M2.851 - Tipología y ciclo de vida de los datos. Práctica 2

Gustavo Álvarez Bea y Adrià Marimon Robert

2022-01-03

Table of Contents

1. Descripción del dataset

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

El conjunto de datos se denomina "winequality-red" y contiene 1.599 observaciones de variantes tintas del vino portugués "Vinho Verde". Se obtiene a partir del siguiente dataset de Kaggle: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009.

En el apartado de descripción del proyecto mencionado, se indica, a su vez, que los datos originales han sido descargados del UCI Machine Learning Repository, de la UC Irvine: https://archive.ics.uci.edu/ml/datasets/wine+quality. El dataset fue creado por Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos y Jose Reis.

Se distribuye como fichero de tipo CSV, separado por comas. Se procede a su carga en el proyecto.

El dataset puede ser de importancia para una persona que se inicia en el mundo del vino puede ser relevante a la hora de producirlo.

```
# Carga del archivo

wine <- read.csv("winequality-red.csv", header=TRUE)
wine1<-wine
wine_raw<-wine
```

Está compuesto por 11 variables fisicoquímicas (entradas) y una sensorial (salida), con tipo de datos numérico decimal las 11 primeras y entero la última.

```
# Examen del tipo de datos de cada variable
str(wine)
```

```
## '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 0.5
8 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 0.06
9 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 ...
```

La descripción de las 11 variables de entrada es la siguiente:

- fixed acidity: acidez fija. Conjunto de ácidos naturales del vino, que permiten conservar sus cualidades naturales. El más importante es el ácido tártico, pero también inciden los ácidos málico, cítrico, succínico y láctico (Agrovin, 2020).
- volatile acidity: acidez volátil. Un vino de calidad acostumbra a manifestar una volatilidad de ácidos baja (Vinetur, 2019).
- citric acid: ácido cítrico, presente en la acidez fija.
- residual sugar: azúcar residual. Cantidad de azúcar que queda en el vino que no ha sido fermentada por las levaduras y que determina si un vino es seco, semiseco, semidule o dulce (Catatu, 2018).
- chlorides: cloruros. Forman parte de las sales minerales del vino (Vinetur, 2015).
- free sulfur dioxide: dióxido de azufre libre, disuelto en el líquido.
- total sulfur dioxide: dióxido de azufre total, como suma de la parte libre y el combinado en las sustancias orgánicas presentes en el vino. Su finalidad es la conservación y la eliminación de bacterias, al aportar funciones antioxidantes y garantizar la estabilidad microbiana del vino. Al oxidarse ante los fenoles presentes en el vino, genera los conocidos sulfitos, y su concentración está sujeta a regulación, por su posible reacción adversa en el consumidor (Carlos Serres, 2017).
- density: densidad. Determina la estructura o espesor en boca del vino.
- pH: pH. Resulta del equilibrio de los ácidos que componen un vino (Lucas Díaz, 2015).
- sulphates: sulfatos. Derivados de los tratamientos aplicados a la viña ante enfermedades como el mildium (siendo el más conocido el sulfato de cobre), y como antireductivos y aportes nutritivos en la fermentación alcohólica (como el sulfato de amonio) (Nieto Pardal, 2014).
- alcohol: alcohol. Porcentaje de alcohol presente en el vino. De media, un vino está compuesto por un 85% de agua, entre un 10% y un 15% de alcohol y, en menor medida, por el resto de sustancias nutritivas, minerales y esenciales (Vinetur, 2015).

Y la variable de salida es la siguiente:

- quality: calidad del vino, en un rango de 3 a 8 puntos.

Con el dataset se pretende entender la calidad de un vino como resultado del análisis de las variables fisicoquímicas de entrada arriba indicadas. La respuesta es trascendente ya que, si se alcanza un modelo capaz de predecir la calidad de un vino, en función de parámetros fácilmente observables y medibles desde un punto de vista técnico, podrían modificarse las producciones con el fin de conseguir mejores caldos,

lo que redundaría en mayores ventas y, consecuentemente, en un mayor beneficio. La industria del vino tiene gran importancia en países de dilatada tradición vitivinícola, como España o Portugal. En el caso español, de la industria del vino dependen más de 400.000 puestos de trabajo, lo que supone un 2,4% del total del empleo y, como sector, genera 24.000 millones de euros anuales (Vinetur, 2020).

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

Debido a que el dataset era un dataset de muestra no ha necesitado de la integración de los datos.

Como punto de partida, no se descarta ninguna de las 11 variables de entrada.

Se crea una nueva variable dicotómica *quality_cat* a partir de la variable *quality*, con los siguientes rangos:

- [3-6]: "0" [1]
- [7-8]: "1" [2]

```
wine[wine$quality <= 6, "quality_cat"] <- 0
wine[wine$quality > 6, "quality_cat"] <- 1
wine$quality_cat <- factor(wine$quality_cat)
wine$quality_cat <- relevel(wine$quality_cat, ref="1")
```

Para su uso posterior se creará en el punto 4 una segunda variable dicotómica partir del azúcar residual, clasificando los vinos en:

- Seco: < 4
- Semiseco: [4-30]

3. Limpieza de los datos

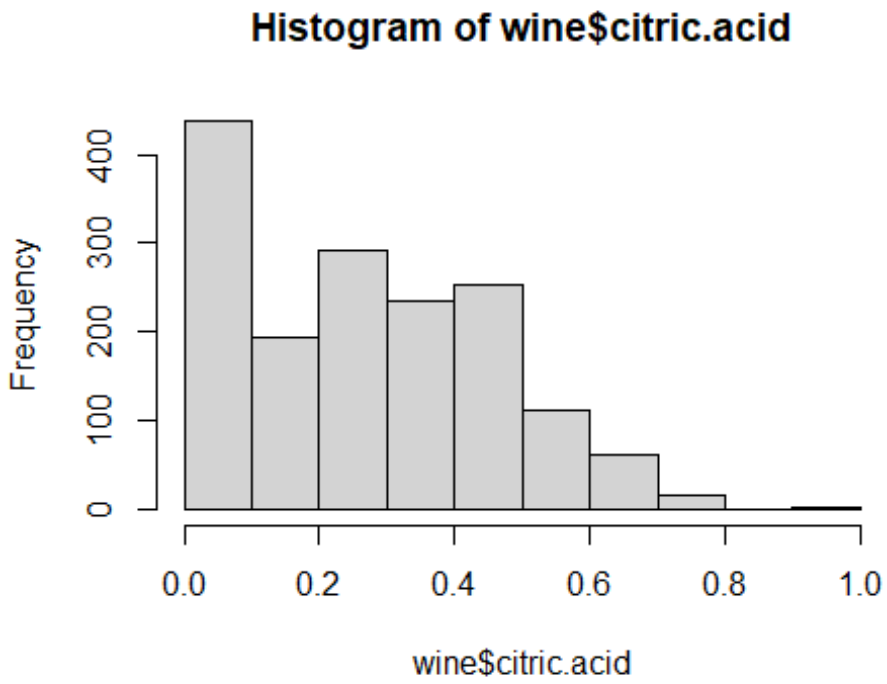
3.1. Ceros o elementos vacíos

No existen observaciones con elementos vacíos.

```
# Valores perdidos
print(paste("Registros NA:", sum(is.na(wine))))

## [1] "Registros NA: 0"
```

Sólo la variable *citric acid* contiene valores a cero [0], en un total de 132 observaciones.

```
# Observaciones con [citric acid]==0
print(paste("Registros [citric acid]==0:", sum(wine$citric.acid==0)))

## [1] "Registros [citric acid]==0: 132"

hist(wine$citric.acid)
```



Teniendo en cuenta que, en dicha variable, el primer valor con datos distintos a cero es 0,01, y que la variable presenta valores en el rango [0-1], se dan por válidas todas las observaciones.

3.2. Identificación y tratamiento de valores extremos

Se obtiene un valor resumen de los datos.

```
# Valores resumen de cada tipo de variable
summary(wine1)
```

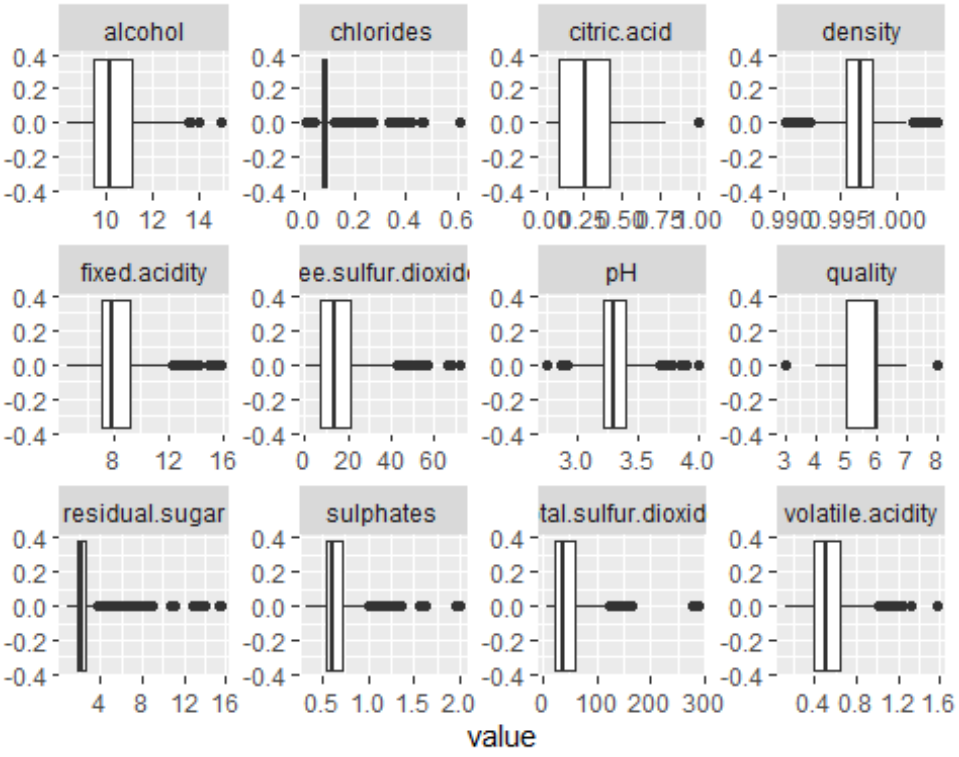
```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :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
```

Para identificar los valores extremos primero de todo graficaremos en un diagrama de caja todas las variables.

```
library(purrr)
# install.packages("tidyr")
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
wine %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_boxplot()
```



Vemos que el dataset puede presentar outliers a nivel visual en las siguientes variables. Como no tenemos suficiente conociemiento del mundo del vino para identificar los outliers, nos basaremos en los valores que estén 1,5 por encima o por debajo del IQR.

```
is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - 1.5 * IQR(x) |
           x > quantile(x, 0.75) + 1.5 * IQR(x))
}
outlier <- data.frame(variable = character(),
                      sum_outliers = integer(),
                      stringsAsFactors=FALSE)
for (j in 1:(length(wine)-1)){
  variable <- colnames(wine[j])
  for (i in wine1[j]){
    sum_outliers <- sum(is_outlier(i))
  }
  row <- data.frame(variable,sum_outliers)
  outlier <- rbind(outlier, row)
}
outlier
```

```
##                variable sum_outliers
## 1         fixed.acidity           49
## 2      volatile.acidity           19
## 3           citric.acid            1
## 4        residual.sugar          155
## 5             chlorides          112
```

```
## 6   free.sulfur.dioxide           30
## 7  total.sulfur.dioxide           55
## 8               density           45
## 9                    pH           35
## 10            sulphates           59
## 11              alcohol           13
## 12              quality           28
```

Tal y como vemos en la tabla anteriormente mostrada vemos que nuestros datos pueden presentar ouliers.

Sin embargo entendemos que si las mediciones de las variables se han hecho de la misma y utilizando los mismos medidores seguramente los outliers se deberán a que hay vinos con características únicas, por lo que solo consideraremos como potenciales outliers aquellos valores de las variables que tengan más de un 4% de valores extremos.

```
for (i in 1:nrow(outlier)){
  if (outlier[i,2]/nrow(wine) * 100 >= 4){
    print(paste(outlier[i,1],
                '=',
                round(outlier[i,2]/nrow(wine) * 100, digits = 2),
                '%'))
  }
}
```

```
## [1] "residual.sugar = 9.69 %"
## [1] "chlorides = 7 %"
```

En este caso serán en las variables: residual.sugar y chlorides, sin embargo despues de consultar en internet los posibles valores de estas dos variables vemos que pueden ser coherentes y se debe más a que la muestra de vinos es muy distinta entre sí. Por ejemplo el residual sugar puede presentar todos los valores de la muestra dependiendo del tipo de vino. No son valores extremos sino valores que se presentan en menor medida.

Despues de este apartado ya tendríamos listo el Dataset para analizarlo.

4. Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar

4.2. Comprobación de la normalidad y homogeneidad de la varianza

Primero de todo realizaremos el estudio de normalidad de las variables mediante el test de normalidad de Shapiro-Wilk por cada uno de los atributos. Plantearemos el siguiente test de hipotesis: H0 la muestra sigue una distribució Normal H1 la muestra NO sigue una destribución Normal.

```
c<-names(wine1)
for (i in 1:ncol(wine1))
{
  cat("Atribut '",c[i],"', ", sep = '')
  pvalor <- shapiro.test(wine[, i])[["p.value"]]
  cat("p-valor '", pvalor,"'\n", sep = '')

}
```

```
## Atribut 'fixed.acidity', p-valor '1.525012e-24'
## Atribut 'volatile.acidity', p-valor '2.692935e-16'
## Atribut 'citric.acid', p-valor '1.021932e-21'
## Atribut 'residual.sugar', p-valor '1.020162e-52'
## Atribut 'chlorides', p-valor '1.179056e-55'
## Atribut 'free.sulfur.dioxide', p-valor '7.694597e-31'
## Atribut 'total.sulfur.dioxide', p-valor '3.573451e-34'
## Atribut 'density', p-valor '1.936053e-08'
## Atribut 'pH', p-valor '1.712237e-06'
## Atribut 'sulphates', p-valor '5.82314e-38'
## Atribut 'alcohol', p-valor '6.644057e-27'
## Atribut 'quality', p-valor '9.515085e-36'
```

Podemos observar por el valor p y las hipotesis que hemos planteado que los valores de los atributos del dataset NO siguen una distribución normal. No obstante cuando el número de observaciónes es grande y debido al teorema del límite central, se podrán utilizar pruebas paramétricas. Así las variables se podrían aproximar a una distribución normal de media 0 y de desviación estándard 1.

Homoscedasticitat: H0: Las variancias poblacionales son iguales, homoscedasticidad H1: Las variancias poblacionales son diferentes, heteroscedasticidad

```
library("car")

leveneTest(wine$alcohol,wine$quality)
```

```
## Warning in leveneTest.default(wine$alcohol, wine$quality): wine$qualit
y coerced
## to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  24.226 < 2.2e-16 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como que le pvalue es<que el nivel de significación podemos rechazar la hipotesiss nula H0, por lo tanto podemos afirmar que hay heteroscedasticitat entre estas dos variables.

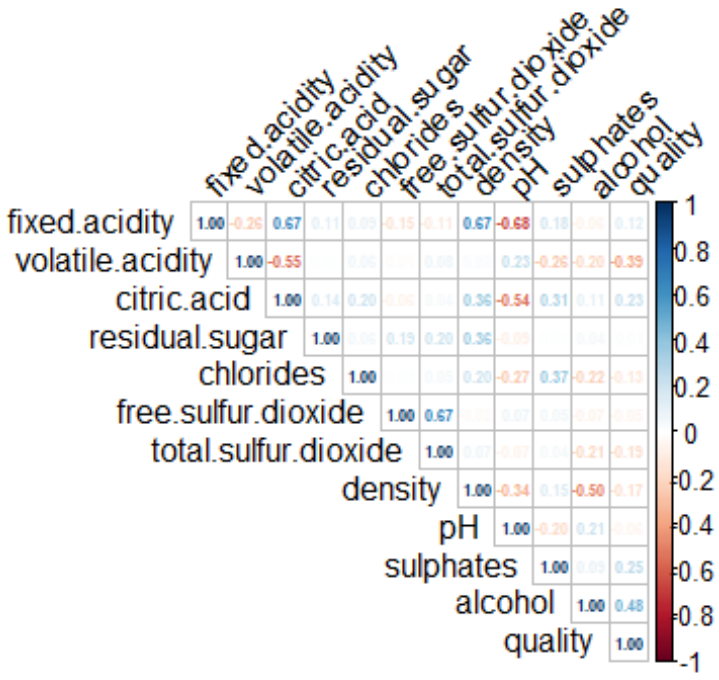
4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos

4.3.1 Queremos responder a la pregunta, qué variables tienen una correlación superior con la calidad del vino?

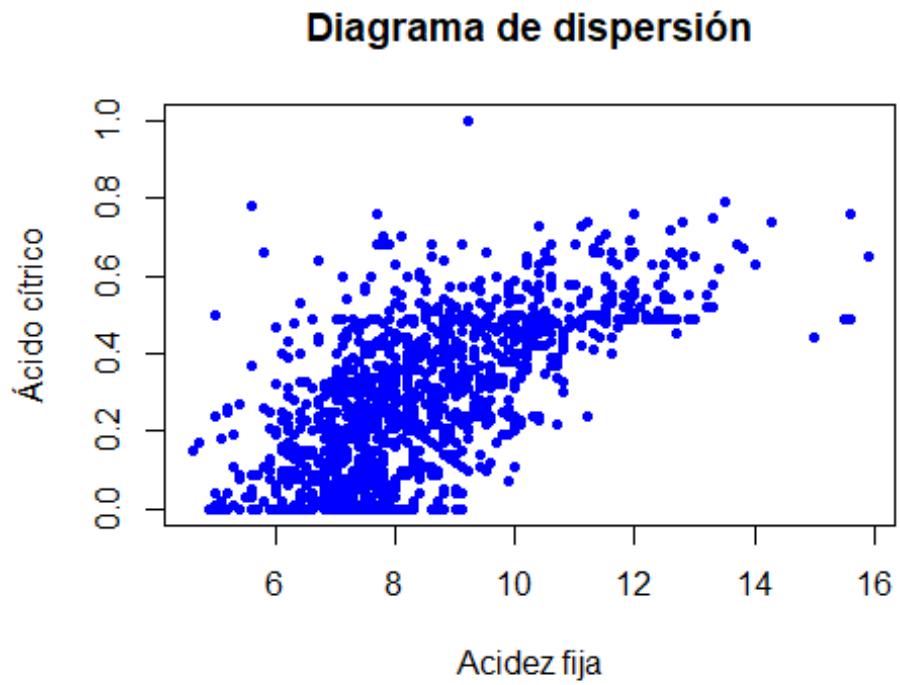
Para contestar a esta pregunta procederemos a realizar un análisis de correlación entre las distintas variables.

En primer lugar se analizarán las correlaciones entre los distintos atributos del dataset

```
library(corrplot)

## corrplot 0.91 loaded

correlation<-cor(wine1)
corrplot(correlation,type="upper", tl.srt=45, number.cex = 0.5, tl.col =
"black", method="number")
```




```
plot(wine$fixed.acidity, wine$citric.acid, pch=20, col="blue", xlab="Acidez fija",
ylab="Ácido cítrico", main="Diagrama de dispersión")
```



Se observa que la variable alcohol es la que presenta una correlación más alta con la calidad del vino. Tambien nos gustaría destactar que existe una fuerte relación entre la acidez fija y el ácido cítrico (presente en la misma), con una correlación de 0,67, e incluso entre este último y la acidez volátil, con una correlación de 0,55.

4.3.2 Segunda prueba estadística:

Para nuestra segunda prueba estadística intentaremos saber si un vino tiene mayor puntuación en función si es seco o semiseco. Primero de todo crearemos una nueva variable que nos indicará si un vino es seco o semiseco siguiendo los siguientes criterios:

- Seco: < 4
- Semiseco: [4-30] Queremos responder a la pregunta de:

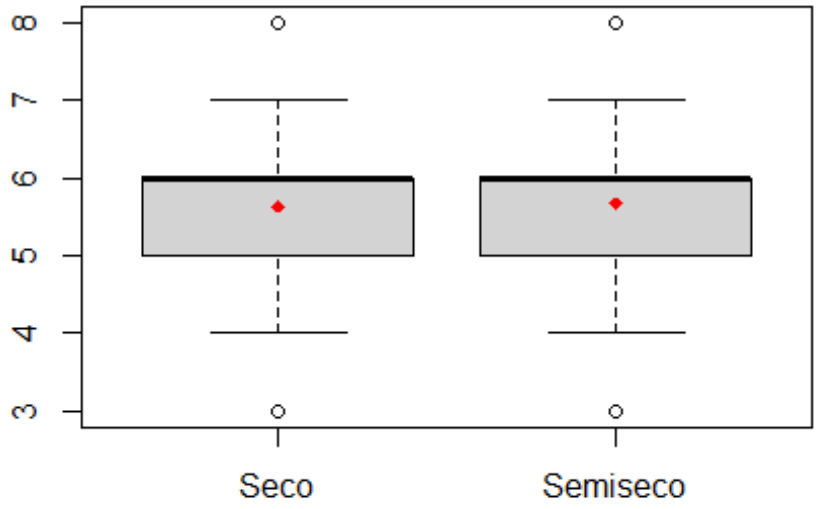
¿La calidad del vino varia en caso de ser seco o semiseco?

H0 : mu1=mu2

H1 : mu1<>mu2

Debido a la teoria del límite central podemos asumir normalidad debido a que n>30

```
wine[wine$residual.sugar  < 4, "seco"] <- "Seco"
wine[wine$residual.sugar  >= 4, "seco"] <- "Semiseco"

wine_seco_calidad<-wine[wine$seco=="Seco",]$quality
wine_semiseco_calidad<-wine[wine$seco=="Semiseco",]$quality

t.test(wine_seco_calidad,wine_semiseco_calidad, alternative="two.sided")

##
##  Welch Two Sample t-test
##
## data:  wine_seco_calidad and wine_semiseco_calidad
## t = -0.52605, df = 153.02, p-value = 0.5996
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2102312  0.1218149
## sample estimates:
## mean of x mean of y
##  5.632262  5.676471

boxplot(wine_seco_calidad,wine_semiseco_calidad,names=c("Seco","Semiseco"
))
medias <- c(mean(wine_seco_calidad),mean(wine_semiseco_calidad))
points(medias,pch=18,col="red")
```



Debido a que el pvalue es superior a 0,05 no podemos rechazar la hipotesis nula, por lo que podríamos afirmar que el hecho de ser seco, semiseco no afecta a la calidad percibida por el usuario.

4.3.3 Tercer test estadístico:

Para nuestra última prueba estadística realizaremos tres modelos predecir la calidad del vino en funcion de las variables del dataset.

Usaremos los siguientes métodos:

- Regresión lineal
- RandomForest
- LogisticRegresion:

Preparamos los subsets de train y test para los algoritmos RandomForest y LogisticRegresion.

```
set.seed(1234)
samp <- sample(nrow(wine), 0.6 * nrow(wine))
train <- wine[samp, ]
test <- wine[-samp, ]
```

4.3.3.1 Regresión lineal

```
model_lm <- lm(quality ~. , data=wine_raw)
summary(model_lm)
```

```
##
## Call:
## lm(formula = quality ~ ., data = wine_raw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.197e+01  2.119e+01   1.036   0.3002
## fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
## volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
## residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
## chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
## free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
## total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## density              -1.788e+01  2.163e+01  -0.827   0.4086
## pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
## sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
## alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

Vemos que la precisión de la regresión lineal no es para nada ideal, la R adjusted es de 0,36.

Por este motivo vamos a realizar dos modelos a partir de los datos para predecir la calidad del vino usando los algoritmos RandomForest y una regresion logística.

4.3.3.2 Random Forest

Primero de todo separaremos nuestros datos entre el dataset de prueba y el de test. Dividiremos el dataset entre 60% prueba y 40% test. Acto seguido construiremos el modelo, con el ntree por defecto, 500.

Acto seguido mostraremos la matriz que nos enseña la predicción vs los valores reales (matriz de confusión) y calcularemos el accuracy del modelo.

```
model <- randomForest(quality_cat ~ fixed.acidity + volatile.acidity + ci
tric.acid +  chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
density + pH + sulphates + alcohol, data = train, ntree=1500)
model
```

```
##
## Call:
##  randomForest(formula = quality_cat ~ fixed.acidity + volatile.acidity
+      citric.acid + chlorides + free.sulfur.dioxide + total.sulfur.dioxi
de +      density + pH + sulphates + alcohol, data = train, ntree = 1500)
##                Type of random forest: classification
##                      Number of trees: 1500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 8.24%
## Confusion matrix:
##    1   0 class.error
## 1 70  63  0.47368421
## 0 16 810  0.01937046
```

```
tst_pred <- predict(model, newdata = test, type = "response")
tst_tab <- table(predicted = tst_pred, actual = test$quality_cat)
tst_tab
```

```
##           actual
## predicted   1   0
##         1  33  14
##         0  51 542
```

```
pred <- predict(model, newdata = test)
sum(diag(tst_tab))/length(train$quality_cat)
```

```
## [1] 0.5995829
```

Podemos apreciar que el modelo cuando lo utilizamos en el set de test nos da un accuracy del 0,59 más alto que la regresión lineal pero no sería suficiente para aceptarlo como modelo nos permita predecir con un alto grado de acierto.

4.3.3.3 Logistic Regresion

```
model_glm <- glm(quality_cat ~ fixed.acidity + volatile.acidity + citric.
acid +  chlorides + free.sulfur.dioxide + total.sulfur.dioxide +    densi
ty + pH + sulphates + alcohol, data = train, family=binomial(link = "logi
t"))
model_glm
```

```
##
## Call:  glm(formula = quality_cat ~ fixed.acidity + volatile.acidity +
##     citric.acid + chlorides + free.sulfur.dioxide + total.sulfur.dioxi
de +
##     density + pH + sulphates + alcohol, family = binomial(link = "logi
t"),
##     data = train)
##
## Coefficients:
##          (Intercept)         fixed.acidity      volatile.acidity
##              67.87335              -0.07601               2.52320
```

```
##          citric.acid             chlorides   free.sulfur.dioxide
##              0.26077              12.38274              -0.01147
## total.sulfur.dioxide               density                    pH
##              0.02732             -56.39700               0.62396
##            sulphates               alcohol
##              -4.36459              -1.04095
##
## Degrees of Freedom: 958 Total (i.e. Null);  948 Residual
## Null Deviance:       772.1
## Residual Deviance: 517.3     AIC: 539.3
```

```
tst_pred <- ifelse(predict(model_glm, newdata = test, type = "response")
> 0.5, "Good Wine", "Bad Wine")
tst_tab <- table(predicted = tst_pred, actual = test$quality_cat)
tst_tab
```

```
##            actual
## predicted     1   0
##   Bad Wine   24  17
##   Good Wine  60 539
```

```
sum(diag(tst_tab))/length(train$quality_cat)
```

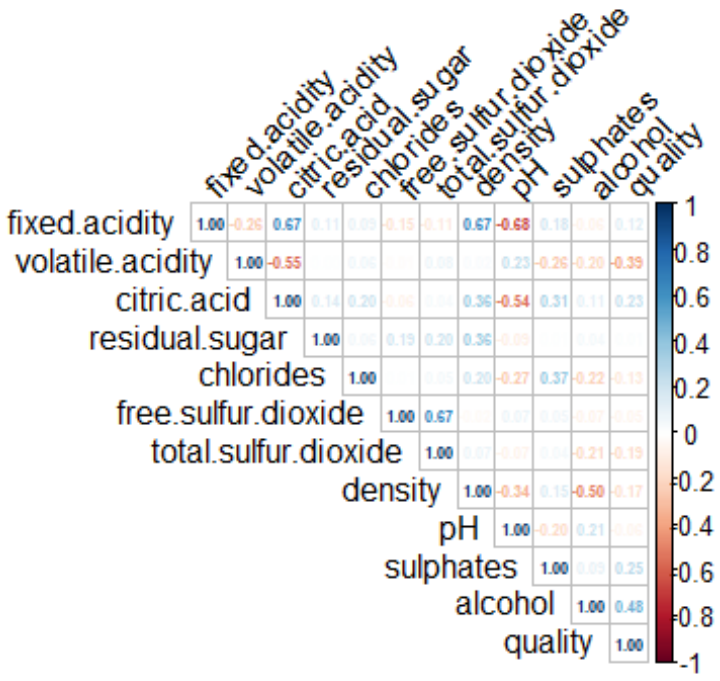
```
## [1] 0.5870699
```

Al igual que usando el algoritmo randomForest no vemos que sea un modelo lo suficientemente bueno para aceptarlo Vemos que la accuracy del modelo random forest es de aproximadamente un 60%, mientras que la de la regresión logística es de 59%, por lo que no podríamos categorizar a ambos de buen modelo, sin embargo el error que presentan es menor que el modelo de regresión lineal.

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

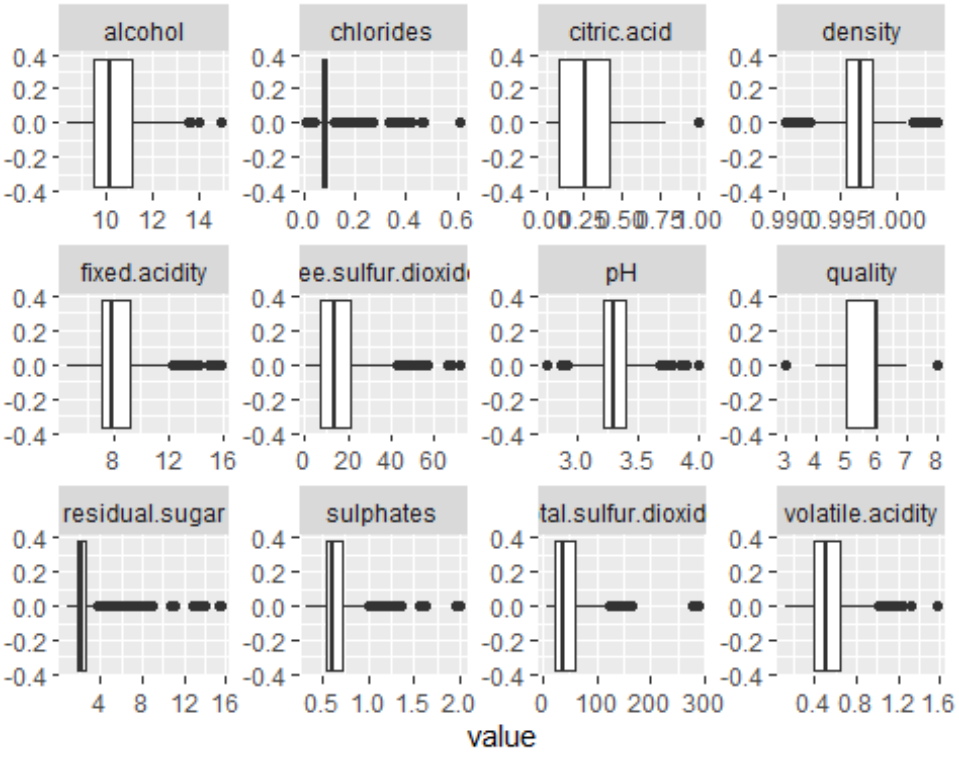
Acto seguido vamos a representar las gráficas y tablas más relevantes que hemos ido viendo a lo largo de la práctica y alguna más que consideramos de interés: Matriz de correlación:

```
correlation<-cor(wine1)
corrplot(correlation,type="upper", tl.srt=45, number.cex = 0.5, tl.col =
"black", method="number")
```



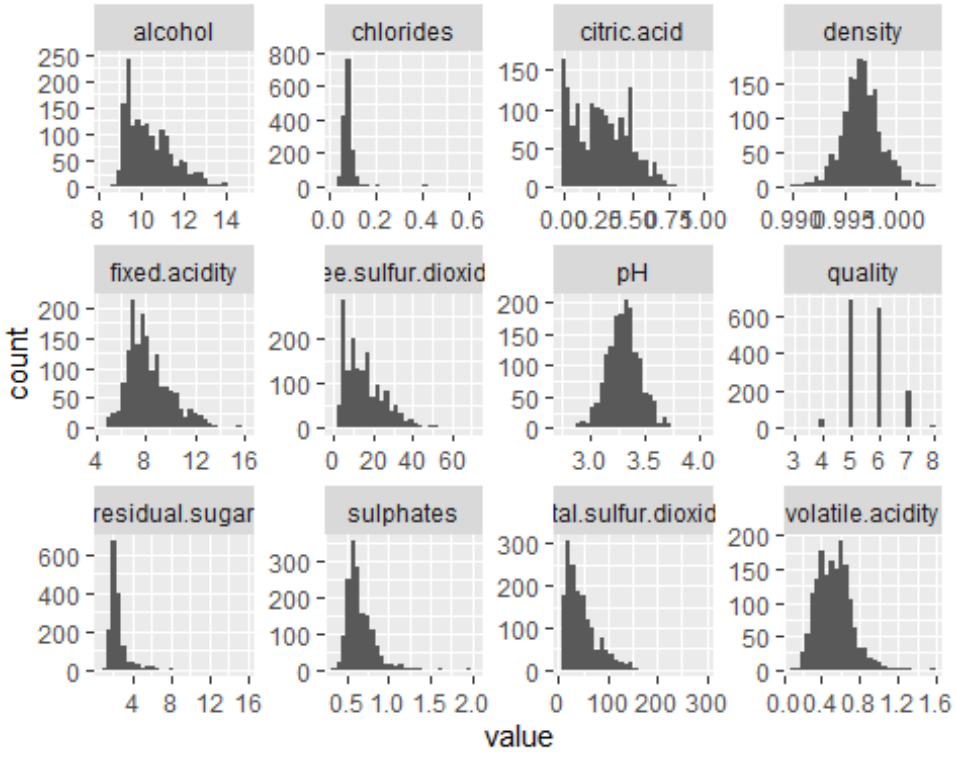
Boxplot de las variables :

```
wine %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_boxplot()
```



Histograma de las variables:

```
wine_raw %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()

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



6. Resolución del problema

Se exporta el dataset tratado, con el nombre *winequality-red-FINAL.csv*.

```
# Exportación del dataset
write.csv(wine, file = "winequality-red-FINAL.csv")
```

Como conclusión si analizamos el dataset vemos que a nivel de datos es un dataset muy pulido y no hemos tenido que realizar grandes tareas de limpieza de datos; los modelos que se han desarrollado no son adecuados para predecir si un vino será bueno o malo.

El problema del dataset en nuestra opinión, peca en origen de cierta ingenuidad. Reducido a la máxima expresión, se pretende conocer la calidad de un vino en base a 7 parámetros principales: acidez, azúcar, salinidad, sulfitos, densidad, sulfatos y alcohol. Pero el vino es mucho más que esto y la complejidad de la técnica enológica, además de la subjetividad, hacen que la definición de calidad se escape de una simple mezcla y concentración de elementos esenciales. Sin embargo, el dataset sí puede ser un buen punto de partida para ayudar a determinar qué características -entre otras muchas- que aparecen recurrentemente en los vinos de mayor prestigio y reconocimiento.

7. Código

El código con el que se ha realizado la limpieza, análisis y representación de los datos conforma parte del presente archivo y figura adjunto en cada uno de los apartados previos desarrollados.

_____