Tipología y cliclo de vida de los datos: PRA2 - Limpieza y análisis de datos

Autor: Diego Martín Montoro, Javier Hernández Hernández

Diciembre 2020


```
library(ggplot2)
library(corrplot)
library(grid)
library(gridExtra)
library(reshape2)
library(stringr)
library(matrixStats)
library(ggmosaic)
library(tidyverse)
```

1 Cuestiones

2 Introducción del conjunto de datos.

El dataset elegido es el siguiente: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

El dataset está conformado de datos de vinos tintos. Tenemos diversa información y un campo objetivo por cada vino que representa la calidad de dicho vino. Las variables son las siguientes:

1. **fixed acidity:** acidez del vino.
2. **volatile acidity:** La acidez volátil es una parte de la acidez total de un vino, formada por los ácidos primarios que ya están presentes en el mosto de uva (málico y tartárico) y los secundarios que son los generados durante los procesos de fermentación (acético, succínico, málico, ...).
3. **citric acid:** cantidad de ácido citrico que contiene el vino.
4. **residual sugar:** cantidad de azúcar que no ha sido fermentada por las levaduras que contiene el vino.
5. **chlorides:** cantidad de cloruros que contiene el vino.
6. **free sulfur dioxide:** corresponde al gas sulfuroso disuelto en el líquido.
7. **total sulfur dioxide:** corresponde al gas sulfuroso disuelto en el líquido combinado con diversas sustancias orgánicas presentes en el mosto o en el vino.
8. **density:** densidad del líquido.
9. **pH:** corresponde con acidez del vino.
10. **sulphates:** sulfitos del vino.
11. **alcohol:** cantidad de alcohol que contiene el vino.
12. **quality:** calidad asignada al vino. Valores de 0 a 10.

La **motivación** para usar este conjunto de datos reside en la cantidad de posibilidades que ofrece a la industria del vino, entre las que se encuentran:

- automatización del proceso de clasificación del vino.
- estimación automática de la calidad del vino.
- estudio de nuevas posibles jerarquías de vinos desconocidas hasta ahora y las propiedades que las definen.
- automatización del proceso de control de parámetros en la cadena de producción del vino.
- Identificación de nuevas configuraciones paramétricas beneficiosas.

El objetivo de esta asignación será obtener un conjunto de datos de vinos tintos que permita estudiar los parámetros que aseguran un vino de calidad, así como brindar una base de conocimiento para posibles automatizaciones futuras del proceso de validación de los vinos.

```
x <- read.csv('winequality-red.csv',sep=",", stringsAsFactors = TRUE)
summary(x)
```

```
##  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
```

```
str(x)
```

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

```
colnames(x) <- c("fixed_acidity","volatile_acidity","citric_acid","residual_sugar","chlorides","free_sulfur_dioxide","total_sulfur_dioxide","density","pH","sulphates","alcohol","quality")
head(x)
```

```
##   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
```

3 Preprocesado de los datos.

A lo largo de los siguientes apartados vamos a realizar un estudio y tratamiento de los datos con el objetivo de llevarlos a un estado adecuado para las tareas tanto de visualización como de diseño de modelos. Para ello comenzaremos con un estudio de outliers, proseguiremos con un tratamiento de los valores ausentes, y como punto final normalización de columnas numéricas por un lado y discretización de las variables pertinentes por otro.

3.1 Estudio de outliers

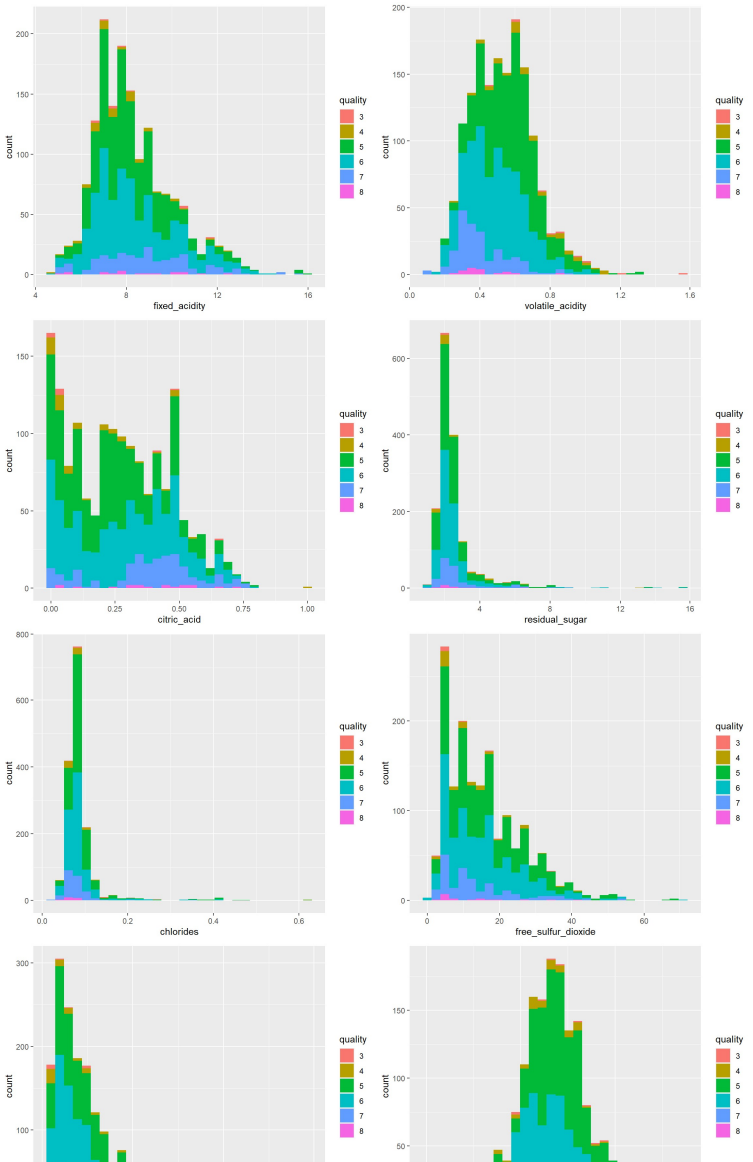
El estudio de outliers es muy importante porque nos permite diferenciar qué datos son valores excepcionales (de gran interés) y qué datos son producto de un error y por tanto no deben de tenerse en cuenta. Si entrenásemos un modelo con estos últimos, el modelo “aprendería” de esa información errónea creando una solución totalmente inválida. Es por esto que comienzo el procedimiento de preprocesado haciendo un estudio de outliers.

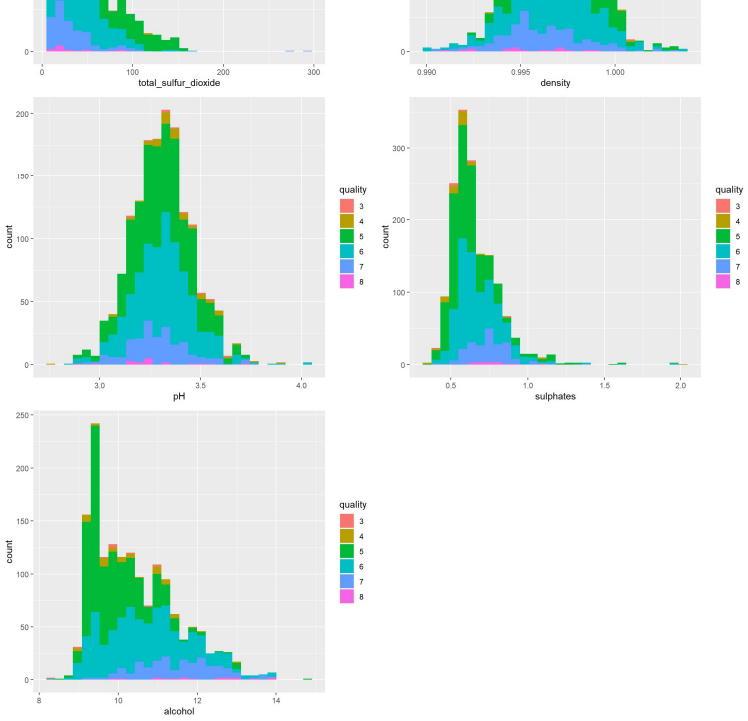
Sabemos que en este contexto valores outliers pueden ser claros indicadores de la alta/baja calidad del vino. Es por eso que antes de determinar que tipo de acciones se tomarán sobre estos comprobaremos si ocasionan algún tipo de efecto sobre la calidad del vino. Los que afecten a dicha variable no serán tratados para preservar dicha información.

Para determinar esto, lo que haremos es generar histogramas superpuestos por calidad, de manera que para cada rango de una distribución podamos observar la cantidad de vinos de cada calidad que hay en ella.

```
x$quality <- as.factor(x$quality)

plots <- list()

p1 <- ggplot(x, aes(x = fixed_acidity,fill = quality)) + geom_histogram()
p2 <- ggplot(x, aes(x = volatile_acidity,fill = quality)) + geom_histogram()
p3 <- ggplot(x, aes(x = citric_acid, fill = quality)) + geom_histogram()
p4 <- ggplot(x, aes(x = residual_sugar, fill = quality)) + geom_histogram()
p5 <- ggplot(x, aes(x = chlorides, fill = quality)) + geom_histogram()
p6 <- ggplot(x, aes(x = free_sulfur_dioxide, fill = quality)) + geom_histogram()
p7 <- ggplot(x, aes(x = total_sulfur_dioxide, fill = quality)) + geom_histogram()
p8 <- ggplot(x, aes(x = density, fill = quality)) + geom_histogram()
p9 <- ggplot(x, aes(x = pH, fill = quality)) + geom_histogram()
p10 <- ggplot(x, aes(x = sulphates, fill = quality)) + geom_histogram()
p11 <- ggplot(x, aes(x = alcohol, fill = quality)) + geom_histogram()

grid.arrange(arrangeGrob(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11, ncol = 2, nrow = 6))
```

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






```
x$quality <- as.numeric(as.character(x$quality))
```

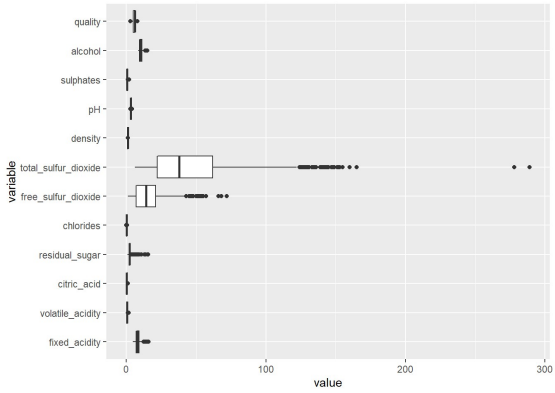
De casi todas las gráficas vemos que hay presencia de valores muy distintos al resto de valores de las distribuciones con distintos niveles de calidad, estos son tan pocos (6 o 7 en algunos casos) que no proporcionan evidencias suficientes para afirmar que valores inusuales puedan causar picos en la calidad, luego podemos proceder al tratamiento habitual de outliers sin miedo a desperdiciar información.

Las medidas de las que disponemos se consiguen tras promediar distintas mediciones sobre una gran variedad de muestras de cierta cantidad de líquido de cada vino. Al desconocer la cantidad de vino usada para cada muestra no podemos establecer mínimos y máximos "legales" para los contenidos de manera que no queda otra que confiar en las herramientas que brinda la estadística. Si en vez de cantidades tuviéramos porcentajes no habría problemas, pero no es el caso.

Procederemos con la determinación de fronteras para cada distribución basándonos en la regla 68-95-99.7 bajo la suposición de normalidad el 99,7% de los datos de una distribución se encuentran en el intervalo definido por $[\mu - 3\sigma , \mu + 3\sigma]$. (Pukelsheim, 1994)

Todo valor situado fuera de estas fronteras será considerado outlier e imputado (por ahora) con valor NA.

```
ggplot(melt(x), aes(x=variable, y=value)) + geom_boxplot() + coord_flip()
```



Como vemos hay presencia de datos extraños, eliminemos aquellos que se hallen fuera de las fronteras establecidas (demasiado extraños como para ser legítimos).
El siguiente bucle itera todas las columnas menos la de calidad (puesto que esta tiene un rango cerrado entre 0 y 10 donde no hay lugar al error) calculando dicho intervalo y aplicando una imputación de valor NA en todos los registros cuyo valor para dicha columna quede fuera del rango calculado.

```
for (col in head(colnames(x),-1) )
{
    media <- mean(x[,col])
    desv <- sd(x[,col])
    x[ (x[,col] < media - 3*desv) | (x[,col] > media + 3*desv) ,col]  <- NA
}
```

Con esto tenemos realizada la detección/tratamiento de outliers, pero siendo poco agresivos ya que en la minería de datos suele interesar la presencia de outliers siempre y cuando estos no generen demasiado caos en los datos y no sean valores descabellados.

3.2 Imputación y tratamiendo de valores NA

Una vez llegados a este punto, además de los valores NA presentes en los datos desde las fuentes de datos de origen, sabemos que durante el procedimiento del tratamiento de outliers se ha agravado el problema como podemos ver:

```
  colSums(is.na(x))
```

```
##        fixed_acidity     volatile_acidity          citric_acid
##                   12                   10                    1
##       residual_sugar            chlorides  free_sulfur_dioxide
##                   30                   31                   22
## total_sulfur_dioxide              density                   pH
##                   15                   18                    8
##            sulphates              alcohol              quality
##                   27                    8                    0
```

Es por esto que ahora debemos solucionar dicho problema comenzando por elegir qué valor será imputado en todos aquellos huecos que presentan los datos. El valor más adecuado es la mediana, puesto que es un estimador más robusto que la media. La media es insesgado pero su varianza es tan alta que no resulta un estimador confiable, la mediana por el contrario es sesgada pero al tener una varianza baja nos permite confiar en que su valor no variará mucho de una muestra a otra de manera que brinda cierta seguridad a la hora de hacer afirmaciones.
Expliquemos esto en un ejemplo trivial: es mejor tener una escopeta que tenga cierta desviación pero que esta sea fija de manera que todos los tiros tenga la "misma calidad", que usar una escopeta cuya desviación cambie haciendo que un tiro salga muy bien y otro catastróficamente mal.

Se sigue el código que implementa la imputación de valores:

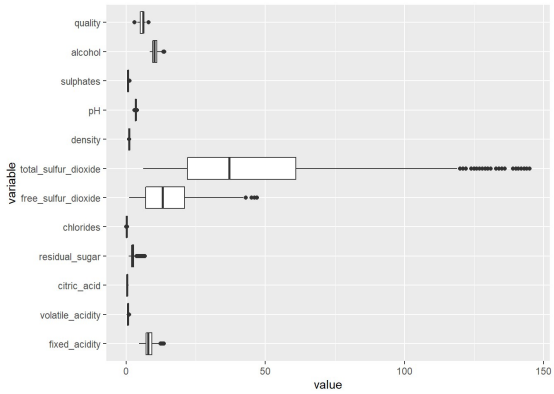
```
for ( col in head(colnames(x),-1) ) x[ is.na(x[,col]) ,col] <- median(x[,col],na.rm = TRUE)
colSums(is.na(x))
```

```
##        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
```

Ahora si, veamos el resultado final de los procesos de limpiado de outliers e imputación de valores:

```
ggplot(melt(x), aes(x=variable, y=value)) + geom_boxplot() + coord_flip()
```



Como vemos la cantidad de outliers detectados por el diagrama de cajas y bigotes ha sido reducido, aún no se aprecian los efectos de la imputación de valores pero esto se debe a la diferencia entre los rangos de las distintas variables. Esto será solucionado en el siguiente paso y podremos observar bien el efecto de la imputación de valores tras la normalización de las variables numéricas.

3.3 Normalización de variables

La fase de normalización de variables numéricas es muy importante puesto que la mayoría de algoritmos de aprendizaje computacional, minería de datos y reconocimiento de patrones basan su entrenamiento en el mismo principio o fundamento matemático: combinación lineal de variables.

Una combinación lineal de variables es una expresión matemática que consiste en:

$$y = \alpha_1 * x_1 + \alpha_2 * x_2 + \ \ldots \ + \alpha_{n-1} * x_{n-1} + \alpha_n * x_n$$

Donde x_i hace referencia a cada variable i y α_i se refiere al coeficiente o peso asignado a la variable i. Mediante la configuración de estos pesos, dado un conjunto de variables se puede obtener el valor de "y" deseado (o próximo al deseado).

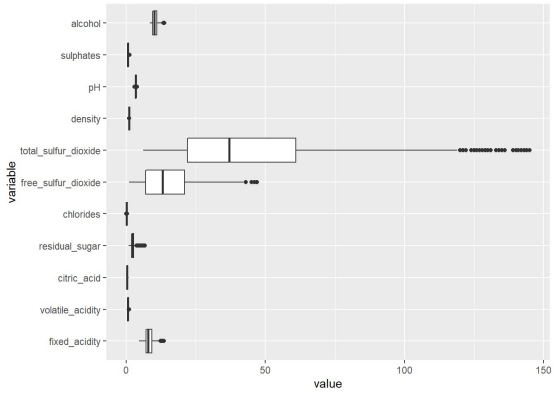
Si una variable tiene un rango de valores por ejemplo 10 órdenes mayor que las demás, entonces su aportación al valor "y" sería 10 veces mayor que las demás de manera que prácticamente solo estaríamos trabajando con el valor de esta, reduciendo drásticamente la flexibilidad y la precisión del modelo que se está construyendo.

Esta es la razón principal por la que se debe normalizar las variables si se pretende aplicar un proceso de minería de datos. Otra razón no menos importante es la adecuación de estas para que la comparación de valores entre distintas distribuciones de las mismas sea posible.

Procederemos a normalizar las variables numéricas mediante la normalización estándar puesto que esta es el procedimiento de normalización más adecuado para métodos de minería de datos que trabajan con distancias.

Como en ocasiones anteriores mantendremos las variables sin normalizar para posteriormente discretizarlas creando así categorías.
NOTA: almacenaremos las variables normalizadas con el posfijo "_n".

```
ggplot(melt(x[,head(colnames(x),-1)]), aes(x=variable, y=value)) + geom_boxplot() + coord_flip()
```

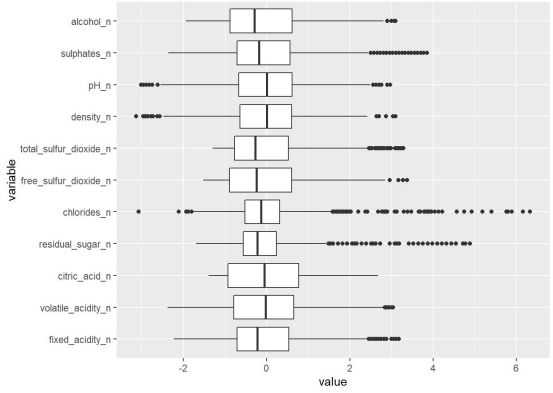


Como podemos ver las variables tiene distribuciones con rangos muy dispares. Se sigue el código que aplica la normalización, hemos usado la función de R scale que implementa la normalización por estandarización.

```
x[,paste(head(colnames(x),-1),"_n",sep="")] <- scale(x[,head(colnames(x),-1)])
```

Podemos apreciar las distribuciones de las variables normalizadas:

```
ggplot(melt(x[,tail(colnames(x),11)]), aes(x=variable, y=value)) + geom_boxplot() + coord_flip()
```



Ahora si podemos apreciar las distribuciones resultantes en las mismas escalas y con un número reducido de outliers (considerados legítimos).

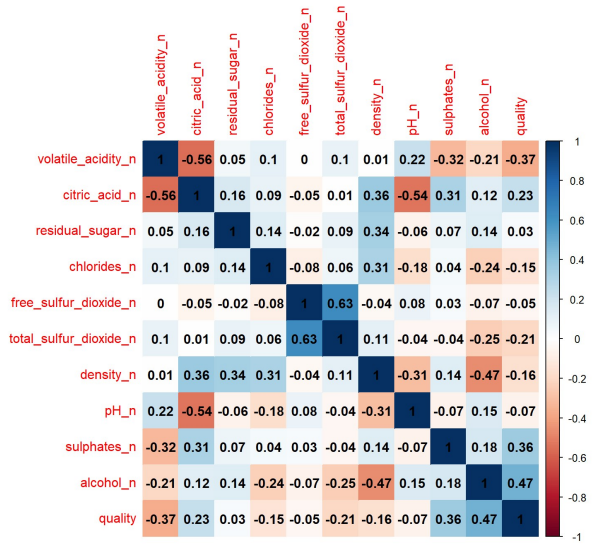
3.4 Discretización de variables

Ahora procederemos a discretizar algunas variables con el objetivo de crear categorías que faciliten y agilicen el proceso de análisis visual / visualización de datos. Buscamos que las visualizaciones que se obtengan más adelante sean más inteligibles. Además de la ventaja anteriormente comentada, permite la partición de los datos en conjuntos de clases que pueden facilitar la obtención de modelos de clasificación, la aplicación de algoritmos de reglas de asociación, etc.

Para decidir qué variables vamos a discretizar aplicaremos un análisis de correlaciones. Este resulta ser una herramienta muy útil puesto que nos permite ver rápidamente que variables tienen relación (ya sea directa o indirecta) con la variable objetivo del dataset. También nos permite saber que variables están correladas y cuáles no, lo que resulta de gran importancia a la hora de la selección de las mismas para el desarrollo de modelos de minería de datos.

En esta ocasión realizaremos dos análisis de correlaciones, uno sobre las variables normalizadas cuyo objetivo es el de ayudarnos a comprender las relaciones existentes entre los componentes que se añaden a los vinos y la calidad de estos. Y otro no tan inteligible cuyo objetivo es el de sentar una pequeña idea de por donde empezar cuando apliquemos algoritmos de minería de datos, y que se basará en analizar las correlaciones entre las componentes principales obtenidas y la variable quality.

```
corr <- cor(x[, c("volatile_acidity_n","citric_acid_n","residual_sugar_n","chlorides_n","free_sulfur_dioxide_n"
,"total_sulfur_dioxide_n" ,"density_n","pH_n","sulphates_n","alcohol_n","quality")])

corrplot(corr,method = "color",addCoef.col = "black")
```



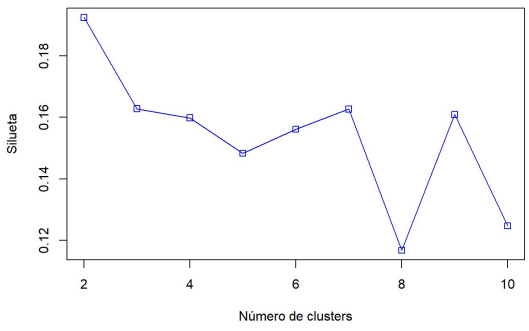
Podemos observar que las variables con mayor influencia sobre la calidad del vino son: la cantidad de alcohol, los sulfatos, la acidez volátil y el ácido cítrico.
Discretizaremos entonces estas variables y la variable quality (variable objetivo del dataset).

En un principio se evaluó la opción de aplicar una discretización usando chi-merge pero se descartó la idea debido a que es necesario conocer las clases (que por ahora ni siquiera existen), también evalué la opción de utilizar la discretización por obtención de intervalos de igual frecuencia pero fué finalmente descartada debido a que esta considera que se da una distribución uniforme de los valores por discretizar y sabemos que no es así.

3.5 Agrupamiento de observaciones mediante clusterización

Queremos estudiar si existe algún comportamiento o característica homogénea entre las observaciones de modo que podamos alcanzar a definir una clasificación no supervisada por pertenencia a clúster. Diseñamos una función para definir los clúster mediante el método k-mean. El número ideal de clúster será aquel

```
# Clasificación de vinos, en base a la calidad.
library(cluster)
set.seed (80)
y <- x[,12:23]
# función para comprobar el número óptimo de clúster.
d <- daisy(y)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(y, i)
  y_cluster     <- fit$cluster
  sk            <- silhouette(y_cluster, d)
  resultados[i] <- mean(sk[,3])
}

# Representación de los cluster
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="Silueta")
```

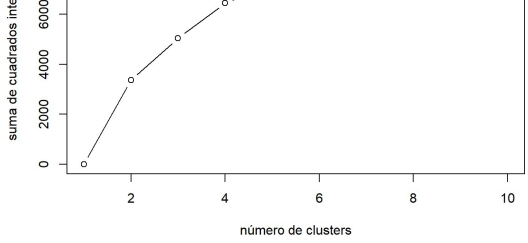


A la vista de la silueta construida para diferentes niveles de agrupamiento (entre 2 y 10), a la vista del gráfico la selección de 2 ó 4 clúster podría ser acertada. A partir de 4, la aportación de cada nuevo clúster a la clasificación es menos significativa.

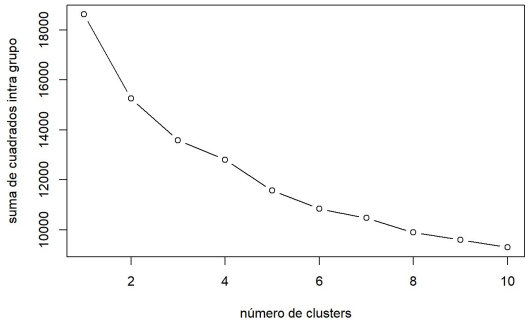
Para reforzar esta visión, y conseguir un mejor criterio entre seleccionar 2 ó 4 clúster, hagamos un estudio del cuadrado de las distancias intra grupos e inter grupos. Una buena selección del número idóneo de clúster será aquel con un bajo valor en la suma del cuadrado de las distancias intra grupo y alto valor en la suma del cuadrado de las distancias inter grupo.

En el gráfico siguiente, se muestra la evolución de estos valores, para un agrupamiento entre 2 y 10 clúster.

```
# Vector cuadrado de las distancias inter grupo
sumbt <-kmeans(y, centers=1)$betweenss
for (i in 2:10) sumbt[i]<- kmeans(y, centers = i)$betweenss

# Vector cuadrado de las distancias intra grupo
sumtotwi <-kmeans(y, centers=1)$tot.withinss
for (i in 2:10) sumtotwi[i]<- kmeans(y, centers = i)$tot.withinss

# Representación de la evolución del cuadrado de las distancias.
plot(1:10, sumbt, type = "b", xlab = "número de clusters", ylab = "suma de cuadrados inter grupo")
```

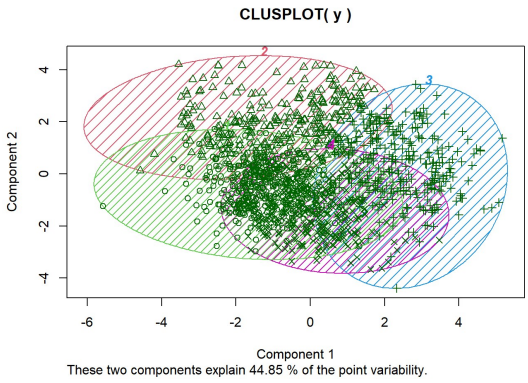



```
plot(1:10, sumtotwi, type = "b", xlab = "número de clusters", ylab = "suma de cuadrados intra grupo")
```

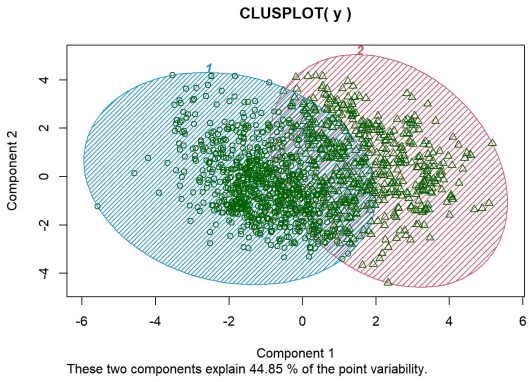


Podríamos afirmar que una clasificación correcta sería para 4 clúster. Descartamos la opción de más de 4 clúster, dado que no aporta mayor diferencia entre clúster, que la elección de 4 clúster. Por el contrario, la curva de 2 a 3 clúster se suaviza bastante. Nos surge la duda si una clasificación en 2 clúster podría ser óptima. Para salir de dudas, pasamos a representar la clasificación de las observaciones en 2 y 4 clúster.

```
# Agrupamiento en 4 clúster
fit4       <- kmeans(y, 4)
y_cluster4 <- fit4$cluster
clusplot(y, fit4$cluster, color=TRUE, shade=TRUE, labels=4, lines=0)
```




```
# Agrupamiento en 2 clúster
fit2       <- kmeans(y, 2)
y_cluster2 <- fit2$cluster
clusplot(y, fit2$cluster, color=TRUE, shade=TRUE, labels=4, lines=0)
```



No existe una clara separación entre los clúster, tanto para una elección de 4 clúster, como de 2 clúster. Las siluetas para cada uno de los agrupamientos, se solapan. Descartamos la idea de agrupar observaciones.

La calidad del vino la vamos a discretizar de otra manera distinta, vamos a diferenciar entre vinos de exquisita calidad(7-10) y vinos normales o de baja calidad(1-6).

3.6 Categorización de variables.

En general entre los datos que tenemos los valores se repiten muchísimo, esto es debido a la propia naturaleza de los datos puesto que estos corresponden a un producto el cual es obtenido tras un preciso y meticuloso proceso en el cual se miden y controlan todas y cada una de las condiciones que influyen en el mismo. Ante tal situación aunada al hecho del mínimo conocimiento a priori del que disponemos lo mejor es discretizar en intervalos que nos ofrezcan una partición "natural" y con natural me refiero a comprensible de los datos, creando así categorías intuitivas.

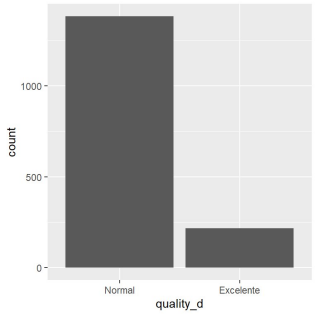
Por estas razones hemos decidido realizar particiones apoyándonos en los parámetros de las distribuciones de los datos. Para eso hemos desarrollado una función que discretiza usando estos como fronteras para crear las categorías baja, media y alta que hacen alusión al valor de dicho elemento que contiene el vino.

```
cortar <- function(x)
{
   q <- quantile(x)
   return( as.factor(cut(x,breaks = c(floor(q[1])-0.0001,q[2],q[4],ceiling(q[5])+0.0000) ,labels = c("Bajo", "M
edio", "Alto") )))
}

for ( col in c("alcohol","sulphates","volatile_acidity","citric_acid","quality") ) x[,paste(col,"_d",sep="")] <
- cortar(x[,col])
```

```
x$quality_d <- as.factor(cut(x$quality,breaks = c(0,6,10) ,labels = c("Normal", "Excelente") ))
```

Podemos observar el resultado por ejemplo con la variable alcohol:

```
ggplot(data = x,aes(x=quality_d)) + geom_bar()
```



Consideramos válida la categorización de la variable quality

3.7 Reducción de la dimensionalidad

La fase de reducción de la dimensionalidad pretende reducir la dimensión de los datos que poseemos reduciendo en la menor medida posible la información que estos albergan. Hay varias maneras de reducir la dimensionalidad: eliminar registros no deseados, eliminar características no deseadas, hallar un número menor de características que sean combinaciones lineales de las variables originales, etc.

Como punto final vamos a reducir el número de características obteniendo una combinación lineal de las variables numéricas normalizadas que tenemos, esto lo lograremos mediante el análisis de componentes principales PCA.

Nuestro objetivo es realizar una proyección de los datos de un espacio N dimensional a un espacio de dimension menor intentando mantener la mayor varianza posible de los datos del espacio de origen en el espacio transformado. Para esto usaremos la técnica PCA. El análisis PCA es una búsqueda de una matriz de transformación que permita pasar de un espacio vectorial de una dimensión mayor a un espacio vectorial de dimensión menor.

Una definición más formal:
"The goal of the PCA technique is to find a lower dimensional space or PCA space (W) that is used to transform the data $(X = \{x_1, x_2, \ldots, x_N\})$ from a higher dimensional space (R^M) to a lower dimensional space (R^k), where N represents the total number of samples or observations and x_i represents ith sample, pattern, or observation. All samples have the same dimension $(x_i \in R^M)$. In other words, each sample is represented by M variables, i.e. each sample is represented as a point in M-dimensional space (Wold et al., 1987). The direction of the PCA space represents the direction of the maximum variance of the given data as shown in Figure 1. As shown in the figure, the PCA space is consists of a number of PCs. Each principal component has a different robustness according to the amount of variance in its direction." (Tharwat, 2016)

Cuando habla de "PCA space(W)", W es la letra que se suele usar en textos científicos para denotar la matriz de transformación de la que hablábamos antes. Esta tiene tantas columnas (k) como componentes principales (PCs), tantas filas como variables en los datos originales (M) y almacena los coeficientes necesarios para proyectar vectores de un espacio de dimensión M (R^M) a un espacio de dimensión k (R^k) menor.

Para aplicar PCA se hace una suposición de linealidad en la que se asume que los datos son una combinación lineal de una cierta base, y por lo tanto puede aplicarse un cambio de base sin perder demasiada información.

```
original_data  <- x[,c("fixed_acidity_n","volatile_acidity_n","citric_acid_n","residual_sugar_n","chlorides_n",
"free_sulfur_dioxide_n","total_sulfur_dioxide_n","density_n","pH_n","sulphates_n","alcohol_n")]
pca <- prcomp(original_data,center=FALSE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.7369 1.3953 1.2476 1.0654 0.94571 0.86538 0.76047
## Proportion of Variance 0.2743 0.1770 0.1415 0.1032 0.08131 0.06808 0.05257
## Cumulative Proportion  0.2743 0.4512 0.5928 0.6959 0.77724 0.84532 0.89790
##                            PC8     PC9    PC10    PC11
## Standard deviation     0.66895 0.5773 0.47857 0.3366
## Proportion of Variance 0.04068 0.0303 0.02082 0.0103
## Cumulative Proportion  0.93858 0.9689 0.98970 1.0000
```

En el resumen para el análisis podemos ver una matriz con tantas columnas como componentes principales (PCs) (combinaciones de nuestras variables) y tres filas, la filas que más nos interesan son las dos últimas porque estas hacen referencia a la varianza que se ha podido traspasar al espacio transformado. La segunda fila expresa la varianza que cada componente principal aporta mientras que la tercera realiza la suma acumulada.

Las componentes principales vienen ordenadas según el porcentaje de varianza que explican (por lo tanto la primera componente representa la dirección de máxima varianza), son ortonormales (vectores perpendiculares unitarios) e incorreladas (no hay redundancia de información).

"The PCA space consists of k principal components. The principal components are orthonormal, uncorrelated, and it represents the direction of the maximum variance. The first principal component $((PC_1 \text{ or } v_1) \in R^{Mx1})$ of the PCA space represents the direction of the maximum variance of the data, the second principal component has the second largest variance, and so on." (Tharwat, 2016)

Vemos que con 10 componentes principales (10 autovectores) obtenemos una representatividad del 98,97% de la variabilidad de los datos originales. Aplicaremos PCA y nos quedaremos con 10 componentes principales.

```
x[,c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9","PC10")] <- data.frame(pca$x[,1:10])
```

Una de las ventajas extras a parte de la reducción de la dimensionalidad, consiste en que al quedar solo aquellas componentes principales que explican la mayor variabilidad, pueden quedar expuestos patrones y relaciones que contemplando todas las componentes quedaban diluidas y pasaban desapercibidas.
Intentamos aportar sentido a las cuatro primeras de las componentes principales, según su relación lineal con el resto de las variables.

```
pca$rotation
```

```
##                                PC1           PC2           PC3           PC4
## fixed_acidity_n         0.51014691 -0.004875429  0.089470006 -0.069503459
## volatile_acidity_n     -0.25542314  0.373300678  0.321621782  0.200501845
## citric_acid_n           0.47347564 -0.178406662 -0.196782171 -0.058575190
## residual_sugar_n        0.16893218  0.102239264  0.002208628  0.789868066
## chlorides_n             0.18634625  0.281507989  0.240181807  0.157791956
## free_sulfur_dioxide_n  -0.08373169  0.300848493 -0.631966970 -0.013534244
## total_sulfur_dioxide_n -0.01607312  0.456205164 -0.518761507  0.008672401
## density_n               0.39848478  0.343880340  0.147958770  0.134584768
## pH_n                   -0.41622173 -0.042564545 -0.014111300  0.304728167
## sulphates_n             0.20718191 -0.221514799 -0.261611370  0.238093384
## alcohol_n              -0.06958898 -0.520890660 -0.181966912  0.367189599
##                                PC5           PC6           PC7           PC8
## fixed_acidity_n         0.16232203 -0.07584476  0.237306448 -0.39625313
## volatile_acidity_n      0.12329028 -0.01989941  0.672954758 -0.10340579
## citric_acid_n           0.10528895  0.07882924 -0.139890373 -0.09798526
## residual_sugar_n        0.31790529 -0.04222893 -0.208421289  0.32769316
## chlorides_n            -0.46706574  0.73657486 -0.142369750 -0.14143617
## free_sulfur_dioxide_n   0.02169907  0.04542440  0.108770361 -0.36307267
## total_sulfur_dioxide_n  0.08373719  0.12339037 -0.005081353  0.20140612
## density_n              -0.07281414 -0.42808203 -0.100871541 -0.29653989
## pH_n                   -0.29576490 -0.27762147 -0.385345720 -0.49270858
## sulphates_n            -0.69323212 -0.24267289  0.413016791  0.26399748
## alcohol_n               0.21752888  0.32681055  0.257747114 -0.35003441
##                                PC9          PC10          PC11
## fixed_acidity_n        -0.0354032420  0.312943110  0.618281658
## volatile_acidity_n      0.2228128090 -0.350071998  0.020267375
## citric_acid_n           0.4072139210 -0.699389940 -0.019947340
## residual_sugar_n       -0.2183016824 -0.071710405  0.175058012
## chlorides_n            -0.0493592526  0.005732680  0.002694861
## free_sulfur_dioxide_n  -0.5482909675 -0.235499537 -0.025372419
## total_sulfur_dioxide_n  0.5757635955  0.350453229  0.072672290
## density_n               0.0447579616  0.174915984 -0.603063312
## pH_n                    0.2792585188 -0.068302704  0.305671629
## sulphates_n             0.0004199383 -0.002231496  0.055984755
## alcohol_n               0.1451566397  0.267677668 -0.346535338
```

Componente 1: Relaciona positivamente la acidez volátil, acidez citrica y densidad, produciendo un descenso del Ph. *Componente 2*: Relaciona la contribución del dióxido de sulfuro a un menor nivel de alcohol del vino. *Componente 3*: Define vinos con un alto contenido en dióxido de sulfuro.

Componente 4: Denota una clara relación entre el nivel de azúcar y la graduación de alcohol.

Finalmente el dataset queda de la siguiente manera:

```
str(x)
```

```
## 'data.frame':    1599 obs. of  38 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               : num  5 5 5 6 5 5 5 7 7 5 ...
##  $ fixed_acidity_n       : num  -0.527 -0.284 -0.284 1.781 -0.527 ...
##  $ volatile_acidity_n    : num  1.04 2.1 1.39 -1.43 1.04 ...
##  $ citric_acid_n         : num  -1.39 -1.39 -1.19 1.49 -1.39 ...
##  $ residual_sugar_n      : num  -0.559 0.236 -0.105 -0.559 -0.559 ...
##  $ chlorides_n           : num  -0.261 0.706 0.442 -0.305 -0.261 ...
##  $ free_sulfur_dioxide_n : num  -0.4589 1.0341 -0.0323 0.1809 -0.4589 ...
##  $ total_sulfur_dioxide_n: num  -0.37 0.718 0.289 0.487 -0.37 ...
##  $ density_n             : num  0.605 0.039 0.152 0.719 0.605 ...
##  $ pH_n                  : num  1.352 -0.732 -0.329 -1.001 1.352 ...
##  $ sulphates_n           : num  -0.6347 0.2626 0.0383 -0.4851 -0.6347 ...
##  $ alcohol_n             : num  -0.971 -0.584 -0.584 -0.584 -0.971 ...
##  $ alcohol_d             : Factor w/ 3 levels "Bajo","Medio",..: 1 2 2 2 1 1 1 2 1 2 ...
##  $ sulphates_d           : Factor w/ 3 levels "Bajo","Medio",..: 2 2 2 2 2 2 1 1 2 3 ...
##  $ volatile_acidity_d    : Factor w/ 3 levels "Bajo","Medio",..: 3 3 3 1 3 3 2 3 2 2 ...
##  $ citric_acid_d         : Factor w/ 3 levels "Bajo","Medio",..: 1 1 1 3 1 1 1 1 1 2 ...
##  $ quality_d             : Factor w/ 2 levels "Normal","Excelente": 1 1 1 1 1 1 1 2 2 1 ...
##  $ PC1                   : num  -1.679 -0.853 -0.755 2.45 -1.679 ...
##  $ PC2                   : num  0.9989 2.1854 1.3311 0.0251 0.9989 ...
##  $ PC3                   : num  1.393 0.122 0.756 -0.683 1.393 ...
##  $ PC4                   : num  -0.167 0.4424 0.0736 -1.5244 -0.167 ...
##  $ PC5                   : num  -0.416 -0.202 -0.283 0.732 -0.416 ...
##  $ PC6                   : num  -1.1233 0.4475 0.0906 -0.2241 -1.1233 ...
##  $ PC7                   : num  -0.219 1.736 0.967 -0.611 -0.219 ...
##  $ PC8                   : num  -0.489 0.402 0.39 -0.455 -0.489 ...
##  $ PC9                   : num  0.119 -0.618 -0.148 0.211 0.119 ...
##  $ PC10                  : num  0.219 0.047 0.267 0.22 0.219 ...
```

3.8 Grabación de fichero depurado y limpio.

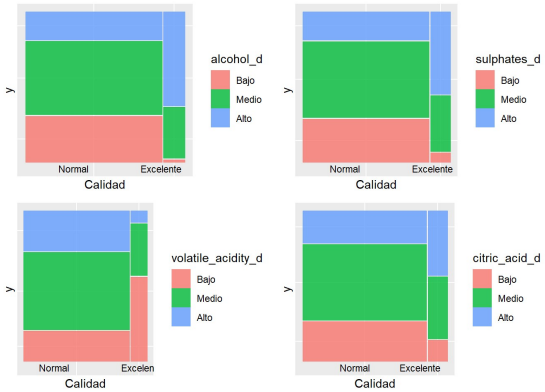
Una vez depurado y limpio el fichero, lo guardamos con nombre "winequality-read_out.csv".

```
# Salvamos el fichero depurado y limpio.

write.csv(x, file ="winequality-red_out.csv")
```

4 Análisis visual del conjunto de datos

En esta fase vamos a realizar un estudio visual que intentará arrojar luz entre la relación existente entre la calidad del vino y las principales características que influyen en esta.

```
p1 <- ggplot(data = x) + geom_mosaic(aes(x = product(quality_d), fill=alcohol_d)) + xlab("Calidad") +   annotat
e(geom="text",x=0.5,y=-0.04,label="            Normal                    Excelente", color="black",size=3)
p2 <- ggplot(data = x) + geom_mosaic(aes(x = product(quality_d), fill=sulphates_d)) + xlab("Calidad") +   annot
ate(geom="text",x=0.5,y=-0.04,label="            Normal                    Excelente", color="black",size=3)
p3 <- ggplot(data = x) + geom_mosaic(aes(x = product(quality_d), fill=volatile_acidity_d)) + xlab("Calidad") +
  annotate(geom="text",x=0.5,y=-0.04,label="            Normal                    Excelente", color="black",si
ze=3)
p4 <- ggplot(data = x) + geom_mosaic(aes(x = product(quality_d), fill=citric_acid_d)) + xlab("Calidad") +   ann
otate(geom="text",x=0.5,y=-0.04,label="            Normal            Excelente", color="black",size=3)

grid.arrange(p1,p2,p3,p4,ncol = 2,nrow=2)
```



De las gráficas podemos concluir que parece ser que:

- La proporción de vinos con alto contenido en alcohol es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
- La proporción de vinos con alto contenido en sulfatos es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
- La proporción de vinos con una acidez volátil baja es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
- La proporción de vinos con alto contenido en ácido cítrico es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.

5 Análisis estadístico

5.1 Estudio de la normalidad

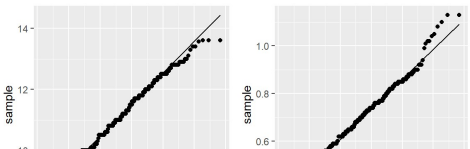
En esta sección vamos a estudiar la normalidad de los datos de manera independiente en los dos conjuntos que hemos separado: vinos de calidad normal y excelentes.

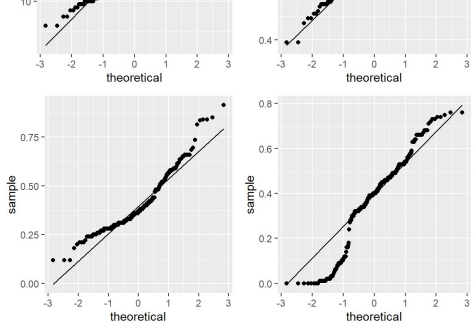
```
x_excelente <- x[x$quality_d == "Excelente",]
x_normal <- x[x$quality_d == "Normal",]
```

Para comprobar la normalidad de los datos en cada variable vamos a generar una serie de gráficas cuantil-cuantil (Normal Q-Q).

Para el caso de los vinos excelentes:

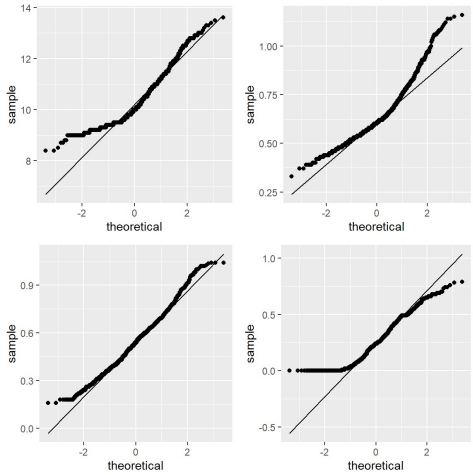
```
p1 <- ggplot(x_excelente, aes(sample = alcohol)) + stat_qq() + stat_qq_line()
p2 <- ggplot(x_excelente, aes(sample = sulphates)) + stat_qq() + stat_qq_line()
p3 <- ggplot(x_excelente, aes(sample = volatile_acidity)) + stat_qq() + stat_qq_line()
p4 <- ggplot(x_excelente, aes(sample = citric_acid)) + stat_qq() + stat_qq_line()
grid.arrange(p1,p2,p3,p4,ncol = 2,nrow=2)
```





Para el caso de los vinos normales:

```
p1 <- ggplot(x_normal, aes(sample = alcohol)) + stat_qq() + stat_qq_line()
p2 <- ggplot(x_normal, aes(sample = sulphates)) + stat_qq() + stat_qq_line()
p3 <- ggplot(x_normal, aes(sample = volatile_acidity)) + stat_qq() + stat_qq_line()
p4 <- ggplot(x_normal, aes(sample = citric_acid)) + stat_qq() + stat_qq_line()
grid.arrange(p1,p2,p3,p4,ncol = 2,nrow=2)
```



Vemos que para ambos conjuntos al graficar en el eje de ordenadas los valores de la muestra y en el eje de abcisas los valores esperados teóricos de la distribución normal la línea de puntos generada en las distintas gráficas se aleja de la diagonal principal lo que significa que probablemente los datos no se distribuyan de manera normal. Además se aprecia que los datos no cumplen la condición de homocedasticidad pueto que se observa que la varianza en cada una de las variables no es constante.

Afianzaremos esta estimación mediante la aplicación de un contraste de hipótesis sobre la normalidad de los datos. El test de Shapiro-Wilk es un contraste de hipótesis que establece como hipótesis nula (H_0) que los datos siguen una distribución normal. En el caso de que el p-value retornado fuera menor al nivel de significación (0.05, 95% de confianza) entonces podríamos rechazar la H_0 y concluir que nuestros datos no se distribuyen normalmente.

```
p_values_excelentes <- c(shapiro.test(x_excelente$alcohol)$p,shapiro.test(x_excelente$sulphates)$p,shapiro.test
(x_excelente$volatile_acidity)$p,shapiro.test(x_excelente$citric_acid)$p)
p_values_excelentes
```

```
## [1] 6.336326e-02 3.009774e-02 7.871624e-09 5.420347e-07
```

```
p_values_normales <- c(shapiro.test(x_normal$alcohol)$p,shapiro.test(x_normal$sulphates)$p,shapiro.test(x_norma
l$volatile_acidity)$p,shapiro.test(x_normal$citric_acid)$p)
p_values_normales
```

```
## [1] 2.522770e-26 4.759972e-25 3.141956e-08 2.409199e-21
```

Los p-values retornados nos afirman que los datos no siguen una distribución normal. La única variable que la sigue es la cantidad de alcohol en los vinos excelentes.

5.2 Estudio de la homocedasticidad.

Veamos ahora si existen evidencias para afirmar que las varianzas son distintas para el conjunto de datos con calidad normal, o excelente. Se plantea el siguiente contraste de hipótesis para la varianza.

$H_0: \sigma_1^2 = \sigma_2^2$
$H_1: \sigma_1^2 \neq \sigma_2^2$

```
# Test de homoscedasticidad sobre las muestras "calidad normal" y "calidad excelente".
mean.normal <- mean(x_normal$alcohol); n1 <- length(x_normal); s1 <- sd(x_normal$alcohol)
mean.excelente <- mean(x_excelente$alcohol); n2 <- length(x_excelente); s2 <- sd(x_excelente$alcohol)
fobs <- s1^2 / s2^2
alfa <- 0.05
fcritL <- qf( alfa/2, df1=n1-1, df2=n2-1 )
fcritU <- qf( 1- alfa/2, df1=n1-1, df2=n2-1)
pvalue <- min(pf( fobs, df1=n1-1, df2=n2-1, lower.tail=FALSE ), pf( fobs, df1=n1-1, df2=n2-1))*2
cat("El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable alcohol, para un
\n nivel de confianza del",(1-alfa)*100,"%, es [",round(fcritL,2),round(fcritU,2),"].\n")
```

```
## El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable alcohol, para un
##  nivel de confianza del 95 %, es [ 0.52 1.92 ].
```

```
cat("El valor del estadistico observado es",round(fobs,2),".\n")
```

```
## El valor del estadistico observado es 0.94 .
```

```
cat("La probabilidad de error tipo I es de",pvalue,".\n")
```

```
## La probabilidad de error tipo I es de 0.8628634 .
```

Se acepta la hipótesis nula que iguala la varianza de la variable alcohol para las muestras de calidad normal y excelente, con un nivel de confianza del 95%. Pasamos ahora a repetir el mismo contraste para la diferencia de varianzas de la variable sulphates.

```
mean.normal <- mean(x_normal$sulphates); n1 <- length(x_normal); s1 <- sd(x_normal$sulphates)
mean.excelente <- mean(x_excelente$sulphates); n2 <- length(x_excelente); s2 <- sd(x_excelente$sulphates)
fobs <- s1^2 / s2^2
alfa <- 0.05
fcritL <- qf( alfa/2, df1=n1-1, df2=n2-1 )
fcritU <- qf( 1- alfa/2, df1=n1-1, df2=n2-1)
pvalue <- min(pf( fobs, df1=n1-1, df2=n2-1, lower.tail=FALSE ), pf( fobs, df1=n1-1, df2=n2-1))*2
cat("El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable sulphates, para
un\n nivel de confianza del",(1-alfa)*100,"%, es [",round(fcritL,2),round(fcritU,2),"].\n")
```

```
## El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable sulphates, para un
##  nivel de confianza del 95 %, es [ 0.52 1.92 ].
```

```
cat("El valor del estadistico observado es",round(fobs,2),".\n")
```

```
## El valor del estadistico observado es 1.01 .
```

```
cat("La probabilidad de error tipo I es de",pvalue,".\n")
```

```
## La probabilidad de error tipo I es de 0.9644305 .
```

Se acepta la hipótesis nula que iguala la varianza de la variable sulphates para las muestras de calidad normal y excelente, con un nivel de confianza del 95%. Pasamos ahora a repetir el mismo contraste para la diferencia de varianzas de la variable volatile_acidity.

```
mean.normal <- mean(x_normal$volatile_acidity); n1 <- length(x_normal); s1 <- sd(x_normal$volatile_acidity)
mean.excelente <- mean(x_excelente$volatile_acidity); n2 <- length(x_excelente); s2 <- sd(x_excelente$volatile_
acidity)
fobs <- s1^2 / s2^2
alfa <- 0.05
fcritL <- qf( alfa/2, df1=n1-1, df2=n2-1 )
fcritU <- qf( 1- alfa/2, df1=n1-1, df2=n2-1)
pvalue <- min(pf( fobs, df1=n1-1, df2=n2-1, lower.tail=FALSE ), pf( fobs, df1=n1-1, df2=n2-1))*2
cat("El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable volatile_acidity
, para un\n nivel de confianza del",(1-alfa)*100,"%, es [",round(fcritL,2),round(fcritU,2),"].\n")
```

```
## El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable volatile_acidity,
para un
##  nivel de confianza del 95 %, es [ 0.52 1.92 ].
```

```
cat("El valor del estadistico observado es",round(fobs,2),".\n")
```

```
## El valor del estadistico observado es 1.31 .
```

```
cat("La probabilidad de error tipo I es de",pvalue,".\n")
```

```
## La probabilidad de error tipo I es de 0.4115522 .
```

Se acepta la hipótesis nula que iguala la varianza de la variable volatile_acidity para las muestras de calidad normal y excelente, con un nivel de confianza del 95%. Pasamos ahora a repetir el mismo contraste para la diferencia de varianzas de la variable citric_acid.

```
mean.normal <- mean(x_normal$citric_acid); n1 <- length(x_normal); s1 <- sd(x_normal$citric_acid)
mean.excelente <- mean(x_excelente$citric_acid); n2 <- length(x_excelente); s2 <- sd(x_excelente$citric_acid)
fobs <- s1^2 / s2^2
alfa <- 0.05
fcritL <- qf( alfa/2, df1=n1-1, df2=n2-1 )
fcritU <- qf( 1- alfa/2, df1=n1-1, df2=n2-1)
pvalue <- min(pf( fobs, df1=n1-1, df2=n2-1, lower.tail=FALSE ), pf( fobs, df1=n1-1, df2=n2-1))*2
cat("El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable citric_acid, par
a un\n nivel de confianza del",(1-alfa)*100,"%, es [",round(fcritL,2),round(fcritU,2),"].\n")
```

```
## El intervalo de aceptación para la igualdad de las varianzas poblacionales en la variable citric_acid, para
un
##  nivel de confianza del 95 %, es [ 0.52 1.92 ].
```

```
cat("El valor del estadistico observado es",round(fobs,2),".\n")
```

```
## El valor del estadistico observado es 0.94 .
```

```
cat("La probabilidad de error tipo I es de",pvalue,".\n")
```

```
## La probabilidad de error tipo I es de 0.8538702 .
```

Se acepta la hipótesis nula que iguala la varianza de la variable volatile_acidity para las muestras de calidad normal y excelente, con un nivel de confianza del 95%.

En conclusión, se acepta que las varianzas de las distintas muestras para niveles de calidad de vino, son iguales, con un nivel de confianza del 95.

5.3 Comparación de parámetros mediante test de hipótesis

Procederemos a comparar proporciones de los niveles de distintos componentes del vino en ambos grupos. No importa que las variables no sigan una distribución normal ya que para los contrastes de proporción solo necesitamos que las variables sigan una distribución de Bernoulli (es decir que los valores consistan en éxitos o fracasos) y eso lo lograremos dicotomizando según necesitemos las variables.

La función que usaremos para realizar los contrastes es la siguiente:

```
CH_proporciones <- function(p1,p2,n1,n2,sig = 0.05)
{
  p0 = (n1*p1+n2*p2)/(n1+n2)
  z <- (p1-p2)/sqrt(p0*(1-p0)*(1/n1+1/n2))
  lim1 <-  qnorm( sig/2 )
  lim2 <-  qnorm( 1 - (sig/2) )
  p_value <- 2*min(  pnorm(z) , 1 - pnorm(z)  )
  return (  c(z,lim1,lim2, p_value) )
}
```

Y ahora obtenemos las distintas proporciones de los datos a estudiar y realizamos los contrastes.

```
n1 <- length(x_excelente$alcohol_d)
n2 <- length(x_normal$alcohol_d)
p1 <- sum(x_excelente$alcohol_d=="Alto")/n1
p2 <- sum(x_normal$alcohol_d=="Alto")/n2
pvalue1 <- CH_proporciones(p1=p1,p2=p2,n1=n1,n2=n2)
pvalue1
```

```
## [1] 14.114258 -1.959964  1.959964  0.000000
```

```
p1 <- sum(x_excelente$sulphates_d=="Alto")/n1
p2 <- sum(x_normal$sulphates_d=="Alto")/n2
pvalue2 <- CH_proporciones(p1=p1,p2=p2,n1=n1,n2=n2)
pvalue2
```

```
## [1] 11.741746 -1.959964  1.959964  0.000000
```

```
p1 <- sum(x_excelente$volatile_acidity_d=="Bajo")/n1
p2 <- sum(x_normal$volatile_acidity_d=="Bajo")/n2
pvalue3 <- CH_proporciones(p1=p1,p2=p2,n1=n1,n2=n2)
pvalue3
```

```
## [1] 11.559425 -1.959964  1.959964  0.000000
```

```
p1 <- sum(x_excelente$citric_acid_d=="Alto")/n1
p2 <- sum(x_normal$citric_acid_d=="Alto")/n2
pvalue4 <- CH_proporciones(p1=p1,p2=p2,n1=n1,n2=n2)
pvalue4
```

```
## [1]  6.950695e+00 -1.959964e+00  1.959964e+00  3.634870e-12
```

Hemos aplicado contrastes bilateral para la diferencia de proporciones y con los resultados afirmaremos o retiraremos las siguientes afirmaciones:

Las hipótesis a aplicar (en todos los casos) son las siguientes:
- $H_0: \pi_1 - \pi_2 = 0$.
- $H_1: \pi_1 - \pi_2 < 0$.
- $H_1: \pi_1 - \pi_2 > 0$.

- La proporción de vinos con alto contenido en alcohol es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
  - El p-value retornado (0) es inferior al nivel de significación (0.05), lo que significa que tenemos evidencia suficiente para descartar la

hipótesis nula.
Podemos comprobar que T_Exp (14.114258) se halla fuera del intervalo de aceptación: [-1.959964,1.959964], concretamente a la derecha de este, lo que indica que $\pi_1 - \pi_2 > 0$ y por lo tanto se verifica la afirmación.

- La proporción de vinos con alto contenido en sulfatos es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
  - El p-value retornado (0) es inferior al nivel de significación (0.05), lo que significa que tenemos evidencia suficiente para descartar la hipótesis nula.
    Podemos comprobar que T_Exp (11.741746) se halla fuera del intervalo de aceptación: [-1.959964,1.959964], concretamente a la derecha de este, lo que indica que $\pi_1 - \pi_2 > 0$ y por lo tanto se verifica la afirmación con un nivel de confianza del 95%.
- La proporción de vinos con una acidez volátil baja es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
  - El p-value retornado (0) es inferior al nivel de significación (0.05), lo que significa que tenemos evidencia suficiente para descartar la hipótesis nula.
    Podemos comprobar que T_Exp (11.559425) se halla fuera del intervalo de aceptación: [-1.959964,1.959964], concretamente a la derecha de este, lo que indica que $\pi_1 - \pi_2 > 0$ y por lo tanto se verifica la afirmación con un nivel de confianza del 95%.
- La proporción de vinos con alto contenido en ácido cítrico es mayor en el conjunto de vinos de excelente calidad que en el de calidad normal.
  - El p-value retornado (3.634870e-12) es inferior al nivel de significación (0.05), lo que significa que tenemos evidencia suficiente para descartar la hipótesis nula.
    Podemos comprobar que T_Exp (6.950695) se halla fuera del intervalo de aceptación: [-1.959964,1.959964], concretamente a la derecha de este, lo que indica que $\pi_1 - \pi_2 > 0$ y por lo tanto se verifica la afirmación con un nivel de confianza del 95%.

Los p-values nulos se dan cuando se aplica un contraste sobre datos con evidencias muy significativas.

5.4 Regresión logística

Vamos a obtener un modelo de regresión logística que explique la relación existente entre los componentes del vino y la calidad de este (normal o excelente). Para ello divideremos los datos en conjuntos de training y test, el conjunto de training será usado para entrenar y generar el modelo y el conjunto de test será usado para calcular el error de generalización y así estimar el riesgo de predicción.

```
library(caret)

x <- x[,c("alcohol","sulphates","volatile_acidity","citric_acid","quality_d")]
levels(x$quality_d) <- c(0,1)

set.seed(5)

indexes = sample(1:nrow(x), size=floor((2/3)*nrow(x)))


x_training <- x[indexes,]
x_test<- x[-indexes,]



model <- glm(data = x_training, quality_d ~ .,family=binomial(link=logit))
summary(model)
```

```
##
## Call:
## glm(formula = quality_d ~ ., family = binomial(link = logit),
##     data = x_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7791  -0.4627  -0.2352  -0.1414   2.9736
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -14.4300     1.4926  -9.668  < 2e-16 ***
## alcohol            1.0590     0.1063   9.964  < 2e-16 ***
## sulphates          4.5933     0.7739   5.935 2.93e-09 ***
## volatile_acidity  -4.1968     0.8995  -4.666 3.07e-06 ***
## citric_acid       -0.2076     0.6710  -0.309    0.757
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 866.14  on 1065  degrees of freedom
## Residual deviance: 604.95  on 1061  degrees of freedom
## AIC: 614.95
##
## Number of Fisher Scoring iterations: 6
```

Vemos que el estimador del parámetro que incorpora la variable citric_acid en el modelo de regresión, no es significativo, para un nivel de confianza del 95%. Por tanto, se descarta este parámetro del modelo de regresión logístico.

Veamos el error de generalización:

```
y_est <- as.factor(predict( model, x_test, type="response") > 0.5)
levels(y_est) <- c(0,1)
print(sprintf("La precisión del clasificador es: %.4f %%",100*sum(y_est == x_test$quality_d) / length(x_test$quality_d)))
```

```
## [1] "La precisión del clasificador es: 88.5553 %"
```

Vemos que la precisión del modelo obtenido es de un 88.5553%, una clasificación excelente.

Ahora veamos los OR del modelo junto con unos intervalos de confianza al 95% para estos:

```
exp(coefficients(model))
```

```
##      (Intercept)          alcohol        sulphates volatile_acidity
##     5.409301e-07     2.883512e+00     9.882278e+01     1.504379e-02
##      citric_acid
##     8.125096e-01
```

```
exp(confint(model))
```

```
##                        2.5 %       97.5 %
## (Intercept)      2.649383e-08 9.317335e-06
## alcohol          2.353910e+00 3.573243e+00
## sulphates        2.195739e+01 4.587573e+02
## volatile_acidity 2.475448e-03 8.438994e-02
## citric_acid      2.160499e-01 3.011046e+00
```

Basándonos en que:

- Un OR = 1 implica que no existe asociación entre la variable respuesta y la covariable.
- Un OR inferior a la unidad se interpreta como un factor de protección, es decir, el suceso es menos probable en presencia de dicha covariable.
- Un OR mayor a la unidad se interpreta como un factor de riesgo, es decir, el suceso es más probable en presencia de dicha covariable.

Observando los IC podemos afirmar con un 95% de confianza:

- El intervalo para la estimación del OR poblacional de la variable alcohol se situa por encima de 1 por lo que podemos decir que dicho OR > 1.
- El intervalo para la estimación del OR poblacional de la variable sulphates se situa por encima de 1 por lo que podemos decir que dicho OR > 1.
- El intervalo para la estimación del OR poblacional de la variable volatile_acidity se situa por debajo de 1 por lo que podemos decir que dicho OR < 1.
- El intervalo para la estimación del OR poblacional de la variable citric_acid contiene numeros por debajo de uno y por encima de uno. No podemos afirmar nada.

A partir de los IC para los OR y los OR devueltos por glm podemos concluir con un nivel de confianza del 95% que:

- El incremento del valor de la variable alcohol en una unidad aumenta el odd de la variable quality_d en un 288% (la probabilidad de ser un vino excelente entre la probabilidad de no ser un vino excelente). **Por lo tanto a mayor cantidad de alcohol más probabilidad de ser un vino excelente**.
- El incremento del valor de la variable sulphates en una unidad aumenta el odd de la variable quality_d en un 931% (la probabilidad de ser un vino excelente entre la probabilidad de no ser un vino excelente). **Por lo tanto a mayor cantidad de sulfatos más probabilidad de ser un vino excelente**.
- El incremento del valor de la variable volatile_acidity en una unidad aumenta el odd de la variable quality_d en un 3,2% (la probabilidad de ser un vino excelente entre la probabilidad de no ser un vino excelente). **Por lo tanto a mayor acidez volátil más probabilidad de ser un**

vino excelente.

La variable citric_acid no se encuentra entre las conclusiones debido a que mediante el intervalo de confianza del 95% obtenido para ella no se puede afirmar si esta reduce o incrementa el odd de la variable quality_d. Aunque observando el intervalo [0.59 , 4.97] se ve que es más probable que esta tenga una influencia incremental, pero no podemos afirmarlo.

6 Bibliografía

Pukelsheim, Friedrich. (1994). The Three Sigma Rule, The American Statistician. 48:2. 88-91. DOI: 10.1080/00031305.1994.10476030

Tharwat, Alaa. (2016). Principal component analysis - a tutorial. International Journal of Applied Pattern Recognition. 3. 197. DOI: 10.1504/IJAPR.2016.079733.