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 ...
```

Vemos que las 11 variables que utilizamos para predecir son variables numéricas continuas y la variable *quality* que es el objetivo es una variable numérica entera.

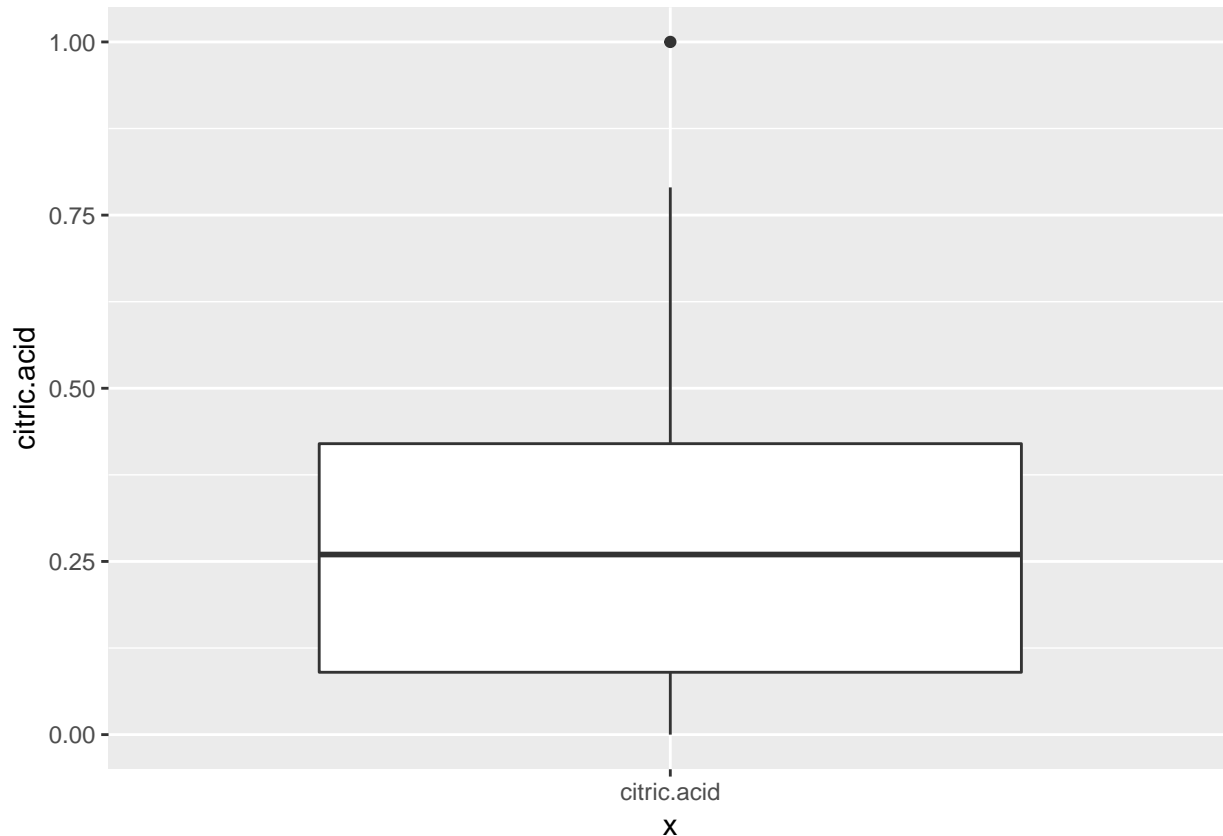
Valores nulos o vacíos.

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

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

Ninguna de las variables parece tener registros nulos o vacíos, sin embargo en ocasiones el valor cero puede indicar valores perdidos o nulos. Observando el sumario mostrado anteriormente de las variables vemos que solo la variable citric.acid llega a tener el valor cero. Vamos a comprobar con un boxplot que no hay muchos valores outliers hacia el cero.

```
ggplot(winedataset, aes(x="citric.acid", y=citric.acid))+geom_boxplot()
```



Las demás variables queda descartado que tengan valores nulos por ceros, y en el caso de citric.acid no parecen que el valor cero sea incorrecto. Lo que si podemos percibir en este boxplot es un punto que parece un valor extremo (“outlier”).

Valores extremos

Los valores extremos (“outliers”) son valores que debido a ser considerablemente distintos al resto de los datos podría considerarse que no perteneces a la población. Para identificarlos podríamos dibujar los boxplots

para todas las variables (como se ha visto en el apartado anterior para la variable citric.acid) pero también podemos utilizar la función `boxplot.stats()` que nos muestra directamente los valores extremos:

```
for (c in colnames(winedataset)){
  print(c)
  print(boxplot.stats(winedataset[,c])$out)
}
```

```
## [1] "fixed.acidity"
## [1] 12.8 12.8 15.0 15.0 12.5 13.3 13.4 12.4 12.5 13.8 13.5 12.6 12.5 12.8
## [15] 12.8 14.0 13.7 13.7 12.7 12.5 12.8 12.6 15.6 12.5 13.0 12.5 13.3 12.4
## [29] 12.5 12.9 14.3 12.4 15.5 15.5 15.6 13.0 12.7 13.0 12.7 12.4 12.7 13.2
## [43] 13.2 13.2 15.9 13.3 12.9 12.6 12.6
## [1] "volatile.acidity"
## [1] 1.130 1.020 1.070 1.330 1.330 1.040 1.090 1.040 1.240 1.185 1.020
## [12] 1.035 1.025 1.115 1.020 1.020 1.580 1.180 1.040
## [1] "citric.acid"
## [1] 1
## [1] "residual.sugar"
## [1] 6.10 6.10 3.80 3.90 4.40 10.70 5.50 5.90 5.90 3.80 5.10
## [12] 4.65 4.65 5.50 5.50 5.50 5.50 7.30 7.20 3.80 5.60 4.00
## [23] 4.00 4.00 4.00 7.00 4.00 4.00 6.40 5.60 5.60 11.00 11.00
## [34] 4.50 4.80 5.80 5.80 3.80 4.40 6.20 4.20 7.90 7.90 3.70
## [45] 4.50 6.70 6.60 3.70 5.20 15.50 4.10 8.30 6.55 6.55 4.60
## [56] 6.10 4.30 5.80 5.15 6.30 4.20 4.20 4.60 4.20 4.60 4.30
## [67] 4.30 7.90 4.60 5.10 5.60 5.60 6.00 8.60 7.50 4.40 4.25
## [78] 6.00 3.90 4.20 4.00 4.00 4.00 6.60 6.00 6.00 3.80 9.00
## [89] 4.60 8.80 8.80 5.00 3.80 4.10 5.90 4.10 6.20 8.90 4.00
## [100] 3.90 4.00 8.10 8.10 6.40 6.40 8.30 8.30 4.70 5.50 5.50
## [111] 4.30 5.50 3.70 6.20 5.60 7.80 4.60 5.80 4.10 12.90 4.30
## [122] 13.40 4.80 6.30 4.50 4.50 4.30 4.30 3.90 3.80 5.40 3.80
## [133] 6.10 3.90 5.10 5.10 3.90 15.40 15.40 4.80 5.20 5.20 3.75
## [144] 13.80 13.80 5.70 4.30 4.10 4.10 4.40 3.70 6.70 13.90 5.10
## [155] 7.80
## [1] "chlorides"
## [1] 0.176 0.170 0.368 0.341 0.172 0.332 0.464 0.401 0.467 0.122 0.178
## [12] 0.146 0.236 0.610 0.360 0.270 0.039 0.337 0.263 0.611 0.358 0.343
## [23] 0.186 0.213 0.214 0.121 0.122 0.122 0.128 0.120 0.159 0.124 0.122
## [34] 0.122 0.174 0.121 0.127 0.413 0.152 0.152 0.125 0.122 0.200 0.171
## [45] 0.226 0.226 0.250 0.148 0.122 0.124 0.124 0.143 0.222 0.039 0.157
## [56] 0.422 0.034 0.387 0.415 0.157 0.157 0.243 0.241 0.190 0.132 0.126
## [67] 0.038 0.165 0.145 0.147 0.012 0.012 0.039 0.194 0.132 0.161 0.120
## [78] 0.120 0.123 0.123 0.414 0.216 0.171 0.178 0.369 0.166 0.166 0.136
## [89] 0.132 0.132 0.123 0.123 0.123 0.403 0.137 0.414 0.166 0.168 0.415
## [100] 0.153 0.415 0.267 0.123 0.214 0.214 0.169 0.205 0.205 0.039 0.235
## [111] 0.230 0.038
## [1] "free.sulfur.dioxide"
## [1] 52 51 50 68 68 43 47 54 46 45 53 52 51 45 57 50 45 48 43 48 72 43 51
## [24] 51 52 55 55 48 48 66
## [1] "total.sulfur.dioxide"
## [1] 145 148 136 125 140 136 133 153 134 141 129 128 129 128 143 144 127
## [18] 126 145 144 135 165 124 124 134 124 129 151 133 142 149 147 145 148
## [35] 155 151 152 125 127 139 143 144 130 278 289 135 160 141 141 133 147
## [52] 147 131 131 131
## [1] "density"
```

```
## [1] 0.99160 0.99160 1.00140 1.00150 1.00150 1.00180 0.99120 1.00220
## [9] 1.00220 1.00140 1.00140 1.00140 1.00140 1.00320 1.00260 1.00140
## [17] 1.00315 1.00315 1.00315 1.00210 1.00210 0.99170 0.99220 1.00260
## [25] 0.99210 0.99154 0.99064 0.99064 1.00289 0.99162 0.99007 0.99007
## [33] 0.99020 0.99220 0.99150 0.99157 0.99080 0.99084 0.99191 1.00369
## [41] 1.00369 1.00242 0.99182 1.00242 0.99182
## [1] "pH"
## [1] 3.90 3.75 3.85 2.74 3.69 3.69 2.88 2.86 3.74 2.92 2.92 2.92 3.72 2.87
## [15] 2.89 2.89 2.92 3.90 3.71 3.69 3.69 3.71 3.71 2.89 2.89 3.78 3.70 3.78
## [29] 4.01 2.90 4.01 3.71 2.88 3.72 3.72
## [1] "sulphates"
## [1] 1.56 1.28 1.08 1.20 1.12 1.28 1.14 1.95 1.22 1.95 1.98 1.31 2.00 1.08
## [15] 1.59 1.02 1.03 1.61 1.09 1.26 1.08 1.00 1.36 1.18 1.13 1.04 1.11 1.13
## [29] 1.07 1.06 1.06 1.05 1.06 1.04 1.05 1.02 1.14 1.02 1.36 1.36 1.05 1.17
## [43] 1.62 1.06 1.18 1.07 1.34 1.16 1.10 1.15 1.17 1.17 1.33 1.18 1.17 1.03
## [57] 1.17 1.10 1.01
## [1] "alcohol"
## [1] 14.00000 14.00000 14.00000 14.00000 14.90000 14.00000 13.60000
## [8] 13.60000 13.60000 14.00000 14.00000 13.56667 13.60000
## [1] "quality"
## [1] 8 8 8 8 8 3 8 8 8 3 8 3 8 3 3 8 8 8 8 8 3 3 8 8 3 3 3 8
```

Podemos observar (entre otros) que el citric.acid con valor 1 es un outlier. Para no eliminar estos datos del dataset vamos a asignarles el valor de percentil 5% a los que sean valores extremos inferiores y el valor del percentil 95% a los valores extremos superiores.

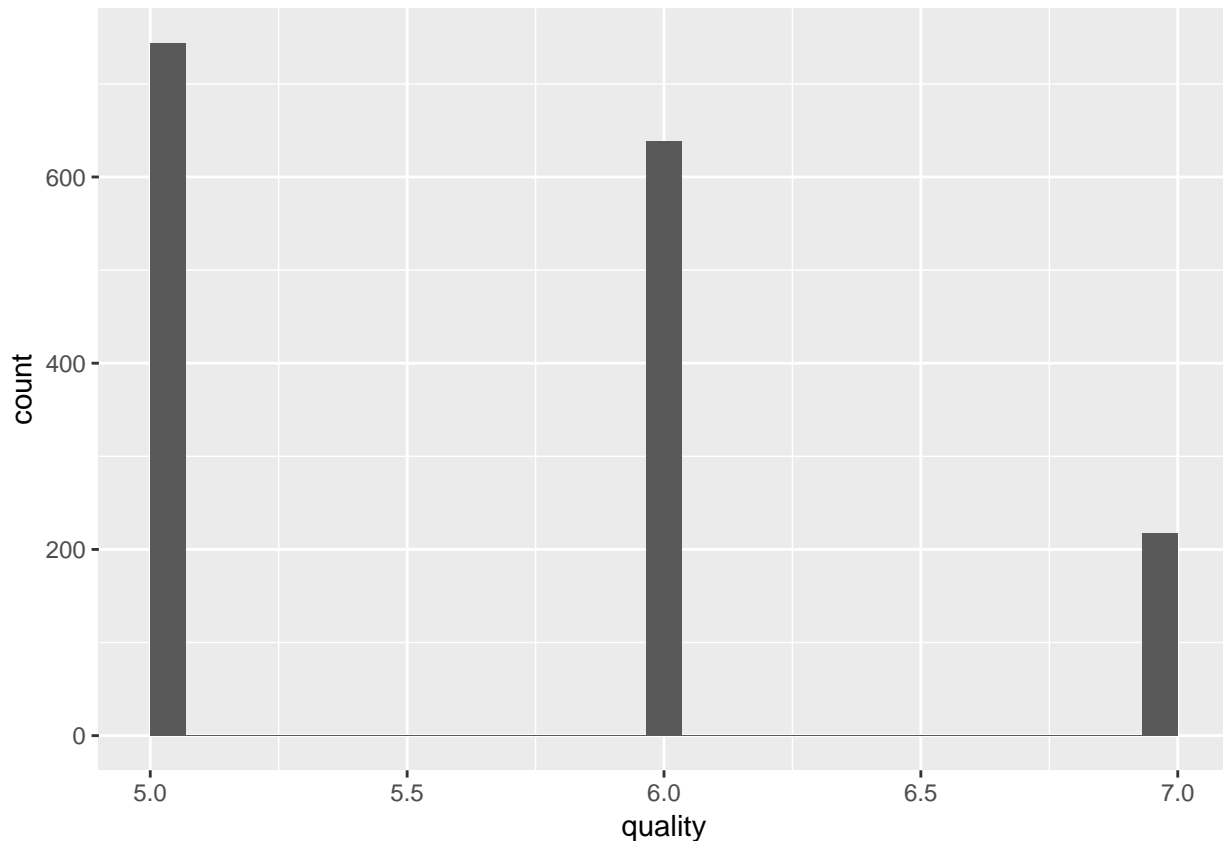
```
cut_outliers <- function(x){
  quantiles <- quantile( x, c(.05, .95))
  x[ x < quantiles[1]] <- quantiles[1]
  x[ x > quantiles[2]] <- quantiles[2]
  return(x)
}

winedataset <- as.data.frame( sapply(winedataset, cut_outliers) )
```

La variable objetivo tenía muchos valores extremos que hemos filtrado también, quedando de la siguiente manera.

```
ggplot(winedataset, aes(x=quality)) + geom_histogram()

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



Por ello vamos a tratar esta variable como una variable categórica con 3 clases.

Análisis de los datos

Comprobación de normalidad

Comprobamos si alguna de las variables cumple las características de una distribución normal. Para ello aplicamos el test de Shaphiro-Wilk. Aplicando este test la hipótesis nula es que la distribución cumple normalidad, por ello, para no rechazar la hipótesis nula el p-valor tiene que ser mayor que el nivel de significación elegido ($\alpha=0.05$).

```
alpha <- 0.05
for(n in names(winedataset)[-12]){
  pvalue <- shapiro.test(winedataset[,n])$p.value
  if(pvalue>alpha){
    cat("Variable ", n, " SI es normal\n")
    print(shapiro.test(winedataset[,n]))
  }else{
    cat("Variable ", n, " NO cumple una distribución normal\n")
  }
}
```

```
## Variable fixed.acidity NO cumple una distribución normal
## Variable volatile.acidity NO cumple una distribución normal
## Variable citric.acid NO cumple una distribución normal
## Variable residual.sugar NO cumple una distribución normal
## Variable chlorides NO cumple una distribución normal
```

```
## Variable free.sulfur.dioxide NO cumple una distribución normal
## Variable total.sulfur.dioxide NO cumple una distribución normal
## Variable density NO cumple una distribución normal
## Variable pH NO cumple una distribución normal
## Variable sulphates NO cumple una distribución normal
## Variable alcohol NO cumple una distribución normal
```

Vemos que según este test ninguna variable cumple una distribución normal, probamos con el test de Anderson-Darling.

```
alpha <- 0.05
for(n in names(winedataset)[-12]){
  pvalue <- ad.test(winedataset[,n])$p.value
  if(pvalue>alpha){
    cat("Variable ", n, " SI es normal\n")
    print(shapiro.test(winedataset[,n]))
  }else{
    cat("Variable ", n, " NO cumple una distribución normal\n")
  }
}
```

```
## Variable fixed.acidity NO cumple una distribución normal
## Variable volatile.acidity NO cumple una distribución normal
## Variable citric.acid NO cumple una distribución normal
## Variable residual.sugar NO cumple una distribución normal
## Variable chlorides NO cumple una distribución normal
## Variable free.sulfur.dioxide NO cumple una distribución normal
## Variable total.sulfur.dioxide NO cumple una distribución normal
## Variable density NO cumple una distribución normal
## Variable pH NO cumple una distribución normal
## Variable sulphates NO cumple una distribución normal
## Variable alcohol NO cumple una distribución normal
```

Comprobación de homogeneidad en la varianza

Dado que las variables que tenemos no siguen una distribución normal vamos a utilizar el test de Levene que es ampliamente utilizado en la literatura para comprobar la homogeneidad de las varianzas de las variables.

```
alpha <- 0.05
for(n in names(winedataset)[-12]){
  pvalue <- leveneTest(winedataset[,n] ~ as.factor(winedataset$quality))$Pr[1]
  if(is.nan(pvalue)){
    pvalue <- 0
  }
  if(pvalue>alpha){
    cat("Variable ", n, " SI tiene una varianza homogénea\n")
    print(leveneTest(winedataset[,n] ~ as.factor(winedataset$quality))$Pr[1])
  }else{
    cat("Variable ", n, " NO tiene una varianza homogénea\n")
  }
}
```

```
## Variable fixed.acidity NO tiene una varianza homogénea
## Variable volatile.acidity NO tiene una varianza homogénea
## Variable citric.acid NO tiene una varianza homogénea
## Variable residual.sugar NO tiene una varianza homogénea
```



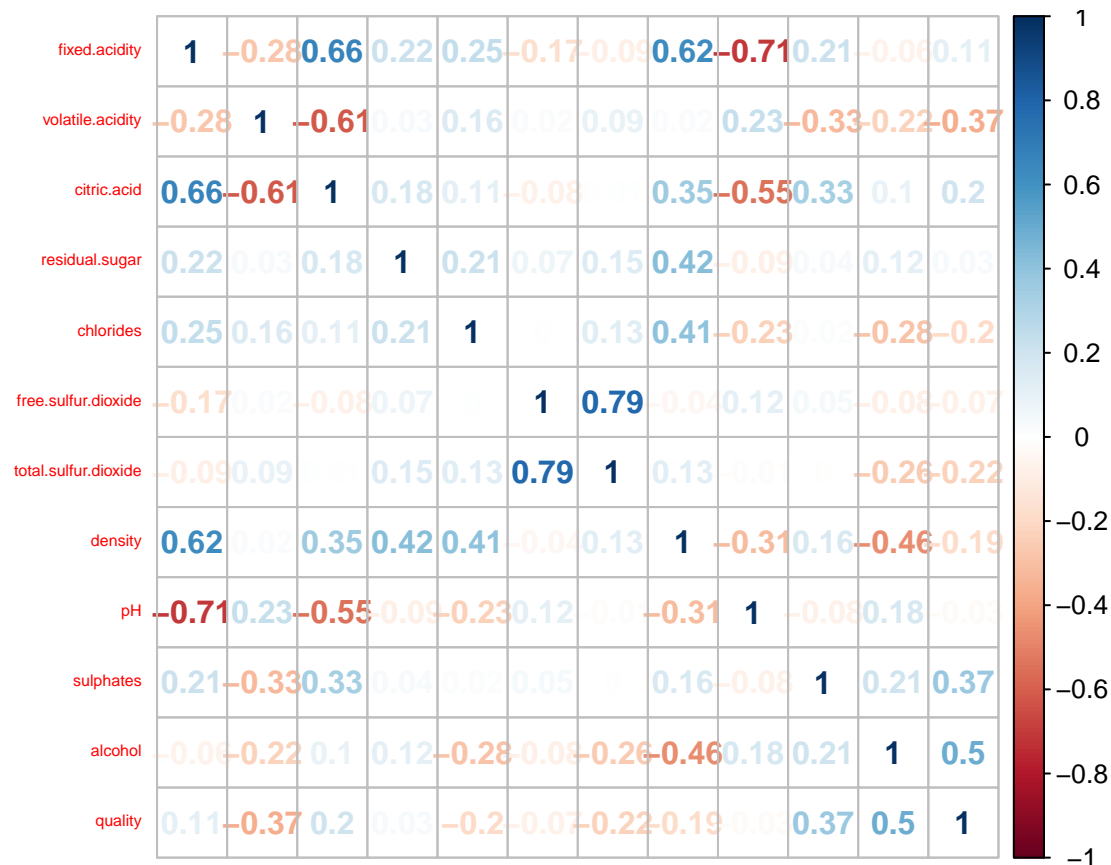
```
## Variable chlorides SI tiene una varianza homogénea
## [1] 0.819282
## Variable free.sulfur.dioxide NO tiene una varianza homogénea
## Variable total.sulfur.dioxide NO tiene una varianza homogénea
## Variable density NO tiene una varianza homogénea
## Variable pH SI tiene una varianza homogénea
## [1] 0.467888
## Variable sulphates SI tiene una varianza homogénea
## [1] 0.1310031
## Variable alcohol NO tiene una varianza homogénea
```

Tenemos 3 variables que tienen una varianza homogénea chlorides, pH y sulphates.

Pruebas estadísticas

Es interesante comprobar la capacidad explicativa de la *quality* para cada variable. Para ello vamos a calcular los coeficientes de correlación de Spearman. Es necesario utilizar los de Spearman ya que nos encontramos con variables que no siguen una distribución normal.

```
corrplot(cor(winedataset, method = c("spearman")), method = "number", tl.pos = 'l', tl.cex= 0.5)
```



Nos encontramos con dos variables que están positivamente correlacionadas con la calidad que son *alcohol* y *sulphates*. Estas dos variables tienen la mayor correlación positiva por lo que a mayor nivel de estas dos mayor calidad en el vino.

Por otro lado la variable *volatile.acidity* tiene la mayor correlación negativa indicando que a mayor nivel de esta la calidad es menor. Como veíamos en la descripción de las variables esta es la cantidad de ácido acético

en el vino que a mayor cantidad, mayor sabor a vinagre. Este relación es por tanto razonable.

Correlación positiva algo inferior tenemos *fixed.acidity* y *citric.acid*.

Y con una correlación negativa pero también pequeña *chlorides*, *total.sulfur.dioxide* y *density*.

Regresión lineal

Vamos a crear un modelo de regresión lineal que estime la calidad del vino. Para ello vamos a crear 11 modelos distintos y quedarnos con el mejor. Como criterio de construcción vamos a ir añadiendo las variables con el coeficiente de correlación mayor en valor absoluto.

```
winedataset$quality <- sapply(winedataset$quality, as.numeric)

cor_data <- cor(winedataset, method = c("spearman"))

quality_cor <- sort(abs(cor_data[, "quality"]))
vars_names <- names(quality_cor)

model11 <- lm(quality ~ residual.sugar + pH + free.sulfur.dioxide
  + fixed.acidity + density +
  chlorides + total.sulfur.dioxide + citric.acid
  + sulphates + volatile.acidity + alcohol,
  data=winedataset)
model10 <- lm(quality ~ pH + free.sulfur.dioxide + fixed.acidity + density +
  chlorides + total.sulfur.dioxide + citric.acid
  + sulphates + volatile.acidity + alcohol,
  data=winedataset)
model9 <- lm(quality ~ free.sulfur.dioxide + fixed.acidity + density +
  chlorides + total.sulfur.dioxide + citric.acid
  + sulphates + volatile.acidity + alcohol,
  data=winedataset)
model8 <- lm(quality ~ fixed.acidity + density +
  chlorides + total.sulfur.dioxide + citric.acid
  + sulphates + volatile.acidity + alcohol,
  data=winedataset)
model7 <- lm(quality ~ density +
  chlorides + total.sulfur.dioxide + citric.acid
  + sulphates + volatile.acidity + alcohol,
  data=winedataset)
model6 <- lm(quality ~ chlorides + total.sulfur.dioxide +
  citric.acid + sulphates + volatile.acidity +
  alcohol, data=winedataset)
model5 <- lm(quality ~ total.sulfur.dioxide + citric.acid +
  sulphates + volatile.acidity + alcohol, data=winedataset)
model4 <- lm(quality ~ citric.acid + sulphates +
  volatile.acidity + alcohol, data=winedataset)
model3 <- lm(quality ~ sulphates + volatile.acidity + alcohol, data=winedataset)
model2 <- lm(quality ~ volatile.acidity + alcohol, data=winedataset)
model1 <- lm(quality ~ alcohol, data=winedataset)
```

En la tabla a continuación visualizamos el valor de R^2 de todos los modelos. EL mejor modelo sería el que

tenga el valor mayor que en este caso es el que utiliza todas las variables.

```
tabla.coeficientes <- matrix(c(
  1, summary(model11)$r.squared,
  2, summary(model12)$r.squared,
  3, summary(model13)$r.squared,
  4, summary(model14)$r.squared,
  5, summary(model15)$r.squared,
  6, summary(model16)$r.squared,
  7, summary(model17)$r.squared,
  8, summary(model18)$r.squared,
  9, summary(model19)$r.squared,
  10, summary(model110)$r.squared,
  11, summary(model111)$r.squared), ncol = 2, byrow = TRUE)

colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes
```

```
##      Modelo      R^2
## [1,]      1 0.2709799
## [2,]      2 0.3372867
## [3,]      3 0.3744801
## [4,]      4 0.3754321
## [5,]      5 0.3902865
## [6,]      6 0.3931586
## [7,]      7 0.3935414
## [8,]      8 0.3999502
## [9,]      9 0.4009950
## [10,]     10 0.4021839
## [11,]     11 0.4032291
```

El anterior modelo era de regresión, ahora vamos a crear un modelo de regresión pero de clasificación multilabel.

En la siguiente tabla vemos el valor de AIC de todos los modelos creados, el que tenga una AIC menor será el mejor modelo.

```
tabla.coeficientes <- matrix(c(
  1, summary(mmodel11)$AIC,
  2, summary(mmodel12)$AIC,
  3, summary(mmodel13)$AIC,
  4, summary(mmodel14)$AIC,
  5, summary(mmodel15)$AIC,
  6, summary(mmodel16)$AIC,
  7, summary(mmodel17)$AIC,
  8, summary(mmodel18)$AIC,
  9, summary(mmodel19)$AIC,
  10, summary(mmodel110)$AIC,
  11, summary(mmodel111)$AIC), ncol = 2, byrow = TRUE)

colnames(tabla.coeficientes) <- c("Modelo", "AIC")
tabla.coeficientes
```

```
##      Modelo      AIC
## [1,]      1 2715.967
## [2,]      2 2571.760
## [3,]      3 2487.231
```

```
## [4,]      4 2478.310
## [5,]      5 2439.622
## [6,]      6 2436.532
## [7,]      7 2438.258
## [8,]      8 2429.983
## [9,]      9 2427.672
## [10,]     10 2428.645
## [11,]     11 2420.196
```

Con el proposito de tener una comparación visual de estos dos modelos seleccionamos aleatoriamente 10 registros del conjunto de datos y mostramos una tabla con la calidad real, la detectada por el modelo de regresión y por el modelo de clasificación.

```
my_index <- sample(1:1599,10)
elements <- winedataset[my_index,]
pred_model <- predict(model11, elements)
pred_mmodel <- predict(mmodel6, elements)
cbind(my_index, pred_model, pred_mmodel, label=winedataset$quality[my_index])
```

```
##      my_index pred_model pred_mmodel label
## 395      395   5.496674          1      5
## 1377     1377   5.165442          1      5
## 1266     1266   5.609463          2      6
## 2         2    5.286374          1      5
## 545      545   5.823876          2      6
## 1311     1311   5.148868          1      5
## 653      653   6.697636          3      5
## 1330     1330   5.134245          1      6
## 290      290   5.666374          1      5
## 367      367   5.688232          1      7
```

A pesar de que las etiquetas reales son enteras el modelo de regresión nos da una información más detallada ya que existen valores continuos entre una clase y otra (que podría relacionarse con la probabilidad de pertenecer a una clase que el modelo de clasificación acaba redondeando).

Árbol de decisión

Vamos a probar un árbol de decisión, en primer lugar para ello partimos el conjunto en conjunto para entrenar y el conjunto donde comprobaremos la capacidad predictiva del modelo.

```
winedataset$quality <- sapply(winedataset$quality, as.factor)

train_index <- sample(1:nrow(winedataset), 0.8 * nrow(winedataset))
test_index <- setdiff(1:nrow(winedataset), train_index)

X_train <- winedataset[train_index, -12]
y_train <- winedataset[train_index, 12]

X_test <- winedataset[test_index, -12]
y_test <- winedataset[test_index, 12]

model <- C50::C5.0(X_train, y_train)
summary(model)
```

```
##
## Call:
```

```

## C5.0.default(x = X_train, y = y_train)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jan  6 22:13:44 2020
## -----
##
## Class specified by attribute `outcome'
##
## Read 1279 cases (12 attributes) from undefined.data
##
## Decision tree:
##
## alcohol > 10.5:
## :...sulphates <= 0.63:
## :   :...alcohol <= 11.4:
## :   :   :...sulphates > 0.59:
## :   :   :   :...pH > 3.48: 5 (5/1)
## :   :   :   :   pH <= 3.48:
## :   :   :   :   :...sulphates <= 0.6: 6 (11)
## :   :   :   :   :   sulphates > 0.6:
## :   :   :   :   :   :...total.sulfur.dioxide > 46: 5 (3)
## :   :   :   :   :   :   total.sulfur.dioxide <= 46:
## :   :   :   :   :   :   :...total.sulfur.dioxide <= 18: 5 (3/1)
## :   :   :   :   :   :   :   total.sulfur.dioxide > 18: 6 (8)
## :   :   :   sulphates <= 0.59:
## :   :   :   :...free.sulfur.dioxide > 26: 6 (5)
## :   :   :   :   free.sulfur.dioxide <= 26:
## :   :   :   :   :...volatile.acidity > 0.59:
## :   :   :   :   :   :...alcohol > 10.8: 5 (21/1)
## :   :   :   :   :   :   alcohol <= 10.8:
## :   :   :   :   :   :   :...free.sulfur.dioxide <= 8: 5 (7/1)
## :   :   :   :   :   :   :   free.sulfur.dioxide > 8: 6 (4)
## :   :   :   :   volatile.acidity <= 0.59:
## :   :   :   :   :...pH > 3.45: 5 (6)
## :   :   :   :   :   pH <= 3.45:
## :   :   :   :   :   :...chlorides <= 0.066:
## :   :   :   :   :   :   :...chlorides <= 0.059: 5 (2/1)
## :   :   :   :   :   :   :   chlorides > 0.059: 7 (5/1)
## :   :   :   :   :   :   chlorides > 0.066:
## :   :   :   :   :   :   :...alcohol <= 11:
## :   :   :   :   :   :   :   :...residual.sugar <= 1.9: 6 (2/1)
## :   :   :   :   :   :   :   :   residual.sugar > 1.9: 5 (10/1)
## :   :   :   :   :   :   :   alcohol > 11:
## :   :   :   :   :   :   :   :...total.sulfur.dioxide <= 11: 5 (3/1)
## :   :   :   :   :   :   :   :   total.sulfur.dioxide > 11: 6 (5)
## :   :   :   alcohol > 11.4:
## :   :   :   :...volatile.acidity <= 0.49:
## :   :   :   :   :...residual.sugar <= 3.9: 6 (48/14)
## :   :   :   :   :   residual.sugar > 3.9:
## :   :   :   :   :   :...pH <= 3.25: 7 (6)
## :   :   :   :   :   :   pH > 3.25:
## :   :   :   :   :   :   :...total.sulfur.dioxide <= 16: 5 (2/1)
## :   :   :   :   :   :   :   total.sulfur.dioxide > 16: 6 (2)
## :   :   :   :   volatile.acidity > 0.49:

```

```

## : :      :...free.sulfur.dioxide <= 4:
## : :      :...fixed.acidity <= 6.3: 7 (2)
## : :      :   fixed.acidity > 6.3: 5 (3)
## : :      free.sulfur.dioxide > 4:
## : :      :...free.sulfur.dioxide > 31: 7 (2)
## : :      free.sulfur.dioxide <= 31:
## : :      :...fixed.acidity > 7: 6 (10/1)
## : :      fixed.acidity <= 7:
## : :      :...alcohol <= 11.5: 6 (5)
## : :      alcohol > 11.5:
## : :      :...citric.acid <= 0.04:
## : :      :...residual.sugar <= 2.15: 6 (7)
## : :      :   residual.sugar > 2.15:
## : :      :   :...pH <= 3.53: 5 (4)
## : :      :       pH > 3.53: 6 (2)
## : :      citric.acid > 0.04:
## : :      :...alcohol <= 11.6: 6 (2)
## : :      alcohol > 11.6:
## : :      :...citric.acid <= 0.12: 5 (7)
## : :      citric.acid > 0.12:
## : :      :...residual.sugar <= 2.15: 5 (2)
## : :      residual.sugar > 2.15: 6 (3/1)
## : sulphates > 0.63:
## : :...alcohol > 11.6:
## : :      :...total.sulfur.dioxide <= 17:
## : :      :   :...fixed.acidity <= 11.6: 7 (17)
## : :      :       fixed.acidity > 11.6: 6 (2)
## : :      total.sulfur.dioxide > 17:
## : :      :...sulphates <= 0.69:
## : :      :   :...alcohol > 12.3: 6 (11)
## : :      :       alcohol <= 12.3:
## : :      :       :...alcohol <= 11.95: 6 (5/1)
## : :      :           alcohol > 11.95: 7 (3)
## : :      sulphates > 0.69:
## : :      :...fixed.acidity > 11.6: 5 (2/1)
## : :      fixed.acidity <= 11.6:
## : :      :...free.sulfur.dioxide <= 18:
## : :      :   :...free.sulfur.dioxide <= 9: 6 (6/1)
## : :      :       free.sulfur.dioxide > 9: 7 (27/2)
## : :      free.sulfur.dioxide > 18:
## : :      :...residual.sugar <= 2.55: 6 (22/8)
## : :      residual.sugar > 2.55: 7 (3)
## : alcohol <= 11.6:
## : :...total.sulfur.dioxide > 95: 5 (6/1)
## : total.sulfur.dioxide <= 95:
## : :...volatile.acidity > 0.37:
## : :   :...free.sulfur.dioxide > 20: 6 (38/3)
## : :   free.sulfur.dioxide <= 20:
## : :   :...chlorides > 0.092:
## : :   :   :...density <= 0.99619: 6 (2)
## : :   :       density > 0.99619:
## : :   :       :...sulphates <= 0.75: 5 (7)
## : :   :           sulphates > 0.75: 6 (3/1)
## : :   chlorides <= 0.092:

```

```

## :           :      :...total.sulfur.dioxide > 65: 5 (2)
## :           :      : total.sulfur.dioxide <= 65:
## :           :      :...free.sulfur.dioxide > 19: 7 (3/1)
## :           :      : free.sulfur.dioxide <= 19:
## :           :      :...chlorides <= 0.064:
## :           :      :...citric.acid > 0.28: 7 (3/1)
## :           :      :   citric.acid <= 0.28:
## :           :      :   :...fixed.acidity <= 8.5: 6 (7/1)
## :           :      :       fixed.acidity > 8.5: 5 (2)
## :           :      : chlorides > 0.064:
## :           :      :...pH > 3.56: 7 (2)
## :           :      : pH <= 3.56:
## :           :      :...chlorides > 0.071: 6 (21)
## :           :      : chlorides <= 0.071:
## :           :      :...pH > 3.49: 6 (4)
## :           :      : pH <= 3.49:
## :           :      :...fixed.acidity <= 10.4: 7 (5/1)
## :           :      : fixed.acidity > 10.4: 6 (3)
## : volatile.acidity <= 0.37:
## : :...total.sulfur.dioxide > 56:
## : :   :...density > 0.99845: 7 (2)
## : :   : density <= 0.99845:
## : :   : :...total.sulfur.dioxide <= 74: 5 (6/1)
## : :   :       total.sulfur.dioxide > 74: 6 (4/1)
## : : total.sulfur.dioxide <= 56:
## : : :...chlorides > 0.095: 6 (5)
## : :   chlorides <= 0.095:
## : : :...chlorides > 0.082: 7 (7)
## : :   chlorides <= 0.082:
## : : :...citric.acid > 0.53: 6 (5/1)
## : :   citric.acid <= 0.53:
## : : :...density <= 0.99414: 6 (3/1)
## : :   density > 0.99414:
## : : :...density <= 0.99542: 7 (12)
## : :   density > 0.99542:
## : : :...pH <= 3.17: 7 (6)
## : :   pH > 3.17:
## : : :...chlorides <= 0.06: 7 (3)
## : :   chlorides > 0.06:
## : : :...chlorides <= 0.071: 6 (10)
## : :   chlorides > 0.071: [S1]
## alcohol <= 10.5:
## :...volatile.acidity <= 0.32:
## :   :...fixed.acidity > 11.6:
## :   :   :...residual.sugar <= 2.4: 7 (6)
## :   :   : residual.sugar > 2.4: 6 (2)
## :   : fixed.acidity <= 11.6:
## :   : :...density <= 0.99654: 5 (10/2)
## :   :   density > 0.99654:
## :   : :...alcohol > 10.4:
## :   :   :...sulphates <= 0.72: 5 (2/1)
## :   :   : sulphates > 0.72: 7 (4)
## :   : alcohol <= 10.4:
## :   : :...citric.acid <= 0.52: 6 (32/4)

```

```

##      :      citric.acid > 0.52:
##      :      :...pH <= 3.18: 6 (3)
##      :      pH > 3.18: 5 (5/1)
## volatile.acidity > 0.32:
## :...total.sulfur.dioxide > 95: 5 (92/4)
##      total.sulfur.dioxide <= 95:
##      :...sulphates <= 0.53: 5 (152/26)
##      sulphates > 0.53:
##      :...alcohol > 9.8:
##      :...total.sulfur.dioxide > 80: 5 (10/1)
##      :      total.sulfur.dioxide <= 80:
##      :      :...residual.sugar > 4.5:
##      :      :      :...sulphates <= 0.65: 6 (3)
##      :      :      sulphates > 0.65:
##      :      :      :...fixed.acidity <= 7.3: 5 (3)
##      :      :      fixed.acidity > 7.3: 7 (4)
##      :      residual.sugar <= 4.5:
##      :      :...pH > 3.42:
##      :      :      :...sulphates > 0.87: 6 (2)
##      :      :      sulphates <= 0.87:
##      :      :      :...free.sulfur.dioxide > 25: 5 (7)
##      :      :      free.sulfur.dioxide <= 25:
##      :      :      :...total.sulfur.dioxide <= 31: 5 (12/2)
##      :      :      total.sulfur.dioxide > 31:
##      :      :      :...density > 0.99803: 5 (2)
##      :      :      density <= 0.99803:
##      :      :      :...pH > 3.51: 6 (3)
##      :      :      pH <= 3.51:
##      :      :      :...fixed.acidity <= 7.2: 6 (2)
##      :      :      fixed.acidity > 7.2: 5 (2)
##      :      pH <= 3.42:
##      :      :...density <= 0.99542: 6 (10)
##      :      density > 0.99542:
##      :      :...residual.sugar <= 1.8:
##      :      :      :...pH <= 3.06: 7 (3)
##      :      :      pH > 3.06:
##      :      :      :...residual.sugar > 1.75: [S2]
##      :      :      residual.sugar <= 1.75:
##      :      :      :...chlorides > 0.109: 5 (3)
##      :      :      chlorides <= 0.109:
##      :      :      :...chlorides > 0.085: 6 (3)
##      :      :      chlorides <= 0.085:
##      :      :      :...density > 0.99678: 5 (4)
##      :      :      density <= 0.99678: [S3]
##      :      residual.sugar > 1.8:
##      :      :...sulphates <= 0.62:
##      :      :      :...density <= 0.99555: 5 (5)
##      :      :      density > 0.99555:
##      :      :      :...fixed.acidity <= 8.7:
##      :      :      :      :...pH <= 3.21: 5 (2)
##      :      :      :      pH > 3.21: [S4]
##      :      :      fixed.acidity > 8.7: [S5]
##      :      sulphates > 0.62:
##      :      :...chlorides > 0.1: [S6]

```



```

##          : chlorides <= 0.1:
##          : ...alcohol <= 10.03333: [S7]
##          : alcohol > 10.03333:
##          : ...chlorides > 0.088: 6 (7)
##          : chlorides <= 0.088:
##          : ...alcohol <= 10.3: [S8]
##          : alcohol > 10.3: [S9]
## alcohol <= 9.8:
## ...fixed.acidity > 10.6:
##     : ...residual.sugar <= 2.6: 6 (15/2)
##     : residual.sugar > 2.6: 5 (3)
## fixed.acidity <= 10.6:
##     : ...citric.acid > 0.37: 5 (45/8)
##     : citric.acid <= 0.37:
##     : ...pH <= 3.12: 5 (9)
##     : pH > 3.12:
##     : ...citric.acid > 0.23:
##     :     : ...chlorides <= 0.071: 5 (5/1)
##     :     : chlorides > 0.071:
##     :     : ...alcohol > 9.4:
##     :     :     : ...fixed.acidity <= 8.4: 6 (21/2)
##     :     :     : fixed.acidity > 8.4: 5 (5/1)
##     :     :     : alcohol <= 9.4:
##     :     :     : ...residual.sugar > 2.3: 6 (5)
##     :     :     : residual.sugar <= 2.3:
##     :     :     : ...density > 0.99803: 6 (4)
##     :     :     : density <= 0.99803:
##     :     :     : ...pH <= 3.14: 6 (2)
##     :     :     : pH > 3.14:
##     :     :     : ...density > 0.99665: 5 (11)
##     :     :     : density <= 0.99665: [S10]
##     citric.acid <= 0.23:
##     : ...free.sulfur.dioxide > 23: 5 (20/1)
##     : free.sulfur.dioxide <= 23:
##     : ...residual.sugar > 4: 6 (5)
##     : residual.sugar <= 4:
##     : ...pH > 3.38:
##     :     : ...volatile.acidity > 0.655: 5 (15)
##     :     : volatile.acidity <= 0.655:
##     :     :     : ...residual.sugar <= 2.05: [S11]
##     :     :     : residual.sugar > 2.05: [S12]
##     : pH <= 3.38:
##     : ...fixed.acidity <= 6.7: 5 (7)
##     : fixed.acidity > 6.7:
##     :     : ...pH > 3.35: [S13]
##     :     : pH <= 3.35:
##     :     :     : ...volatile.acidity <= 0.605: [S14]
##     :     :     : volatile.acidity > 0.605: [S15]
##
## SubTree [S1]
##
## volatile.acidity <= 0.31: 7 (3)
## volatile.acidity > 0.31: 6 (4)
##

```

```

## SubTree [S2]
##
## free.sulfur.dioxide <= 15: 7 (3)
## free.sulfur.dioxide > 15: 5 (2)
##
## SubTree [S3]
##
## total.sulfur.dioxide <= 33: 6 (3)
## total.sulfur.dioxide > 33: 5 (2)
##
## SubTree [S4]
##
## total.sulfur.dioxide <= 13: 5 (2)
## total.sulfur.dioxide > 13: 6 (28/5)
##
## SubTree [S5]
##
## free.sulfur.dioxide <= 12: 5 (11/1)
## free.sulfur.dioxide > 12:
## :...residual.sugar <= 2.1: 5 (3)
##     residual.sugar > 2.1: 6 (5)
##
## SubTree [S6]
##
## free.sulfur.dioxide > 13: 6 (3/1)
## free.sulfur.dioxide <= 13:
## :...fixed.acidity <= 11.6: 5 (6/1)
##     fixed.acidity > 11.6: 6 (2/1)
##
## SubTree [S7]
##
## free.sulfur.dioxide <= 26: 6 (11/1)
## free.sulfur.dioxide > 26: 7 (2)
##
## SubTree [S8]
##
## citric.acid <= 0.39: 5 (3)
## citric.acid > 0.39:
## :...chlorides <= 0.072: 5 (3/1)
##     chlorides > 0.072: 6 (3)
##
## SubTree [S9]
##
## alcohol <= 10.4: 6 (7)
## alcohol > 10.4:
## :...residual.sugar <= 2.2: 6 (3)
##     residual.sugar > 2.2: 5 (2)
##
## SubTree [S10]
##
## citric.acid <= 0.35: 5 (2)
## citric.acid > 0.35: 6 (2)
##
## SubTree [S11]

```

```

##
## total.sulfur.dioxide <= 50: 5 (20/1)
## total.sulfur.dioxide > 50: 6 (3/1)
##
## SubTree [S12]
##
## residual.sugar <= 2.25: 6 (6)
## residual.sugar > 2.25:
## :...alcohol <= 9.4: 6 (2)
##     alcohol > 9.4:
##     :...pH <= 3.49: 6 (2)
##         pH > 3.49: 5 (8)
##
## SubTree [S13]
##
## total.sulfur.dioxide > 40: 5 (4)
## total.sulfur.dioxide <= 40:
## :...sulphates <= 0.57: 5 (2/1)
##     sulphates > 0.57: 6 (10/1)
##
## SubTree [S14]
##
## total.sulfur.dioxide <= 26: 6 (11/1)
## total.sulfur.dioxide > 26:
## :...free.sulfur.dioxide <= 11: 5 (5)
##     free.sulfur.dioxide > 11:
##     :...density <= 0.99656: 5 (4/1)
##         density > 0.99656: 6 (6)
##
## SubTree [S15]
##
## sulphates > 0.55: 5 (25/4)
## sulphates <= 0.55:
## :...alcohol > 9.55: 5 (3)
##     alcohol <= 9.55:
##     :...residual.sugar <= 2.85: 6 (7)
##         residual.sugar > 2.85: 5 (3/1)
##
##
## Evaluation on training data (1279 cases):
##
##      Decision Tree
##      -----
##      Size      Errors
##
##      153  130(10.2%)  <<
##
##
##      (a)  (b)  (c)  <-classified as
##      ----  ----  ----
##      574   21   2    (a): class 5
##      61   448   4    (b): class 6
##      9    33   127   (c): class 7
##

```

```
##
## Attribute usage:
##
## 100.00% alcohol
## 89.13% volatile.acidity
## 88.27% sulphates
## 80.84% total.sulfur.dioxide
## 44.88% fixed.acidity
## 43.24% pH
## 40.89% free.sulfur.dioxide
## 38.62% residual.sugar
## 32.29% citric.acid
## 23.61% density
## 21.34% chlorides
##
##
## Time: 0.0 secs
```

Comprobamos el poder predictivo.

```
predictions_train <- predict(model, X_train, type="class")
accuracy_train <- 100*sum(predictions_train == y_train) / length(predictions_train)
predictions_test <- predict(model, X_test, type="class")
accuracy_test <- 100*sum(predictions_test == y_test) / length(predictions_test)
print(sprintf("La precisión del árbol en entrenamiento es: %.4f %%", accuracy_train))
```

```
## [1] "La precisión del árbol en entrenamiento es: 89.8358 %"
```

```
print(sprintf("La precisión del árbol en prueba es: %.4f %%", accuracy_test))
```

```
## [1] "La precisión del árbol en prueba es: 70.6250 %"
```

Vemos que sobre el conjunto de entrenamiento la accuracy ha sido mejor pero se degrada sobre el conjunto de prueba.

```
cm <- table(y_test,Predicted=predictions_test)
#El paquete caret nos ofrece estadísticas sobre la matrix de confusión
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##      Predicted
## y_test   5    6    7
##      5 120   26    1
##      6   35   78   12
##      7    6   14   28
##
```

```
## Overall Statistics
```

```
##
```

```
##              Accuracy : 0.7062
##              95% CI : (0.653, 0.7556)
##      No Information Rate : 0.5031
##      P-Value [Acc > NIR] : 1.241e-13
##
```

```
##              Kappa : 0.515
##      McNemar's Test P-Value : 0.1679
##
```

```
## Statistics by Class:
##
##           Class: 5 Class: 6 Class: 7
## Sensitivity      0.7453   0.6610   0.6829
## Specificity      0.8302   0.7673   0.9283
## Pos Pred Value   0.8163   0.6240   0.5833
## Neg Pred Value    0.7630   0.7949   0.9522
## Prevalence       0.5031   0.3688   0.1281
## Detection Rate    0.3750   0.2437   0.0875
## Detection Prevalence 0.4594   0.3906   0.1500
## Balanced Accuracy 0.7878   0.7142   0.8056
```

Conclusiones

En primer lugar se ha verificado que las variables del dataset no contenían valores nulos y que todas eran candidatas a aportar información en el análisis. Hemos cambiado los valores outliers por los percentiles 5% y 95% respectivamente para los valores extremos inferiores y superiores, entre los que también se encontraba la variable objetivo.

Se ha confirmado mediante test estadísticos la no normalidad de las variables y se ha empleado los coeficientes de correlación de Spearman para ver las variables más influyentes en la calidad. De esta manera hemos visto que el grado de alcohol y los sulfitos tienen un efecto positivo en la calidad, mientras que la cantidad de ácido acético influye en la disminución de la calidad.

Por otro lado se han creado dos modelos de regresión para estimar la calidad. Estos modelos no tienen una gran capacidad predictiva pero como se menciona en la introducción pueden ser utilizados para disminuir la cantidad de vinos que ha de catar un experto a los inicialmente estimados por el modelo.

Finalmente hemos creado un árbol de decisión con una precisión balanceada del 70% entre las 3 clases que es razonable ya que un clasificador que tuviera un criterio aleatorio al elegir entre tres clases tendría una precisión del 33%.