Práctica2: Limpieza y análisis de datos

Red Wine Quality

Estudio análisis calidad del vino

Tipología del ciclo de datos

Curso 2020-21

Jordi Lladó Valero

José Ángel Ubieto Pitarque

Índice

1. Introducción y detalle de la actividad

1.1. Descripción

La siguiente actividad se realizará un caso práctico de tratamiento de conjunto de datos (dataset) que puede ser, el creado en la práctica 1 anterior o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com). El objetivo principal es aprender a identificar los datos relevantes para un proyecto analítico y utilizar las herramientas de integración, limpieza, validación y análisis de estas que aporta la programación con R.

1.2. 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.

1.3. Competencias

En esta práctica se desarrollan las siguientes competencias del Máster 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.

2. Desarrollo de la práctica

Para la realización de la práctica se escogió la base de datos winequality-red que se encuentra en el repositorio www.kaggle.com (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009). La información proporcionada en la web sobre esta base de datos es la siguiente:

"Los dos conjuntos de datos están relacionados con variantes tinto y blanco del vino portugués "Vinho Verde". Para más detalles, consulte la referencia [Cortez et al., 2009]. Debido a cuestiones de privacidad y logística, solo están disponibles las variables fisicoquímicas (entradas) y sensoriales (la salida) (por ejemplo, no hay datos sobre tipos de uva, marca de vino, precio de venta del vino, etc.). Estos conjuntos de datos pueden verse como tareas de clasificación o regresión. Las clases están ordenadas y no equilibradas (por ejemplo, hay vinos mucho más normales que excelentes o malos)."

Después de observar el archivo de la base de datos, solo hay un único archivo .csv referente a los datos de los distintos vinos rojos. Esta base de datos está formada por 12 atributos y un total de 1599 entradas. También cabe considerar que el siguiente estudio se podría ampliar buscando la base de datos de vino blanco.

Todo el desarrollo de la práctica y los archivos relacionados se encontrarán en el siguiente enlace Github:

https://github.com/master-ciencia-datos/PR2-limpieza-de-datos

2.1. Descripción de los datos

Se procede a la descripción de los distintos atributos de la base de datos winequality-red.csv. Las variables de entrada, basadas en test físico-químicos [Cortez et al., 2009] son:

fixed_acidity: conjunto de los ácidos naturales procedentes de la uva (tartárico, málico, cítrico y succínico) o formados en la fermentación maloláctica (láctico). … sumada a la acidez volátil, da como resultado la acidez total de un vino. (tartaric acid - g / dm^3)

volatile_acidity: proviene de los ácidos de cadena corta de la serie acética (acético, fórmico, propiónico, butírico) y de algunas de sus combinaciones como el acetato de etilo originados durante la fermentación, que pueden proporcionar al vino el desagradable olor y sabor a "picado" arruinando la producción. (acetic acid - g / dm^3, Valores máximos aceptados 1.2-1.4 g/l)

citric_acid: (E-330) es un acidificante para corregir la acidez en mostos y vinos, además posee una acción estabilizante como antioxidante. El ácido cítrico forma complejos naturales con Fe(III), por tanto, su adición puede reforzar esta acción secuestrando una cierta cantidad del hierro contenido en el vino. (g / dm^3)

residual_sugar: es la cantidad total de azúcar que queda en el vino que no ha sido fermentada por las levaduras, y parte de ese azúcar no fermentado son las Pentosas, azúcares presentes en el vino en concentraciones cercanas a 1 gramo por litro de mosto. Las pentosas son: Arabinosa. Xilosa. Es difícil encontrar vinos con cantidades inferiores a 1 gramo/litro, i vinos con cantidades superiores a 45 gramos/litro son considerados dulces (g / dm^3)

chlorides: Cantidad de sal en el vino. El contenido de cloruros en los vinos es variable, en general es inferior a 0,5 g/l, expresado en cloruro de sodio límite máximo por Ley, y excepcionalmente puede pasar de 1 g/l, sobre todo en viñedos ubicados en terrenos salinos o cerca del mar. (sodium chloride - g / dm^3)

free_sulfur_dioxide: la forma libre de SO_2 existe en equilibrio entre el SO_2 molecular (disuelto con gas) y el ion bisulfito, previene del crecimiento microbiológico y de la oxidación del vino. (mg / dm^3)

total_sulfur_dioxide: cantidad total de SO_2 libre y enlazadas de SO_2. En concentraciones bajas, el SO_2 es raramente indetectable en el vino, pero el SO2 libre a concentraciones por encima de 50 ppm (mg/L) SO_2 se evidencia en el olfato y el gusto.

density: la densidad del agua es cercana es parecida a la del vino, dependiendo la cantidad de alcohol y azúcares que contenga (mg / dm^3).

pH: describe como de ácido o básico es un vino. Los valores van de 0 (muy ácido) a 14 (muy básico). La mayoría de los vinos se encuentran entre 3-4 de la escala del pH.

sulphates: son aditivos del vino. Pueden contribuir a niveles dióxido de sulfuro SO_2, que actúa como antimicrobiano y antioxidante. Además, contribuyen a activar la fermentación alcohólica y tienen efectos sobre la maceración, el color, el olor y el gusto del vino. (potassium sulphate - g / dm^3)

alcohol: porcentaje de alcohol que contiene el vino (% volumen)

quality: variable de salida (basada en datos sensoriales, sus valores van de 0 a 10)

2.2. Importancia del dataset

Después de la lectura inicial de la base de datos, se plantea que la cuestión ¿qué vino tiene mayor calidad? ¿Con esto se podría determinar que variables tienen mayor influencia en la calidad, por ejemplo, una mayor cantidad de alcohol implica una mejor calidad? ¿La calidad del vino es influida por la presencia de ácido cítrico? ¿La calidad del vino es superior si éste tiene menos contenido de ácidos volátiles?

Estudios anteriores, utilizaron este dataset para crear un modelo que pronostique la calidad de un vino según distintas variables físico-químicas. Por lo tanto, se podría observar que variables influyen más, que relación habría entre ellas e intentar crear un modelo de regresión predictivo de la calidad. Con este modelo permitiría con solo un

análisis físico-químico calificar un vino y en el caso de tener datos económicos poner un precio a un nuevo vino.

2.3. Lectura y limpieza de datos

Se procederá a la lectura del archivo 'winequality-red' que se encuentra en formato CSV. Para ello se adjuntará en el documento final el archivo original, sin tener así necesidad de ir al enlace de la base de datos. Como se puede observar en el siguiente código, la primera fila no se hará la lectura y se le darán los distintos nombres de los atributos mediante names(). También se realizará la lectura de las 5 primeras entradas (head()).

```
##### Apertura y exploración de datos

df <- read.csv('winequality-red.csv', sep = ",", stringsAsFactors = FALSE
, header = FALSE,  strip.white = T, skip=1)

names(df) <- c("fixed_acidity","volatile_acidity","citric_acid","residual
_sugar","chlorides","free_sulfur_dioxide","total_sulfur_dioxide","density
","pH","sulphates","alcohol","quality")

filas=dim(df)[1]
#Visualización de las 5 primeras entradas de la base de datos
head(df)
```

```
##   fixed_acidity volatile_acidity citric_acid residual_sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free_sulfur_dioxide total_sulfur_dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

A continuación, se mirará que tipo es cada atributo.

```
# Observación de los distintos tributos
sapply(df, function(x) class(x))
```

```
##        fixed_acidity     volatile_acidity          citric_acid
##            "numeric"            "numeric"            "numeric"
##       residual_sugar           chlorides  free_sulfur_dioxide
##            "numeric"            "numeric"            "numeric"
## total_sulfur_dioxide              density                   pH
##            "numeric"            "numeric"            "numeric"
##            sulphates              alcohol              quality
##            "numeric"            "numeric"            "integer"
```

Como se puede observar todas las variables son numéricas menos la variable quality que es entera. Seguiremos buscando si en la base de datos hay algún valor vacío o valor no asignado.

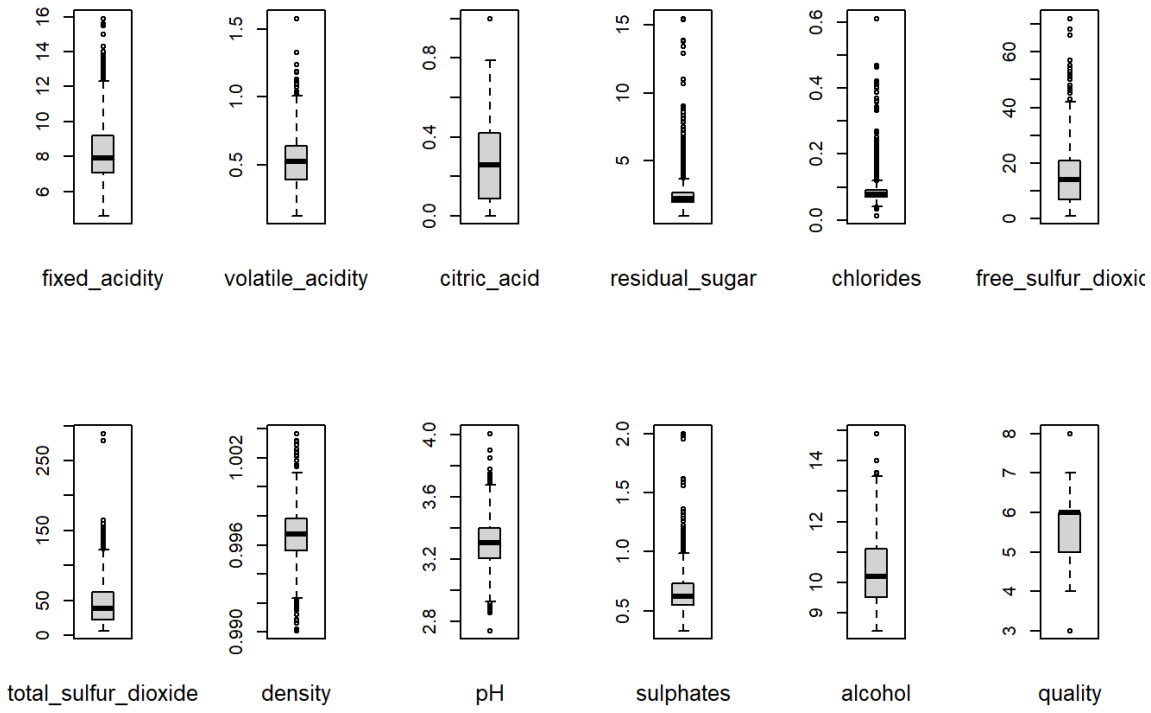
```
colSums(is.na(df))
```

```
##        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
```

En este caso, no hay ningún valor vacío o no asignado. En el caso de que hubiese, se podría tratar de distintas formas, por ejemplo, hay bases de datos que para un valor vacío utilizan el carácter '?', entonces se trataría este carácter primero asignándolo como N.A. (del inglés, Not Available) u 'otros' (dependiendo el tipo de variable que fuera). También se podrían imputar los valores N.A. como la media del resto de valores del atributo. Todos estos pasos, dependerían del tipo de dato con el que estuviéramos trabajando. También cabe considerar que habría otra forma para determinar los valores vacíos de la base de datos, esta sería mediante la siguiente instrucción: #sapply(df, function(x) sum(is.na(x)))

Por último, en este apartado se visualizarán y analizarán los distintos valores extremos que haya en el dataset mediante diagrama de caja y la función boxplot.stats().

```
# Estructura general gráfica 2 filas, 6 columnas
oldpar = par(mfrow = c(2,6))

#Realización de las distintas gráficas boxplot
for ( i in 1:12 ) {
  boxplot(df[[i]])
  mtext(names(df)[i], cex = 0.8, side = 1, line = 2)
}
```




```
boxplot.stats(df$fixed_acidity)$out
```

```
##  [1] 12.8 12.8 15.0 15.0 12.5 13.3 13.4 12.4 12.5 13.8 13.5 12.6 12.5 12.8 12.8
## [16] 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 12.5 12.9
## [31] 14.3 12.4 15.5 15.5 15.6 13.0 12.7 13.0 12.7 12.4 12.7 13.2 13.2 13.2 15.9
## [46] 13.3 12.9 12.6 12.6
```

```
boxplot.stats(df$volatile_acidity)$out
```

```
##  [1] 1.130 1.020 1.070 1.330 1.330 1.040 1.090 1.040 1.240 1.185 1.020 1.035
## [13] 1.025 1.115 1.020 1.020 1.580 1.180 1.040
```

```
boxplot.stats(df$citric_acid)$out
```

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

```
boxplot.stats(df$residual_sugar)$out
```

```
##   [1]  6.10  6.10  3.80  3.90  4.40 10.70  5.50  5.90  5.90  3.80  5.10  4.65
##  [13]  4.65  5.50  5.50  5.50  5.50  7.30  7.20  3.80  5.60  4.00  4.00  4.00
##  [25]  4.00  7.00  4.00  4.00  6.40  5.60  5.60 11.00 11.00  4.50  4.80  5.80
##  [37]  5.80  3.80  4.40  6.20  4.20  7.90  7.90  3.70  4.50  6.70  6.60  3.70
##  [49]  5.20 15.50  4.10  8.30  6.55  6.55  4.60  6.10  4.30  5.80  5.15  6.30
##  [61]  4.20  4.20  4.60  4.20  4.60  4.30  4.30  7.90  4.60  5.10  5.60  5.60
##  [73]  6.00  8.60  7.50  4.40  4.25  6.00  3.90  4.20  4.00  4.00  4.00  6.60
##  [85]  6.00  6.00  3.80  9.00  4.60  8.80  8.80  5.00  3.80  4.10  5.90  4.10
##  [97]  6.20  8.90  4.00  3.90  4.00  8.10  8.10  6.40  6.40  8.30  8.30  4.70
## [109]  5.50  5.50  4.30  5.50  3.70  6.20  5.60  7.80  4.60  5.80  4.10 12.90
## [121]  4.30 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 13.80
## [145] 13.80  5.70  4.30  4.10  4.10  4.40  3.70  6.70 13.90  5.10  7.80
```

```
boxplot.stats(df$chlorides)$out
```

```
##   [1] 0.176 0.170 0.368 0.341 0.172 0.332 0.464 0.401 0.467 0.122 0.178 0.146
##  [13] 0.236 0.610 0.360 0.270 0.039 0.337 0.263 0.611 0.358 0.343 0.186 0.213
##  [25] 0.214 0.121 0.122 0.122 0.128 0.120 0.159 0.124 0.122 0.122 0.174 0.121
##  [37] 0.127 0.413 0.152 0.152 0.125 0.122 0.200 0.171 0.226 0.226 0.250 0.148
##  [49] 0.122 0.124 0.124 0.143 0.222 0.039 0.157 0.422 0.034 0.387 0.415 0.157
##  [61] 0.157 0.243 0.241 0.190 0.132 0.126 0.038 0.165 0.145 0.147 0.012 0.012
##  [73] 0.039 0.194 0.132 0.161 0.120 0.120 0.123 0.123 0.414 0.216 0.171 0.178
##  [85] 0.369 0.166 0.166 0.136 0.132 0.132 0.123 0.123 0.123 0.403 0.137 0.414
##  [97] 0.166 0.168 0.415 0.153 0.415 0.267 0.123 0.214 0.214 0.169 0.205 0.205
## [109] 0.039 0.235 0.230 0.038
```

```
boxplot.stats(df$free_sulfur_dioxide)$out
```

```
## [1] 52 51 50 68 68 43 47 54 46 45 53 52 51 45 57 50 45 48 43 48 72 43 51 51 52
## [26] 55 55 48 48 66
```

```
boxplot.stats(df$total_sulfur_dioxide)$out
```

```
##  [1] 145 148 136 125 140 136 133 153 134 141 129 128 129 128 143 144 127 126 145
## [20] 144 135 165 124 124 134 124 129 151 133 142 149 147 145 148 155 151 152 125
## [39] 127 139 143 144 130 278 289 135 160 141 141 133 147 147 131 131 131
```

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

```
##  [1] 0.99160 0.99160 1.00140 1.00150 1.00150 1.00180 0.99120 1.00220 1.00220
## [10] 1.00140 1.00140 1.00140 1.00140 1.00320 1.00260 1.00140 1.00315 1.00315
## [19] 1.00315 1.00210 1.00210 0.99170 0.99220 1.00260 0.99210 0.99154 0.99064
## [28] 0.99064 1.00289 0.99162 0.99007 0.99007 0.99020 0.99220 0.99150 0.99157
## [37] 0.99080 0.99084 0.99191 1.00369 1.00369 1.00242 0.99182 1.00242 0.99182
```

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

```
##  [1] 3.90 3.75 3.85 2.74 3.69 3.69 2.88 2.86 3.74 2.92 2.92 2.92 3.72 2.87 2.89
## [16] 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 4.01 2.90
## [31] 4.01 3.71 2.88 3.72 3.72
```

```
boxplot.stats(df$sulphates)$out
```

```
##  [1] 1.56 1.28 1.08 1.20 1.12 1.28 1.14 1.95 1.22 1.95 1.98 1.31 2.00 1.08 1.59
## [16] 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 1.07 1.06
```

```
## [31] 1.06 1.05 1.06 1.04 1.05 1.02 1.14 1.02 1.36 1.36 1.05 1.17 1.62 1.06 1.18
## [46] 1.07 1.34 1.16 1.10 1.15 1.17 1.17 1.33 1.18 1.17 1.03 1.17 1.10 1.01
```

```
boxplot.stats(df$alcohol)$out
```

```
##  [1] 14.00000 14.00000 14.00000 14.00000 14.90000 14.00000 13.60000 13.60000
##  [9] 13.60000 14.00000 14.00000 13.56667 13.60000
```

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

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

Después de su análisis, las variables 'volatile_acidity', 'chlorides', 'total_sulfur_dioxide' y 'sulphates' muestran valores extremos muy alejados. En este caso, se podrían llegar a producir, es decir, estos valores podrían ser reproducibles en otros vinos y pueden tener significado, ya que por ejemplo valores elevados de ácidos volátiles ('volatile_acidity) o sulfitos (total_sulfur_dioxide) se relacionarían con una baja calidad del vino.

2.4. Análisis de datos

2.4.1. Selección de los grupos de datos a analizar/comparar

Inicialmente se utilizarán todas las variables del dataset para el análisis estadístico. Primero de todo se observarán los distintos grupos de calidad que ya están definidos, para ver cuantas observaciones hay en cada grupo.

```
#Cantidad de muestras según variable quality calidad
table(df$quality)
```

```
##
##   3   4   5   6   7   8
##  10  53 681 638 199  18
```

Como se puede observar, hay un total de 6 grupos, donde no aparecen las calidades inferiores a 3 y las superiores a 8, ya que en este dataSet no se ha dado el caso que hubiese vino de esta calidad. Por este motivo, se generará una nueva variable, que contenga menos grupos, pero se contemple las calidades extremas. Esta nueva variable se llamará 'gruposcalidad' y quedará definida como: calidad Baja B<=4 (valor 0), Media M>4 y <6 (valor 1), Buena B>=6 y <7 (valor 2), Muy Buena MB>=7 y <9 (valor 3) y

Premium P>=9 (valor 4). Con estos grupos, se podrán realizar distintos tests de hipótesis.

```
#Creación grupos calidad según quality
df$gruposcalidad[df$quality<=4]<-0
df$gruposcalidad[df$quality>4 & df$quality<6]<-1
df$gruposcalidad[df$quality>=6 & df$quality<7]<-2
df$gruposcalidad[df$quality>=7 & df$quality<9]<-3
df$gruposcalidad[df$quality>=9]<-4

#Cantidad de observaciones según la variable gruposcalidad
table(df$gruposcalidad)

##
##   0   1   2   3
##  63 681 638 217
```

En este caso, se habrían generado 5 grupos, y vemos que al menos 4 de ellos contienen observaciones. Sólo el grupo con calidad Premium ha quedado sin datos. Estos resultados se tendrán en cuenta a la hora de hacer los distintos tests de hipótesis.

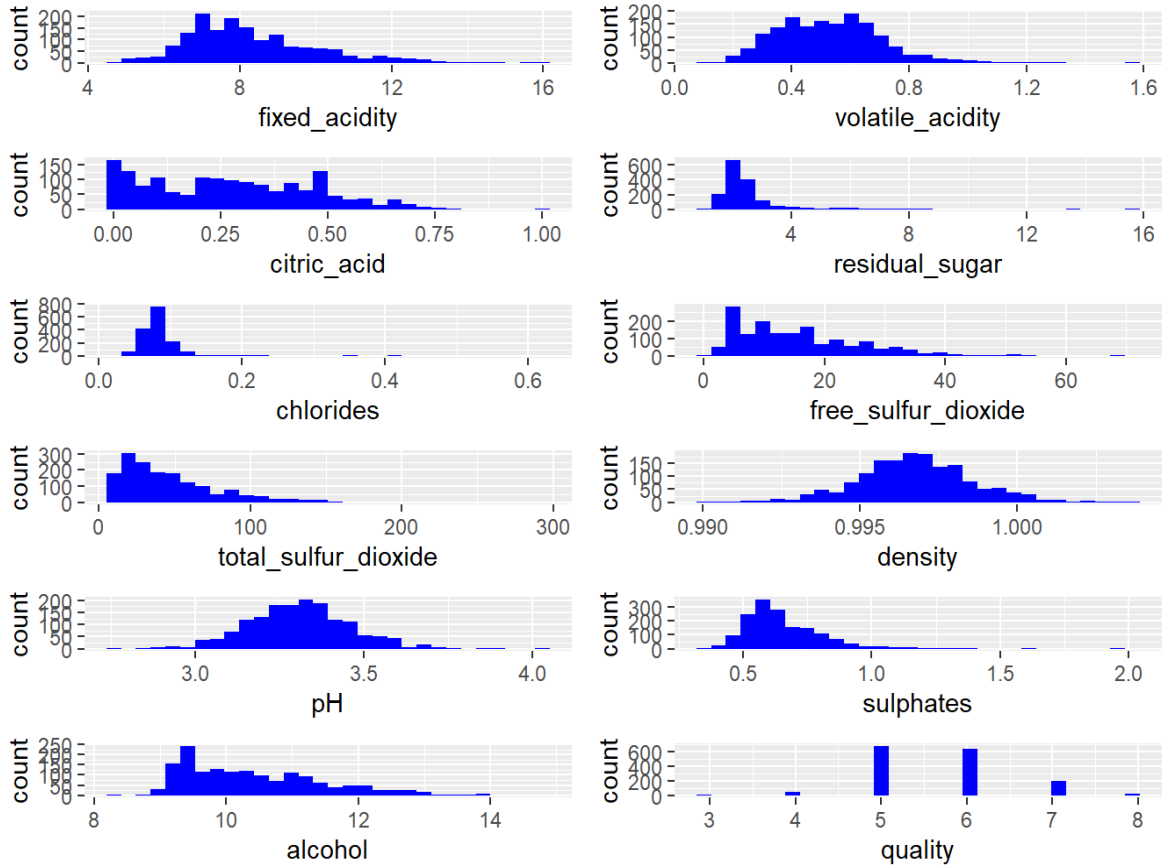
Para el resto de los atributos del dataset se buscarán si hay correlaciones entre ellos y se intentará realizar un modelo de regresión lineal con el objetivo de determinar la calidad del vino según los distintos parámetros.

2.4.2. Comprobación de la normalidad y homogeneidad de la varianza

Para la comprobación de la normalidad de los datos, primero se harán distintos histogramas de las variables para tener una primera visualización de su distribución.

```
p1<-ggplot(df, aes(x = fixed_acidity)) + geom_histogram(fill = "blue")
p2<-ggplot(df, aes(x = volatile_acidity)) + geom_histogram(fill = "blue")
p3<-ggplot(df, aes(x = citric_acid)) + geom_histogram(fill = "blue")
p4<-ggplot(df, aes(x = residual_sugar)) + geom_histogram(fill = "blue")
p5<-ggplot(df, aes(x = chlorides)) + geom_histogram(fill = "blue")
p6<-ggplot(df, aes(x = free_sulfur_dioxide)) + geom_histogram(fill = "blu
e")
p7<-ggplot(df, aes(x = total_sulfur_dioxide)) + geom_histogram(fill = "bl
ue")
p8<-ggplot(df, aes(x = density)) + geom_histogram(fill = "blue")
p9<-ggplot(df, aes(x = pH)) + geom_histogram(fill = "blue")
p10<-ggplot(df, aes(x = sulphates)) + geom_histogram(fill = "blue")
p11<-ggplot(df, aes(x = alcohol)) + geom_histogram(fill = "blue")
```

```
p12<-ggplot(df, aes(x = quality)) + geom_histogram(fill = "blue")

grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, nrow = 6)
```



Como se puede observar, pocas de las variables podrían seguir una distribución normal, entre ellas podrían ser 'volatile_acidity', 'density' y 'pH'. Con el objetivo de verificar la suposición de la normalidad, se realizarán los tests de **Shapiro-Wilk** y de **Kolmogorov-Smirnov** para estas variables.

```
#Comprobación de la normalidad por Shapiro.test
shapiro.test(df$volatile_acidity)

##
##  Shapiro-Wilk normality test
##
## data:  df$volatile_acidity
## W = 0.97434, p-value = 2.693e-16
```

```
shapiro.test(df$density)

##
##  Shapiro-Wilk normality test
##
## data:  df$density
## W = 0.99087, p-value = 1.936e-08

shapiro.test(df$pH)

##
##  Shapiro-Wilk normality test
##
## data:  df$pH
## W = 0.99349, p-value = 1.712e-06

shapiro.test(df$quality)

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

ks.test(df$volatile_acidity, pnorm, mean(df$volatile_acidity), sd(df$vola
tile_acidity))

## Warning in ks.test(df$volatile_acidity, pnorm, mean(df$volatile_acidit
y), : ties
## should not be present for the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  df$volatile_acidity
## D = 0.054662, p-value = 0.0001416
## alternative hypothesis: two-sided

ks.test(df$density, pnorm, mean(df$density), sd(df$density))

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

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  df$density
## D = 0.044787, p-value = 0.003274
## alternative hypothesis: two-sided
```

```
ks.test(df$pH, pnorm, mean(df$pH), sd(df$pH))

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

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  df$pH
## D = 0.040368, p-value = 0.01091
## alternative hypothesis: two-sided

ks.test(df$quality, pnorm, mean(df$quality), sd(df$quality))

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

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  df$quality
## D = 0.24982, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Como se puede observar en los tests Shapiro-Wilk y Kolmogorov-Smirnov, los valores p-value obtenidos han sido inferiores a alpha=0.05 (nivel de significancia) indicando que la hipótesis nula es rechazada y por tanto los datos no cuentan con una distribución normal. Se verificarán también para el resto de las variables con una adaptación de la función que planteó T. Gutiérrez (2017).

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

```
## Variables que no siguen una distribución normal:
## fixed_acidity, volatile_acidity, citric_acid,
## residual_sugar, chlorides, free_sulfur_dioxide,
## total_sulfur_dioxide, density, pH,
## sulphates, alcohol, quality
## gruposcalidad
```

Como se puede verificar, todas las variables no siguen una distribución normal. Para el test de homogeneidad de las varianzas se aplicará el test de **Fligner-Killeen** ya que los datos no siguen la distribución normal. En este test, la hipótesis nula asume igualdad de varianzas en los diferentes grupos de datos, por lo que p-valores inferiores al nivel de significancia indicarán heterocedasticidad. En este caso, primero se realizará en modo comparativo el test de la variable citric_acid con la variable quality y con la variable gruposcalidad, para observar cómo influye la agrupación con distintos niveles. Después se realizará una tabla con los valores p_value obtenidos de aplicar el test Fligner-Killen de las distintas variables con los distintos grupos de la variable gruposcalidad.

```
#Comparación del test fligner con variable quality
fligner.test(citric_acid ~ quality, data = df)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  citric_acid by quality
## Fligner-Killeen:med chi-squared = 10.916, df = 5, p-value = 0.05307

fligner.test(citric_acid ~ gruposcalidad, data = df)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  citric_acid by gruposcalidad
## Fligner-Killeen:med chi-squared = 10.293, df = 3, p-value = 0.01623

#tabla->Cálculo p_val de fligner test de las variables con los distintos
de la variable gruposcalidad
tab.pvalue <- matrix(c('fixed_acidity', fligner.test(fixed_acidity ~ grup
oscalidad, data = df)$p.value,
                       'volatile_acidity', fligner.test(volatile_acidity
~ gruposcalidad, data = df)$p.value,
                       'citric_acid', fligner.test(citric_acid ~ gruposc
```

```
alidad, data = df)$p.value,
                          'residual_sugar', fligner.test(residual_sugar ~ g
ruposcalidad, data = df)$p.value,
                          'chlorides', fligner.test(chlorides~ gruposcalida
d, data = df)$p.value,
                          'free_sulfur_dioxide', fligner.test(free_sulfur_d
ioxide ~ gruposcalidad, data = df)$p.value,
                          'total_sulfur_dioxide', fligner.test(total_sulfur
_dioxide ~ gruposcalidad, data = df)$p.value,
                          'density', fligner.test(density ~ gruposcalidad,
data = df)$p.value,
                          'pH', fligner.test(pH ~ gruposcalidad, data = df)
$p.value,
                          'sulphates', fligner.test(sulphates ~ gruposcalid
ad, data = df)$p.value,
                          'alcohol', fligner.test(alcohol ~ gruposcalidad,
data = df)$p.value),
                         ncol = 2, byrow = TRUE)
colnames(tab.pvalue) <- c("variable", "p_value")
tab.pvalue
```

```
##       variable               p_value
##  [1,] "fixed_acidity"        "1.87021032897887e-07"
##  [2,] "volatile_acidity"     "4.13053032723214e-07"
##  [3,] "citric_acid"          "0.0162326521434384"
##  [4,] "residual_sugar"       "0.0398591053295684"
##  [5,] "chlorides"            "0.161129569591473"
##  [6,] "free_sulfur_dioxide"  "0.00337275061796985"
##  [7,] "total_sulfur_dioxide" "2.66038926216217e-29"
##  [8,] "density"              "1.83078551649716e-10"
##  [9,] "pH"                   "0.922078395972653"
## [10,] "sulphates"            "0.041474433590482"
## [11,] "alcohol"              "6.56967463884807e-28"
```

Se puede observar un cambio, por el hecho de realizar el test Fligner-Killeen respeto a la variable quality o a la variable gruposcalidad. Por ejemplo, en la variable ácido cítrico respecto a quality las varianzas entre los grupos son similares (p_value>0.05), en cambio, respecto a la variable gruposcalidad las varianzas entre grupos son distintas (p_value<0.05). Podemos decir, que el hecho de tener los datos en menos grupos, ha producido en este caso que las varianzas entre grupos sean distintas.

Por lo que hace las variables'chlorites' y 'pH' con los niveles de la variable gruposcalidad, tienen varianzas similares ya que el valor p_value fue superior a 0.05 y

no se pudo rechazar la hipótesis nula. Para el resto de las variables, las varianzas fueron distintas.

2.4.3. Análisis estadístico

Se empezará el análisis estadístico de forma introductoria con el análisis estadístico descriptivo, ya que no se hizo en los anteriores apartados. También se realizarán tres pruebas estadísticas para intentar resolver las cuestiones o problemas planteados inicialmente.

2.4.3.0. Análisis estadístico descriptivo

```
summary (df)
```

```
##  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    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
##  gruposcalidad
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.631
##  3rd Qu.:2.000
##  Max.   :3.000
```

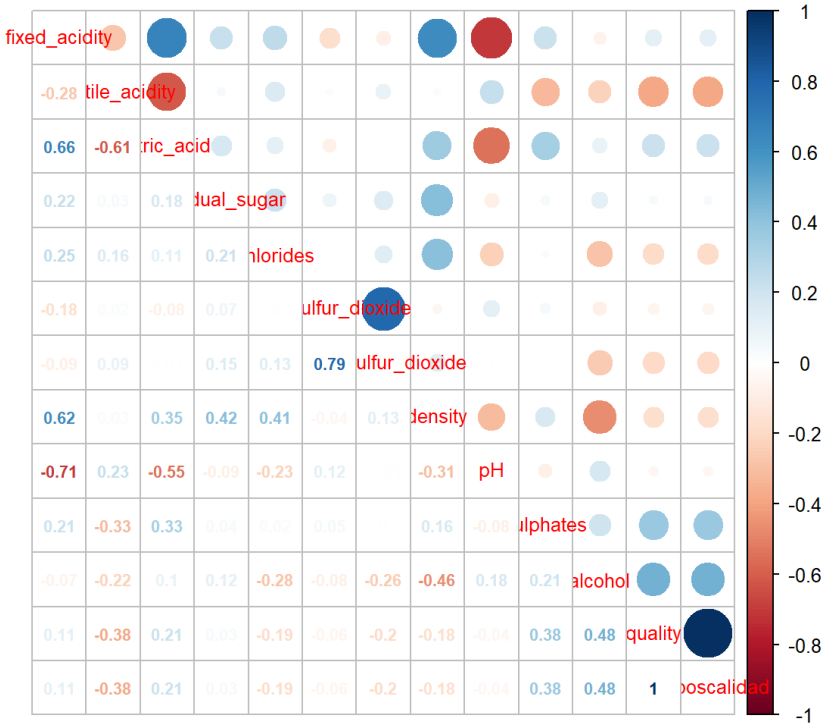
Como se puede observar en el análisis estadístico las variables 'residual_sugar', 'free_sulfur_dioxide', 'total_sulfur_dioxide' se confirma que tienen valores muy lejanos de los que podrían esperarse (sobre todo la variable total_sulfur_dioxide). Otro dato importante por analizar es la variable 'quality' en que se observa que la media de los

vinos se encuentra en 5.636. Esto nos da una idea que en gran parte los vinos que se analiza son de un rango medio.

2.4.3.1. Correlación entre variables

Se realizará la correlación de las distintas variables entre todas ellas, mediante el test de correlación **Spearman**, ya que las variables del estudio no siguen una distribución normal, en caso de que la hubieran seguido una distribución normal, se hubiera aplicado la correlación de **Pearson**. Para determinar la correlación se utilizará la función cor() con el método 'Spearman', y se mostrarán los resultados con una figura en círculos y sus valores. Cuando mayor sean los círculos, mayor será su correlación. Y según su color, la correlación será positiva si es azul, y negativa si es roja. Respeto a los valores numéricos de la correlación, cuando más cercanos a +1 o -1 mayor será su correlación.

```
corr.res<-cor(df, method="spearman")
corrplot.mixed(corr.res,upper="circle",number.cex=.7,tl.cex=.8)
```



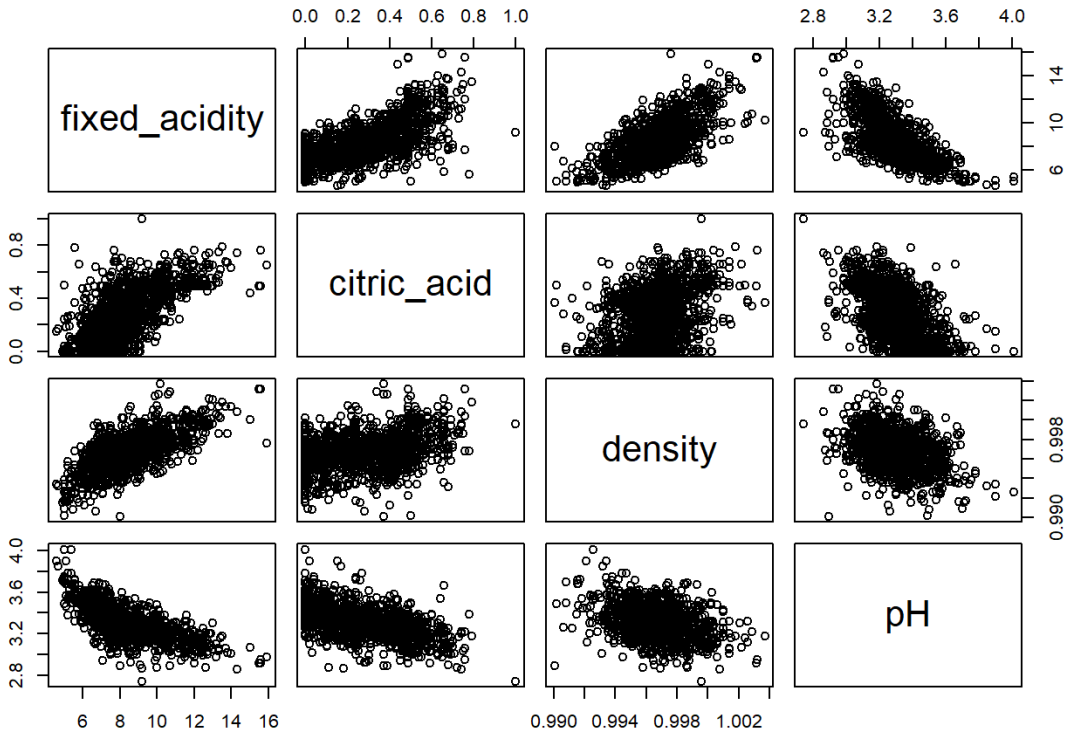
Como se puede observar la mayores correlaciones han sido para el conjunto de variables 'fixed_acidity'-'citric_acid' (correlación de 0.66), 'volatile_acidity'-'citric_acid'

(-0.61), 'fixed_acidity'-'density' (0.62), 'fixed_acidity'-'pH' (-0.71) y 'alcohol'-'density' (-0.46).

Para la variable de más interés 'quality', los valores de correlación han sido relativamente muy bajos, en que el coeficiente de dicha variable con la variable alcohol ha sido de 0.48, con 'volatile_acidity' -0.38 y con 'sulphates' de 0.38. Uno de los motivos por el cual la correlación de las distintas variables con la variable calidad tengan valores bajos, podría ser que 'quality' sea de tipo entero, y por tanto, su rango de valores dentro de un mismo grupo no sea muy elevado. Y en consecuencia no haya una nube de puntos muy definida.

A modo resumen se mostrarán la disposición de las muestras según las variables 'fixed_acidity', 'citric_acid', 'density' y 'pH', con ello se puede observar las tendencias positivas y negativas de las relaciones entre variables.

```
df2<-subset(df, select=c(1,3,8,9))
pairs(df2[, colnames(df2)])
```



2.4.3.2. Tests hipótesis

Se realizarán distintos tests de hipótesis con el objetivo de resolver distintas cuestiones que se han planteado.

¿La calidad del vino es superior si éste tiene menos contenido de ácidos volatiles?

Para resolver esta cuestión se hará un test de hipótesis de dos muestras donde se comparan las distintas medias de cada una de ellas. Las muestras que se escogerán son el contenido de acidez volátil en función de los grupos de calidad (gruposcalidad). Cabe recordar que el grupo de calidad =4 no tenía ninguna observación

```
va0<-df$volatile_acidity[df$gruposcalidad==0]
va1<-df$volatile_acidity[df$gruposcalidad==1]
va2<-df$volatile_acidity[df$gruposcalidad==2]
va3<-df$volatile_acidity[df$gruposcalidad==3]
```

Una vez realizado los distintos grupos se realizará un test de hipótesis, comparando una muestra de menor calidad, con la de máxima calidad (habrá un total de tres)

```
Contraste de hipótesis de dos muestras (ácidos volátiles según calidad)

            H0 : μ1 = μ2  (Hipótesis nula)

            H1 : μ1 > μ2  (Hipótesis alternativa)

Donde μ1 es la media de ácidos volátiles para vinos de menor calidad y μ2
es la media para vinos de mayor calidad (en todos los casos, ésta será si
empre la de grupo de calidad 3).
La hipótesis nula será que las dos medias sean iguales, en caso de no ace
ptar la hipótesis nula, la media de peor calidad será superior.
```

También se aprovechará para verificar el test de varianzas (homogeneidad)

```
#verificación del test de varianzas
var.test(va0, va3)

##
##  F test to compare two variances
```

```
##
## data:  va0 and va3
## F = 2.9261, num df = 62, denom df = 216, p-value = 8.893e-09
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  2.001734 4.475139
## sample estimates:
## ratio of variances
##           2.926065

#test hipótesis acidez volatil -- calidad
t.test(va0, va3, alternative="greater", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  va0 and va3
## t = 9.7293, df = 74.702, p-value = 3.215e-15
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2641236       Inf
## sample estimates:
## mean of x mean of y
## 0.7242063 0.4055300

t.test(va1, va3, alternative="greater", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  va1 and va3
## t = 14.668, df = 408.54, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.1522345       Inf
## sample estimates:
## mean of x mean of y
## 0.5770411 0.4055300

t.test(va2, va3, alternative="greater", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  va2 and va3
## t = 7.8433, df = 410.66, p-value = 1.914e-14
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.07262673        Inf
## sample estimates:
```

```
## mean of x mean of y
## 0.4974843 0.4055300
```

Como se pueden observar en los resultados, las medias de los distintos grupos analizados son distintas ya que el valor p-value del test ha sido en todos los casos inferior a alpha:0.05, por lo tanto, no se acepta la hipótesis nula y la media de ácidos volátiles del grupo de peor calidad será superior al de mayor calidad. También se confirma que las dos varianzas de las muestras no son iguales.

Estos resultados nos pueden permitir diferenciar las calidades según la cantidad de ácidos volátiles que tenga el vino, y, por lo tanto, podría ser una de las variables clave para la modelización.

¿La calidad del vino es superior si éste tiene más contenido de ácido cítrico?

Se procederá a realizar los grupos de ácido cítrico según la variable grupos calidad del vino, igual que se hizo en la pregunta anterior.

```
ca0<-df$citric_acid[df$gruposcalidad==0]
ca1<-df$citric_acid[df$gruposcalidad==1]
ca2<-df$citric_acid[df$gruposcalidad==2]
ca3<-df$citric_acid[df$gruposcalidad==3]

Contraste de hipótesis de dos muestras (ácido cítrico según calidad)

            H0 : μ1 = μ2  (Hipótesis nula)

            H1 : μ1 < μ2  (Hipótesis alternativa)

Donde μ1 es la media de ácido cítrico para vinos de menor calidad y μ2 es
la media de mayor calidad (en todos los casos, ésta será siempre la de ca
lidad 7).
La hipótesis nula será que las dos medias sean iguales, en caso de no ace
ptar la hipótesis nula, la media de peor calidad será inferior.

#test de varianzas
var.test(ca0, ca3)

##
##  F test to compare two variances
##
## data:  ca0 and ca3
```

```
## F = 1.1378, num df = 62, denom df = 216, p-value = 0.4981
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7783958 1.7402060
## sample estimates:
## ratio of variances
##           1.137832

#test hipótesis ácido cítrico -- calidad
t.test(ca0, ca3, alternative="less", var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  ca0 and ca3
## t = -7.1802, df = 278, p-value = 3.189e-12
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.1562227
## sample estimates:
## mean of x mean of y
## 0.1736508 0.3764977

t.test(ca1, ca3, alternative="less", var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  ca1 and ca3
## t = -9.2803, df = 896, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.1092478
## sample estimates:
## mean of x mean of y
## 0.2436858 0.3764977

t.test(ca2, ca3, alternative="less", var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  ca2 and ca3
## t = -6.7022, df = 853, p-value = 1.865e-11
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.0774477
## sample estimates:
## mean of x mean of y
## 0.2738245 0.3764977
```

En este caso sólo se hizo el test de hipótesis para los distintos grupos de calidad con el de mayor calidad (ca3). Se rechazo la hipótesis nula, ya que el valor de p-value fue inferior a 0.05, y se acepta que la media de ácido cítrico es inferior para los vinos de menor calidad respeto al de mayor calidad. También se verificó la no igualdad de varianzas entre los dos grupos.

¿La calidad del vino es superior si éste tiene más contenido de alcohol?

Se procederá a realizar los grupos de alcohol según los grupos de calidad del vino, igual que se hizo en la pregunta anterior.

```
a0<-df$alcohol[df$gruposcalidad==0]
a1<-df$alcohol[df$gruposcalidad==1]
a2<-df$alcohol[df$gruposcalidad==2]
a3<-df$alcohol[df$gruposcalidad==3]

Contraste de hipótesis de dos muestras (alcohol según calidad)

            H0 : μ1 = μ2  (Hipótesis nula)

            H1 : μ1 < μ2  (Hipótesis alternativa)

Donde μ1 es la media alcohol para vinos de menor calidad (en todos los ca
sos, ésta será siempre la de calidad 4) y μ2 es la media de mayor calidad
.
La hipótesis nula será que las dos medias sean iguales, en caso de no ace
ptar la hipótesis nula, la media de peor calidad será inferior.

#test de varianzas
var.test(a0, a3)

##
##  F test to compare two variances
##
## data:  a0 and a3
## F = 0.84617, num df = 62, denom df = 216, p-value = 0.4427
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5788706 1.2941411
## sample estimates:
## ratio of variances
##           0.846173
```

```
#test hipótesis alcohol -- calidad
t.test(a0, a1, alternative="less", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  a0 and a1
## t = 2.6552, df = 69.578, p-value = 0.9951
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.5146669
## sample estimates:
## mean of x mean of y
## 10.215873  9.899706

t.test(a0, a2, alternative="less", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  a0 and a2
## t = -3.3652, df = 78.906, p-value = 0.0005916
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.2090637
## sample estimates:
## mean of x mean of y
##  10.21587  10.62952

t.test(a0, a3, alternative="less", var.equal=FALSE)

##
##  Welch Two Sample t-test
##
## data:  a0 and a3
## t = -9.7131, df = 108.19, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.079757
## sample estimates:
## mean of x mean of y
##  10.21587  11.51805
```

En este caso, los distintos tests mostraron distintos resultados, ya que la comparación entre los grupos a0 y a1, se tuvo que aceptar la hipótesis nula ya que la media de alcohol fue en la calidad inferior (gruposcalidad 0) fue superior a la grupocalidad 1. En cambio, para el resto de tests, sí que se rechazó la hipótesis nula y se aceptó que el contenido de alcohol era inferior.

Por último y como objetivo final de la práctica, en los siguientes subapartados se realizarán distintos modelos de regresión lineal, regresión polinómica, random Forest, … que permitan predecir la calidad de un vino a partir de los distintos atributos analizados (todos ellos son regresores cuantitativos).

2.4.3.3. Modelo predictivo regresión lineal

Se empezarán con la realización de distintos modelos de regresión lineal, para ello se crearán un grupo de datos de entreno y otros grupos que se utilizará como test. El método que se utilizará para la partición de datos es el método de exclusión, donde los datos se dividen aleatoriamente en dos conjuntos independientes, el de entrenamiento y el de test. Típicamente, dos tercios de los datos se asignan al conjunto de entrenamiento, y el tercio restante se reserva para testear el modelo (L. Subirats et al(2019)). En nuestro caso, la muestra de entreno va a presentar un 70% del dataset original.

```
# División de los datos en train y test, método exclusión

set.seed(250)
id_train <- sample(1:nrow(df), size = 0.7*nrow(df), replace = FALSE)

df.train <- df[id_train, ]
df.test  <- df[-id_train, ]
```

También con el objetivo de ver como varia la métrica de rendimiento R^2 se generarán cuatro modelos con distintas variables y todos los datos del dataset y dos modelos con los datos de entreno (df.train).

```
#Generación de modelos lineales
modelo1<- lm(gruposcalidad ~ alcohol + volatile_acidity + sulphates, data
= df)
modelo2<- lm(gruposcalidad ~ volatile_acidity + sulphates + total_sulfur_
dioxide + alcohol, data = df)
modelo3<- lm(gruposcalidad ~ volatile_acidity + sulphates + chlorides + t
otal_sulfur_dioxide + alcohol + pH, data = df)
modelo4<- lm(gruposcalidad~. -quality, data=df)

#Modelos con entreno
modelo5<- lm(gruposcalidad ~ volatile_acidity + sulphates + chlorides + t
```

```
otal_sulfur_dioxide + alcohol + pH, data = df.train)
modelo6<- lm(gruposcalidad~. -quality, data=df.train)
```

Las distintas métricas R^2 obtenidas de los modelos se muestran en la siguiente tabla:

```
# Tabla con los coeficientes de determinación de cada modelo
tab.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
                             2, summary(modelo2)$r.squared,
                             3, summary(modelo3)$r.squared,
                             4, summary(modelo4)$r.squared,
                             5, summary(modelo5)$r.squared,
                             6, summary(modelo6)$r.squared),
                             ncol = 2, byrow = TRUE)
colnames(tab.coeficientes) <- c("Modelo", "R^2")
tab.coeficientes
```

```
##      Modelo       R^2
## [1,]      1 0.3351411
## [2,]      2 0.3449959
## [3,]      3 0.3567689
## [4,]      4 0.3609884
## [5,]      5 0.3526147
## [6,]      6 0.3573692
```

El coeficiente R^2 nos mide el grado que los distintos valores quedan ajustados en el modelo. Cuando este valor es cercano a 1, el modelo es considerado bueno, en cambio cuando el valor es más cercano a 0, el modelo es considerado como malo. Como se puede observar en los distintos modelos de regresión lineal, el valor R^2 es muy bajo, considerando que los modelos generados son de baja calidad o malos. También se observa que el hecho de introducir más variables en el modelo, el ajuste (R^2) ha augmentado muy poco. No siempre con el augmento de variables en el modelo genera una mejor respuesta. Respeto a los modelos creados a partir de los datos de entreno (df.train) no hubo mejora con el coeficiente R^2.

Para terminar la regresión lineal se hará la predicción de distintos valores considerados para vino buenos y malos, así como los datos de testeo (df.test)

```
x1<-data.frame(volatile_acidity=0.25, sulphates=0.45, chlorides=0.08, tot
al_sulfur_dioxide=0.4, alcohol=14.5, pH=2.9)
```

```
x2<-data.frame(volatile_acidity=1.4, sulphates=2.01, chlorides=0.45, tota
l_sulfur_dioxide=150, alcohol=10, pH=3.7)

predict(modelo3, x1)

##        1
## 3.127757

predict(modelo3, x2)

##         1
## 0.7484777

a<-predict(modelo6, df.test)
head(a)

##        2        7       12       13       14       23
## 1.134914 1.138168 1.631119 1.192519 1.867978 1.700973

predict(modelo5, x1)

##        1
## 3.170653

predict(modelo5, x2)

##         1
## 0.6780777
```

Para intentar mejorar el modelo se crearán otros modelos mediante la regresión polinómica y el randonForest.

2.4.3.4. Modelo predictivo regresión polinómica

Previo al siguiente modelo predictivo de regresión polinómica, se hicieron otros intentando observar cual podría ser el mejor. Los distintos R^2 permanecían bajos, así que a continuación solo se mostrará un único modelo derivado del modelo 5 donde las variables 'sulphates' y 'pH' son linéales y cuadráticas (se puede observar en el sumario del modelo).

```
modelo_pol1<- lm(gruposcalidad ~ volatile_acidity + poly(sulphates,2) + a
lcohol + chlorides + total_sulfur_dioxide +  poly(pH,2) , data = df.train
)
```

```
summary(modelo_pol1)

##
## Call:
## lm(formula = gruposcalidad ~ volatile_acidity + poly(sulphates,
##     2) + alcohol + chlorides + total_sulfur_dioxide + poly(pH,
##     2), data = df.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24561 -0.37732 -0.05595  0.46219  1.78799
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.6234579  0.2337113  -2.668  0.00775 **
## volatile_acidity      -0.9149477  0.1138280  -8.038 2.32e-15 ***
## poly(sulphates, 2)1    4.0116149  0.6958551   5.765 1.06e-08 ***
## poly(sulphates, 2)2   -3.3238487  0.6374752  -5.214 2.20e-07 ***
## alcohol                0.2803633  0.0190813  14.693  < 2e-16 ***
## chlorides             -1.1857438  0.4759496  -2.491  0.01287 *
## total_sulfur_dioxide -0.0018790  0.0005655  -3.323  0.00092 ***
## poly(pH, 2)1          -2.3481795  0.6715690  -3.497  0.00049 ***
## poly(pH, 2)2          -1.4751283  0.6288694  -2.346  0.01917 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.606 on 1110 degrees of freedom
## Multiple R-squared:  0.3727, Adjusted R-squared:  0.3681
## F-statistic: 82.42 on 8 and 1110 DF,  p-value: < 2.2e-16
```

Con este modelo generado (modelo_pol1) y el modelo 5 se van a hacer comparaciones sobre distintas métricas de evaluación (R^2, MSE, RMSE y MAE). Se generará una función para que nos dé el resultado de cada una de ellas y se aplicará para los dos modelos.

```
indicator <- function(model, y_pred, y_true) {
     adj.r.sq <- summary(model)$adj.r.squared
     mse <- MSE(y_pred, y_true)
     rmse <- RMSE(y_pred, y_true)
     mae <- MAE(y_pred, y_true)
     print(paste0("Adjusted R-squared: ", round(adj.r.sq, 4)))
     print(paste0("MSE: ", round(mse, 4)))
     print(paste0("RMSE: ", round(rmse, 4)))
     print(paste0("MAE: ", round(mae, 4)))
}
#Métricas para modelo5
```

```
indicator(model = modelo5, y_pred = modelo5$fitted.values, y_true = df.tr
ain$gruposcalidad)
```

```
## [1] "Adjusted R-squared: 0.3491"
## [1] "MSE: 0.3759"
## [1] "RMSE: 0.6131"
## [1] "MAE: 0.4908"
```

```
#Métricas para modelo_pol1
indicator(model = modelo_pol1, y_pred = modelo_pol1$fitted.values, y_true
= df.train$gruposcalidad)
```

```
## [1] "Adjusted R-squared: 0.3681"
## [1] "MSE: 0.3643"
## [1] "RMSE: 0.6036"
## [1] "MAE: 0.4832"
```

Como se puede observar, los resultados obtenido para el modelo siguen siendo de baja calidad o malos. Para intentar mejorar el modelo, se creará un modelo RandomForest.

2.4.3.5. Modelo RandomForest

Para poder realizar un modelo RandomForest, será necesario pasar la variable dependiente a tipo 'factor'. Para ello se convertirá los datos de entreno df.train.

```
#Conversión variable gruposcalidad a tipo factor
df.train$gruposcalidad<-as.factor(df.train$gruposcalidad)
```

Generamos modelo randomForest con todas las variables menos quality y utilizando los datos de entreno df.train. También se mostrarán los resultados obtenidos del modelo y se calculará la precisión de éste a partir de su matriz de confusión.

```
# Create random forest for regression
quality.rf <- randomForest(gruposcalidad~. -quality, data = df.train)
# Resultados RandomForest
quality.rf
```

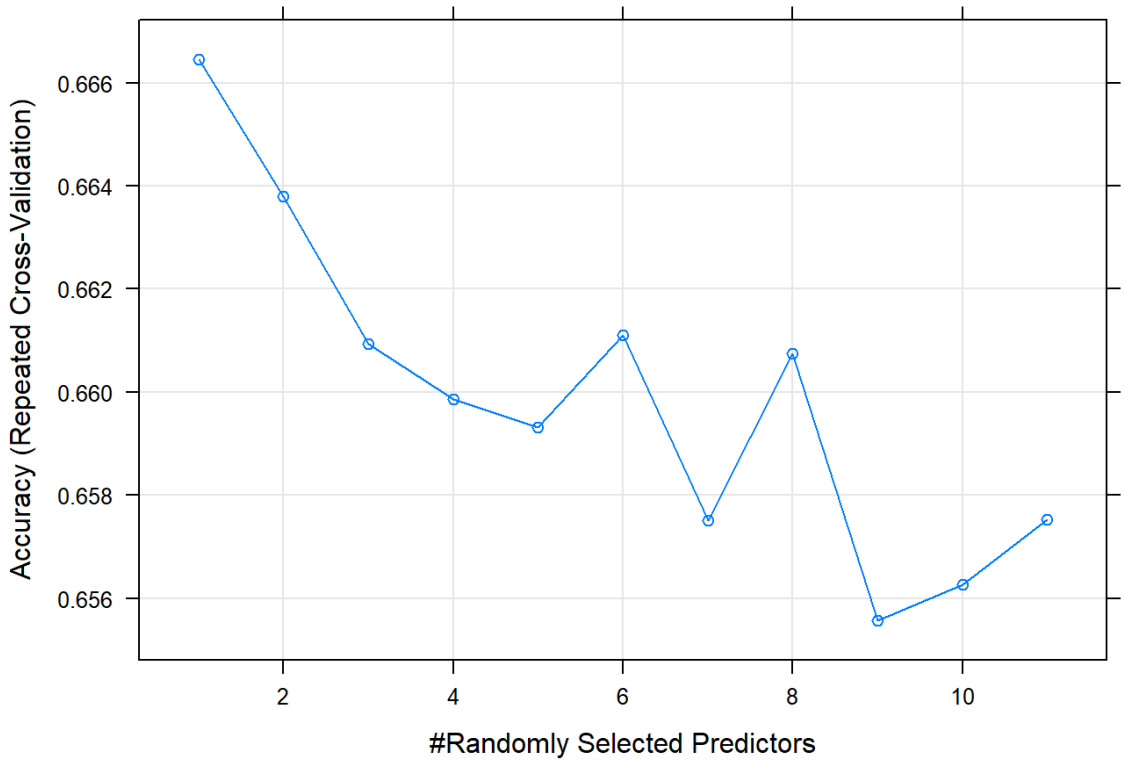
```
##
## Call:
##  randomForest(formula = gruposcalidad ~ . - quality, data = df.train)
##                Type of random forest: classification
##                      Number of trees: 500
```

```
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 32.08%
## Confusion matrix:
##   0   1   2  3 class.error
## 0 2  29  16  1   0.9583333
## 1 0 374 100  1   0.2126316
## 2 1 100 315 35   0.3015521
## 3 0   9  67 69   0.5241379

#Precisión train
(2+374+315+69)/nrow(df.train)

## [1] 0.6791778
```

Con este modelo se ha calculado el OOB (out of bag error) que es un método de la medida del error de predicción para los modelos RandomForest. En este caso ha sido del 32.08%. También se calculó la precisión del modelo con los datos de entreno que fue del 68% aprox.

Para intentar mejorar este modelo RandomForest, se realizará un entrenamiento mediante la librería 'caret' train function. Se utilizará un método de validación cruzada (repeatedcv) de 5 veces y repetido 5 veces. El método 'repeatedcv' se utiliza para especificar la validación cruzada repetida de K veces (y el argumento repeticiones controla el número de repeticiones).

```
df$gruposcalidad<-as.factor(df$gruposcalidad)
#Entreno y generación de modelo
t.ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 5)
rf.grid <- expand.grid(mtry = 1:11)
#Creación modelo a partir de los datos df.train
rf.train <- train(gruposcalidad ~ . -quality, data = df.train, method = "
rf",
                  trControl = t.ctrl, tuneGrid = rf.grid,
                  preProcess = c("center", "scale"))
plot(rf.train)
```



Como se puede observar en el gráfico, el modelo llega a obtener una precisión entre el 65 y el 67% según la cantidad de distintos predictores (extraoficialmente, se probó este modelo con todos los datos y aplicaron cross validation, y se obtuvo una precisión superior al 70%).

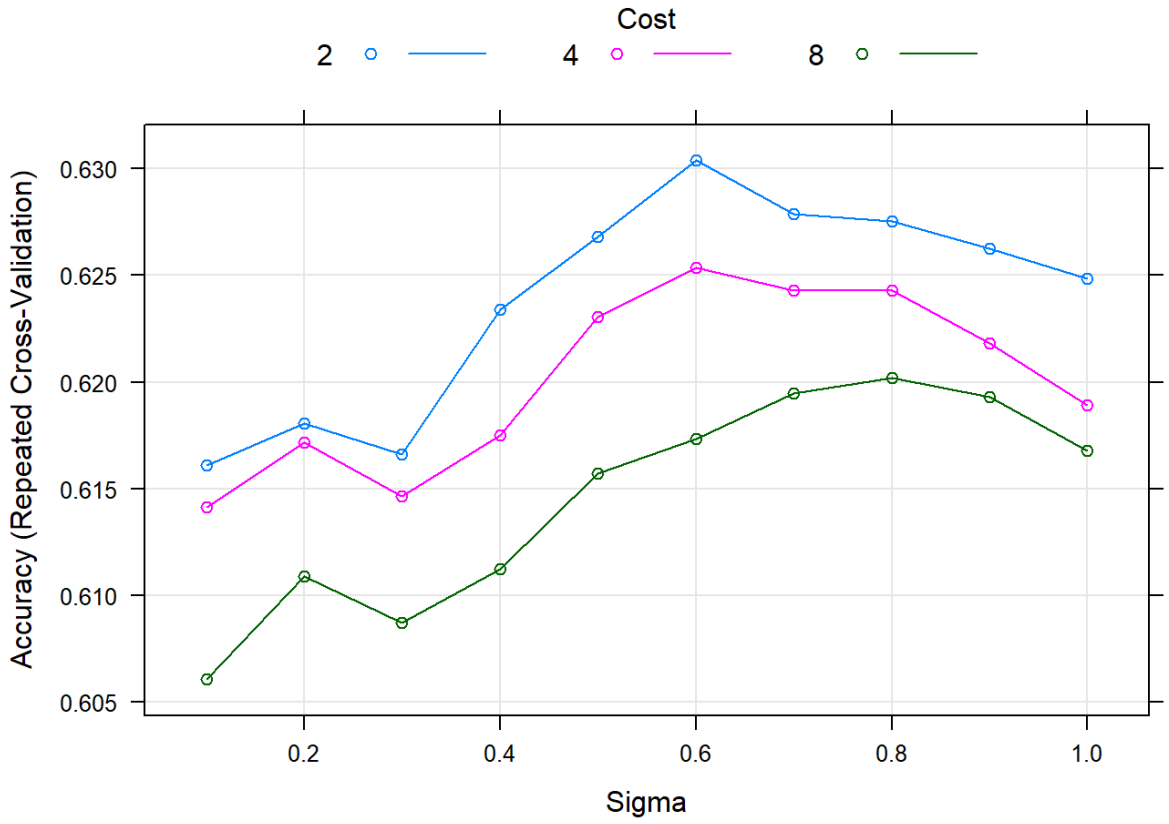
2.4.3.7. Support vector machine (regresion)

De forma adicional e introductoria se realizará otro modelo, Support Vector Regresion (o también Suport Vector Machine, radial Kernel), para comparar con el resto de modelos. Los SVM son modelos de clasificación o de análisis de la regresión. Una máquina de vectores coge como entrada un set de datos y predice, por cada una de estas entradas a cuál de las dos posibles clases pertenece. Mediante el entrenamiento con datos de entrada previamente clasificadas, se establece un modelo que separa las dos clases entrantes. Este modelo N-dimensional establece una frontera entre las dos tipologías establecidas, esta se sitúa en el punto en el que la diferencia entre clases sea

lo más grande posible y el margen de error sea cero (dataset separable) o mínimo (dataset no separable). Se denominan vectores de apoyo a los puntos que conforman las dos líneas paralelas al hiperplano, siendo esta distancia la mayor posible (margen).

Este modelo, igual que el segundo modelo RandomForest, también se utilizará un entrenamiento con validación cruzada. Se tendrán en cuenta el método para escoger los Kernels (núcleos) que fue el método radial basis function.

```
svm.grid <- expand.grid(C = 2^(1:3), sigma = seq(0.1, 1, length = 10))
svm.mod <- train(gruposcalidad ~ . -quality, data = df.train, method = "s
vmRadial",
                 trControl = t.ctrl, tuneGrid = svm.grid,
                 preProcess = c("center", "scale"))
plot(svm.mod)
```



Como se puede ver en el gráfico anterior, la mejor precisión del modelo Support Vector Machine fue del 63% aproximadamente, con un coste de 2 (dos grupos o hiperplanos) y sigma del 0.6.

2.4.3.8. Comparación de modelos

Para finalizar la práctica se realizará una comparación de los distintos modelos: modelo 6 (modelo regresión lineal con todas la variables y datos entreno (método exclusión)), quality.rf (random forest con método de entreno exclusión) y rf.train (random forest con método de entreno validación cruzada a partir de los datos entreno anteriores). Para ello, se hará una predicción con los datos reservados para el test (df.test) y después se generará la matriz de confusión con los datos reales por medio de la función confusionMatrix(). La función confusionMatrix() estima los parámetres de precisión, sensibilidad y especificidad del modelo entre otros. (Nota: la predicción para el modelo6 (a) se redondeó y se transformó como factor)

```
#Matriz de confusión para modelo6 regresión lineal
confusionMatrix(as.factor(round(a,0)), as.factor(df.test$gruposcalidad))
```

```
## Warning in levels(reference) != levels(data): longer object length is
not a
## multiple of shorter object length
```

```
## Warning in confusionMatrix.default(as.factor(round(a, 0)),
## as.factor(df.test$gruposcalidad)): Levels are not in the same order fo
r
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2   3
##          0   0   0   0   0
##          1   8 144  44   2
##          2   7  62 138  57
##          3   0   0   5  13
##
## Overall Statistics
##
##                Accuracy : 0.6146
##                  95% CI : (0.5694, 0.6583)
##     No Information Rate : 0.4292
##     P-Value [Acc > NIR] : 2.489e-16
##
##                   Kappa : 0.3609
##
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3
## Sensitivity           0.00000   0.6990   0.7380  0.18056
## Specificity           1.00000   0.8029   0.5700  0.98775
## Pos Pred Value            NaN   0.7273   0.5227  0.72222
## Neg Pred Value        0.96875   0.7801   0.7731  0.87229
## Prevalence            0.03125   0.4292   0.3896  0.15000
## Detection Rate        0.00000   0.3000   0.2875  0.02708
## Detection Prevalence  0.00000   0.4125   0.5500  0.03750
## Balanced Accuracy     0.50000   0.7510   0.6540  0.58415
```

```
#MAtriz de confusión para modelo randomForest con método entrenamiento de exclusión
proba<-predict(quality.rf, newdata = df.test)
confusionMatrix(as.factor(proba), as.factor(df.test$gruposcalidad))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2   3
##          0   1   3   0   0
##          1  10 157  43   2
##          2   4  42 131  32
##          3   0   4  13  38
##
## Overall Statistics
##
##                Accuracy : 0.6812
##                  95% CI : (0.6375, 0.7228)
##     No Information Rate : 0.4292
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4887
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3
## Sensitivity          0.066667   0.7621   0.7005  0.52778
## Specificity          0.993548   0.7993   0.7338  0.95833
## Pos Pred Value       0.250000   0.7406   0.6268  0.69091
## Neg Pred Value       0.970588   0.8172   0.7934  0.92000
## Prevalence           0.031250   0.4292   0.3896  0.15000
## Detection Rate       0.002083   0.3271   0.2729  0.07917
## Detection Prevalence 0.008333   0.4417   0.4354  0.11458
## Balanced Accuracy    0.530108   0.7807   0.7172  0.74306
```

```
#MAtriz de confusión para modelo randomForest con método entrenamiento validación cruzada
rf.pred2 <- predict(rf.train, df.test)
confusionMatrix(as.factor(rf.pred2), as.factor(df.test$gruposcalidad))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2   3
##          0   1   2   0   0
##          1  10 160  46   2
##          2   4  43 135  39
##          3   0   1   6  31
##
## Overall Statistics
##
##                Accuracy : 0.6812
##                  95% CI : (0.6375, 0.7228)
##     No Information Rate : 0.4292
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4806
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3
## Sensitivity          0.066667   0.7767   0.7219  0.43056
## Specificity          0.995699   0.7883   0.7065  0.98284
## Pos Pred Value       0.333333   0.7339   0.6109  0.81579
## Neg Pred Value       0.970650   0.8244   0.7992  0.90724
## Prevalence           0.031250   0.4292   0.3896  0.15000
## Detection Rate       0.002083   0.3333   0.2812  0.06458
## Detection Prevalence 0.006250   0.4542   0.4604  0.07917
## Balanced Accuracy    0.531183   0.7825   0.7142  0.70670

#Matriz confusión modelo SVM con método entrenamiento validación cruzada
svm.test <- predict(svm.mod, df.test)
confusionMatrix(as.factor(svm.test), as.factor(df.test$gruposcalidad))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2   3
##          0   1   2   0   0
##          1  11 156  54  11
##          2   3  44 121  32
##          3   0   4  12  29
##
```

```
## Overall Statistics
##
##                Accuracy : 0.6396
##                  95% CI : (0.5948, 0.6826)
##     No Information Rate : 0.4292
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4149
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3
## Sensitivity          0.066667   0.7573   0.6471  0.40278
## Specificity          0.995699   0.7226   0.7304  0.96078
## Pos Pred Value       0.333333   0.6724   0.6050  0.64444
## Neg Pred Value       0.970650   0.7984   0.7643  0.90115
## Prevalence           0.031250   0.4292   0.3896  0.15000
## Detection Rate       0.002083   0.3250   0.2521  0.06042
## Detection Prevalence 0.006250   0.4833   0.4167  0.09375
## Balanced Accuracy    0.531183   0.7400   0.6887  0.68178
```

Con estos resultados se pueden comparar los distintos modelos en los tests posteriores. La mayor precisión fueron por los dos modelos randomForest, aproximadamente del 68%, seguido del SVM 64% y por último el modelo de regresión lineal (61%, donde se tuvieron que redondear los valores predichos, si se hubiese truncado la precisión hubiera sido peor). También se obtuvo la sensibilidad y especificidad de estos modelos para los distintos grupos, la mayor sensibilidad (verdaderos positivos) fue detectada para la clase 1 (aprox.77.7% rf.pred) y la mayor especificidad (Verdaderos negativos) para la clase 0. Las peores sensibilidades fueron detectadas para la clase 3

3. Conclusión

En la presente práctica de limpieza y análisis de datos se escogió un dataSet referente a la calidad de un vino (distintas variables físico-químicas) con el objetivo de responder distintas preguntas y crear un modelo predictivo de la calidad del vino.

- La primera parte de la práctica se realizó una limpieza de los datos, donde todas las variables eran del tipo numérico y todas ellas mostraban valores extremos outliers. En nuestro caso, estos valores extremos o outliers no se eliminaron ya que podrían darse en situaciones reales.

- En la segunda parte, análisis de datos, se analizó la normalidad de los datos, donde se observó que ninguna de las variables seguía una función normal, y las variables pH y Chlorides mostraban igualdad de varianzas respeto la variable quality (el resto de las variables sus varianzas eran distintas).

- En el análisis estadístico, se realizó un análisis descriptivo de las variables, se determinó la correlación que había entre las distintas variables y se realizaron distintos test de hipótesis con el objetivo de resolver distintas preguntas. De forma general se observó poca correlación entre variables, las mayores fueron para los conjuntos de variables 'fixed_acidity'-'citric_acid' (correlación de 0.66), 'volatile_acidity'-'citric_acid' (-0.61), 'fixed_acidity'-'density' (0.62), 'fixed_acidity'-'pH' (-0.71) y 'alcohol'-'density' (-0.46). También se comprobó que a una mayor calidad del vino la presencia de ácidos volátiles será inferior y también tendrán mayor cantidad de ácido cítrico. Respeto a la cantidad de alcohol, los distintos test de hipótesis realizados mostraron distintos resultados y no se pudo verificar de forma general que los vinos de mayor calidad tuviesen mayor contenido de alcohol.

- Por último, se realizaron distintos modelos predictivos para la variable calidad. Se pudo comparar los métodos de regresión lineal múltiple, regresión polinómica, random Forest y se empezó a estudiar un modelo vector support machine o regression. También se utilizaron distintas formas para la partición de los datos (exclusión y validación cruzada) y se utilizaron distintas métricas de rendimiento para la comparación de los modelos. De forma general y en este caso, se concluyó que los modelos RandomForest fueron los que mayor precisión tenían para la predicción de la calidad de los vinos.

4. Bibliografía

J. Gibergans (2018) Regresión lineal simple. Universitat Oberta de Catalunya

J. Gibergans (2018) Regresión lineal múltiple. Universitat Oberta de Catalunya

T. Gutiérrez (2017) Práctica 2: Limpieza y validación de los datos

D. Liviano y M. Pujol (2019) Análisis de datos y estadística con R y R-commander. Universitat Oberta de Catalunya.

D. Liviano y M. Pujol (2019) Modelos de regresión y análisis multivarible con R-commander. Universitat Oberta de Catalunya.

L. Subirats, D.O. Oswaldo, M. Calvo (2019) Introducción a la limpieza y análisis de los datos. Universitat Oberta de Catalunya

Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.

A. Sethi, (2020) Support Vector Regression Tutorial for Machine Learning. URL: https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/

5. Signatura documento

Signatura de los participantes en cada apartado de la práctica

Contribuciones	**Firma**
Investigación previa	J.A.Ubieto, J. Lladó
Redacción de las respuestas	J.A.Ubieto, J. Lladó
Desarrollo código	J.A.Ubieto, J. Lladó