PREPROCESADO DE DATOS

Marc Román Porras

Mayo 2021

Table of Contents

El preprocesamiento de datos previo al análisis estadístico de estos es esencial. En muchos casos, las fuentes de información no son coherentes, completas y veraces, por lo que se hace imprescindible preparar los datos. No debemos pasarnos por alto este paso previo al análisis, ya que saltar o reducir esta fase porque puede tener consecuencias muy graves en los resultados (sesgos, conclusiones equivocadas, …).

A continuación se disponen los apartados y subapartados de actividad en el desarrollo de la asignatura Tipología y ciclo de vida de los datos con la finalidad de preprocesar los datos del fichero "winequality-red.CSV" para posteriormente realizar una breve estadística descriptiva, modelado y si aplica, algún análisis de componentes principales (ACP) con algunas variables cuantitativas.

** COMENTARIO: LA INSTALACIÓN DE LAS LIBRERÍAS QUE SE HAN USADO A LO LARGO DE LA PEC Y QUE PREVIAMENTE NO ESTABAN INSTALADAS, (INSTALL.PACKAGES()) SE HAN INTRODUCIDO ALLÍ DONDE DEBEN IR EN LOS DIFERENTES FRAGMENTOS DE R COMENTADOS YA QUE SINO LA GENERACIÓN DEL .HTML NO SE PUEDE LLEVAR A CABO)

1. DESCRIPCIÓN DEL DATASET ¿Por qué es importante y qué pregunta/problema pretende responder?

Se dice que la enología se caracteriza con el conjunto de conocimientos y técnicas relativos a los procesos de elaboración y crianza de vinos. Casi considerada una ciencia, si más no un arte por todo lo que el vino aporta a nuestra cultura.

Los enólogos, dichos como las personas que ponen en práctica el conocimiento vitivinícola, afirman que la acidez es una de las grandes virtudes del vino. De hecho en varias ocasiones hemos podido escuchar que un vino sin acidez, "es un vino muerto"… Pero, ¿Por qué es tan importante la acidez en un vino?

Bien, la respuesta a esta pregunta se fundamenta en la procedencia de la uva, de una fruta, siendo los vinos ácidos por naturaleza. Asimismo, la acidez es imprescindible en el vino, tanto desde el punto de vista de su conservación, como de sus propiedades organolépticas y, en última instancia, su degustación. En este punto podemos formularnos otra pregunta;

¿de dónde procede exactamente la acidez, cuántos tipos de ácidos podemos encontrarnos en un vino y qué aporta cada uno de ellos?

La respuesta a esta pregunta, puede ser extensa, puesto que a pesar del gran componente que aporta en acidez la uva, no toda procede de esta. Durante la fermentación, la conservación y el envejecimiento se pueden catabolizar o anabolizar nuevos compuestos ácidos, que además son partícipes en la formación de nuevas sustancias (polifenoles o compuestos cromáticos).

En definitiva podemos afirmar que los ácidos tartárico, málico y cítrico, proceden principalmente de la uva, mientras que de la fermentación, conservación y el envejecimiento los más frecuentes son el láctico, succínico y acético.

Pero, en este punto, ¿Por qué nos disponemos a hablar de acidez? La pregunta anterior tiene su respuesta en cuanto a la calidad del vino. Un vino debe su calidad en gran parte a su acidez, sin embargo los parámetros totales que influyen en la calidad de un vino son el ph, la acidez total, la acidez volátil y el ácido málico.

Es en este punto, en base a los conceptos anteriores de calidad, ácidos y ph, donde entra nuestro dataset winequality-red.csv. A partir de las definiciones de los ácidos comentados, así como del ph, vamos a llevar a cabo un análisis descriptivo y estadístico a lo largo de la actividad relacionando en términos generales calidad y acidez.

Antes, debemos tener claro que define cada uno de los conceptos anteriores:

ph: El ph es precisamente el coeficiente que indica el grado de acidez o basicidad de una solución acuosa. En términos vitivinícolas, no solo afecta a la acidez, sino también al color y conservación del vino. Sus valores típicos están entre 3.1 y 3.9.

Acidez total: esta medida refleja la suma de todos los ácidos del vino, los valores normales oscilan entre 4,50 y 6,00 gr/l. (Volátil y no volátil).

Acidez volátil: esta medida calcula el ácido acético de un vino. Los valores normales van de 0,30 a 0,60 g/l. Habitualmente escuchamos que un vino “está picado”, este hecho se designa cuando presenta el vino una acidez volátil por encima de 1 g/l y aromas que recuerdan al vinagre y al barniz.

Ácido málico: esta medida es la responsable de la sensación en boca ácida, de frescor. Los vinos tintos, que han realizado la fermentación maloláctica para eliminar este ácido málico, presentan menos acidez de este tipo frente a blancos y rosados.

Ácido cítrico: es un ácido poco abundante en la uva, de 150 a 300 mg/litro de mosto. Después es fermentado por las bacterias lácticas y desaparece.

Una vez identificados los elementos que dotan de calidad a un vino, presentamos el dataset, disponiéndose a extraer los parámetros anteriores, mediante un análisis estadístico y descriptivo.

2.INTEGRACIÓN Y SELECCIÓN DE LOS DATOS A ANALIZAR

En una primera instancia, el primer paso a llevar a cabo es la lectura del archivo .csv y la muestra de las 6 primeras filas de este:

```
# Leo y almaceno la totalidad del fichero de vinos:
base.datos <- read.csv("winequality-red.csv", header=TRUE)
# Para verificar como R ha leído dicho archivo, podemos acceder a sus 5
primeras/últimas filas con head() y/o tail() y observar si la información es
correcta.
head(base.datos)
```

```
##   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
```

Una vez disponemos del archivo y de las capacidades de manipulado de este, antes de proceder con el análisis y las consideraciones vitivinícolas, el primer paso a llevar a cabo es la observación de los datos de los que disponemos. Para ello, podemos hacer uso de diversas cláusulas que R nos proporciona:

```
# Muestra de los datos:
str(base.datos)
```

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

Como vemos, con str() observamos las clases de los datos, su nombre (vemos a qué se corresponde cada columna) y un ejemplo en cuanto a los datos en sí.

A continuación, y siguiendo con la línea de lo expuesto en el apartado introductorio de esta actividad, vamos a proceder a reducir el tamaño inicial de la muestra de datos de la que disponemos tras la lectura del dataset. En este punto, nos quedaremos con aquellas variables que describen la acidez del vino y su calidad, y que por otro lado, pueden contrastar con dicha acidez (azúcar y alcohol), afectando a su calidad y gusto, para los posteriores análisis que llevemos a cabo. A continuación nos quedamos por tanto con los siguientes campos en subconjunto:

fixed acidity: la mayoría de los ácidos involucrados con el vino o fijos o no volátiles (no se evaporan fácilmente). [numeric]. Rango 4.6-15.9 gr/L.

volatile acidity: la cantidad de ácido acético en el vino, que en niveles demasiado altos puede provocar un sabor desagradable a vinagre. [numeric]. Rango 0.12-1.58 gr/L.

citric acid: Encontrado en pequeñas cantidades, el ácido cítrico puede agregar 'frescura' y sabor a los vinos. [numeric]. Rango 0-1 gr/L.

residual sugar: la cantidad de azúcar que queda después de que se detiene la fermentación, es raro encontrar vinos con menos de 1 gramo / litro… [numeric]. Rango 0.9-15.5 gr/L.

alcohol: porcentage de alcohol presente en el vino. [numeric]. Rango entre 8.4 y 14.9 %.

pH: describe qué tan ácido o básico es un vino en una escala de 0 (muy ácido) a 14 (muy básico). [numeric]. Rango entre 2.74-4.01.

quality: variable de salida basada en datos de sensibilidad del vino. [integer]. Rango entre 0 y 10 en escala.

En este punto, nos deshacemos de los demás campos disponiendo únicamente de los que nos interesan en nuestra muestra para proceder con el preprocesado y análisis:

```
#install.packages("dplyr") # Lo instalo si no lo tengo
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

base.datos <- base.datos %>%
select(c(quality,fixed.acidity, volatile.acidity, citric.acid,
residual.sugar, alcohol, pH))
head(base.datos, 10)

##    quality fixed.acidity volatile.acidity citric.acid residual.sugar
alcohol
## 1        5           7.4             0.70        0.00            1.9
9.4
## 2        5           7.8             0.88        0.00            2.6
9.8
## 3        5           7.8             0.76        0.04            2.3
9.8
## 4        6          11.2             0.28        0.56            1.9
9.8
## 5        5           7.4             0.70        0.00            1.9
9.4
## 6        5           7.4             0.66        0.00            1.8
9.4
## 7        5           7.9             0.60        0.06            1.6
9.4
## 8        7           7.3             0.65        0.00            1.2
10.0
## 9        7           7.8             0.58        0.02            2.0
9.5
## 10       5           7.5             0.50        0.36            6.1
10.5
##      pH
## 1  3.51
## 2  3.20
## 3  3.26
## 4  3.16
## 5  3.51
## 6  3.51
## 7  3.30
## 8  3.39
## 9  3.36
## 10 3.35
```

Una vez disponemos del subconjunto debidamente compuesto por las diferentes columnas, procedemos con la limpieza y el preprocesamiento de los datos.

3. LIMPIEZA DE DATOS

A lo largo de este apartado, generalmente vamos a comprobar para cada uno de los campos, que sus ceros, en caso de contener, se encuentren mínimamente justificados, que si disponen de valores missing, null o N/A, éstos sean tratados/sustituidos y/o aproximados de acuerdo a su naturaleza y de la homogeneización del conjunto de datos en sí en cuanto a valores y decimales.

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

Primeramente, nos disponemos a observar si en nuestro conjunto total de datos de disponen campos missing. para ello llevamos a cabo el siguiente procedimiento:

```
# Visualizo si existe algún valor null en la totalidad de variables:
# install.packages("dplyr") # librería comentada. Descomentar si no se
dispone de ella.
library(dplyr)
base.datos %>%
select(everything()) %>%
summarise_all(funs(sum(is.na(.))))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
##   quality fixed.acidity volatile.acidity citric.acid residual.sugar
alcohol pH
## 1       0             0                0           0              0
0  0
```

Como output, el fragmento de código anterior nos proporciona un dataFrame con la contención de un entero que nos muestra el total de valores null de las diferentes componentes en caso de que estas dispongan. Como vemos, ninguno de nuestros campos presenta missing values.

Acto seguido, nos disponemos a observar la cantidad de valores “0” de los que dispone nuestro dataset columna por columna:

```
# Número total de ceros en Quality:
sum(base.datos$quality == 0)
```

```
## [1] 0
```

```
# Número total de ceros en Fixed Acidity (aacidez no volátil):
sum(base.datos$fixed.acidity == 0)
```

```
## [1] 0
```

```
# Número total de ceros acidez volátil:
sum(base.datos$volatile.acidity == 0)
```

```
## [1] 0
```

```
# Número total de ceros en ácido cítrico:
sum(base.datos$citric.acid == 0)
```

```
## [1] 132
```

```
# Número total de ceros en Azúcar residual:
sum(base.datos$residual.sugar == 0)
```

```
## [1] 0
```

```
# Número total de ceros en Alcohol:
sum(base.datos$alcohol == 0)
```

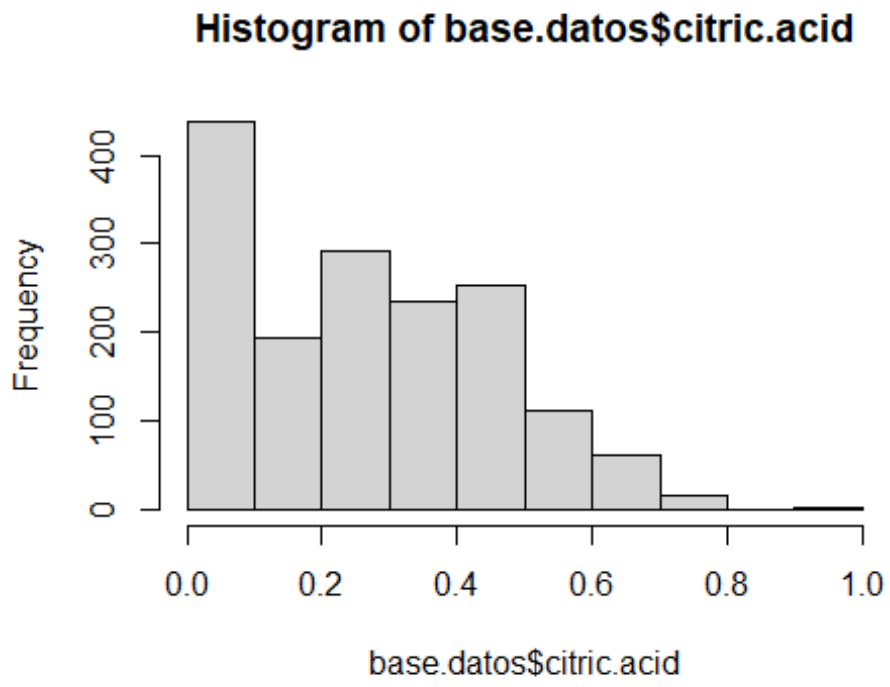
```
## [1] 0
```

```
# Número total de ceros en pH:
sum(base.datos$pH == 0)
```

```
## [1] 0
```

Como vemos, únicamente la variable de ácido cítrico dispone de valores 0, (132 en total) por lo que centraremos en este punto especial atención a observar cómo se distribuye la variable y dar pié a justificar dichos ceros. Para ello, primeramente mostraremos cómo se distribuye en frecuencias dicha variable:

```
# Distribución de frecuencias de la variable ácido cítrico:
hist(base.datos$citric.acid)
```



Como vemos, el histograma aglutina buena parte de sus valores en torno al cero, definiendo una cola hacia la derecha… En este punto la pregunta que nos hacemos es la siguiente; ¿Es real/posible el hecho de disponer de vinos con ácido cítrico 0? ¿Qué supondría este hecho?

En general la presencia de ácido cítrico en la uva es poco abundante. Su concentración oscila entre 0,1 y 1 g/l. En general el ácido cítrico es el encargado de aportar al vino frescura, sensaciones agradables, frutales y aromáticas.

En este punto, conociendo dicha información, como vemos tiene sentido que dispongamos de vino con baja o nula concentración de ácido cítrico, puesto que en dichos casos se trataría de un vino no afrutado y algo más seco. Por tanto, mantenemos dichas consideraciones y valores nulos en dicha variable y procedemos con nuestro preprocesamiento.

El último punto de este apartado va a consistir en la homogeneización de los decimales de la totalidad de nuestras variables. Como hemos podido observar anteriormente, las columnas de ácido volátil, ácido cítrico y pH disponen de 2 decimales mientras que las demás variables tan solo disponen de un valor decimal. Cómo no vamos a filtrar ni sacar conclusiones a nivel centesimal, vamos a redondear dichos valores a las décimas. para ello hemos implementado la siguiente función:

```
# Homogenizo la totalidad del dataset:
base.datos <- round(base.datos, 1)

base.datos$volatile.acidity <- round(base.datos$volatile.acidity, 1)
```

```r
# Muestro el conjunto de datos al completo:
head(base.datos)
```

```
##   quality fixed.acidity volatile.acidity citric.acid residual.sugar
alcohol  pH
## 1       5           7.4              0.7         0.0            1.9
9.4 3.5
## 2       5           7.8              0.9         0.0            2.6
9.8 3.2
## 3       5           7.8              0.8         0.0            2.3
9.8 3.3
## 4       6          11.2              0.3         0.6            1.9
9.8 3.2
## 5       5           7.4              0.7         0.0            1.9
9.4 3.5
## 6       5           7.4              0.7         0.0            1.8
9.4 3.5
```

Una vez disponemos de un conjunto homogéneo, sin missings y sin ceros, vamos a estudiar dato por dato (columna por columna) la presencia de valores extremos:

3.2 Identificación y tratamiento de valores extremos:

Los valores extremos son peligrosos en el estudio estadístico ya que pueden sesgar significativamente el análisis. Aún y así, es esencial entender y observar de qué tipo de valores extremos disponemos en función de la variable que estamos considerando en análisis. A continuación, analizamos variable por variable y procedemos de un tratamiento de extremos para cada caso particular:

VARIABLE CALIDAD

Representación del boxplot y de los valores outliers:

```r
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$quality,main="Wine Quality")
```


```
# Valor de los Outliers:
boxplot.stats(base.datos$quality)$out
```

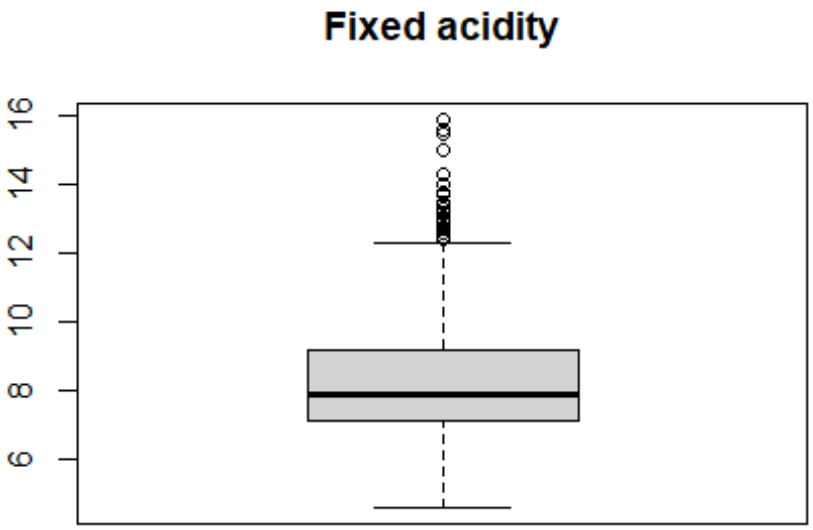
```
##  [1] 8 8 8 8 8 3 8 8 8 3 8 3 8 3 3 8 8 8 8 3 3 8 8 3 3 3 8
```

En este caso, para calidad, hemos obtenido múltiples valores de 3 y 8, pero teniendo en cuenta que estamos midiendo en una escala del 0 al 10 la calidad de un vino, estos valores realmente no son extremos en relación a la variable que estamos midiendo ya que están totalmente dentro de rango y no tendría sentido anular estos valores… Por ello, a pesar de que el 3 y el 8 se representan como valores extremos en la distribución de la variable quality, no vamos a alterar dicha distribución y la dejamos como está. Para Calidad NO haremos tratamiento los Outliers!!!

VARIABLE FIXED ACIDITY

Representación del boxplot y de los valores outliers:

```
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$fixed.acidity,main="Fixed acidity")
```




```
# Valor de los Outliers:
boxplot.stats(base.datos$fixed.acidity)$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
```

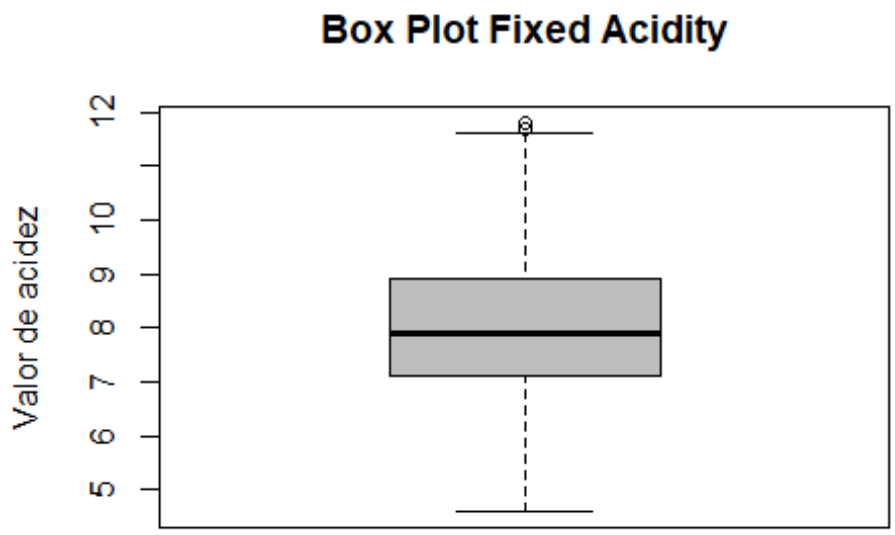
Para Fixed acidity, si que tiene sentido tratar los valores outliers que aparecen, ya que en este caso dicha variable no se rige por ningún rango de valores en los que debe estar contenida como era el caso de calidad. Por ese motivo, los valores que aparecen están totalmente alejados de los percentiles de distribución que muestra el boxplot, por lo que en este caso, eliminaremos dichos valores extremos de nuestra muestra, mediante la introducción de valores missing allí donde se den.

Posteriormente, identificamos dichos valores NA's introducidos y los sustituiremos por el valor medio de la distribución de la variable como convención escogida y solución al problema de outliers. Este paso lo llevaremos a cabo en caso de disponer valores extremos que no nos interesen en nuestra distribución para las siguientes variables de análisis.

Procedemos con el tratamiento de outliers:

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_fixed <- base.datos$fixed.acidity >= 11.9
```

```r
# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$fixed.acidity[condicion_outliers_fixed] <- NA
# Renombro NA's por valor medio:
base.datos$fixed.acidity[is.na(base.datos$fixed.acidity)] <-
mean(base.datos$fixed.acidity, na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
outliers_fixed <- boxplot.stats(base.datos$fixed.acidity)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$fixed.acidity, main = "Box Plot Fixed Acidity",
        ylab = "Valor de acidez",
        col = "grey")
```

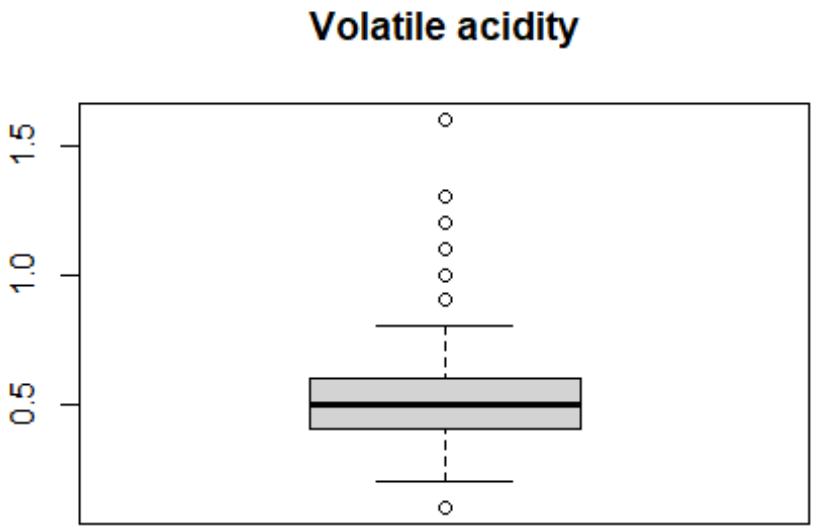


Observando el boxplot final tras la reestructuración vemos como el tratamiento de outliers ha sido correcto.

VARIABLE VOLATILE ACIDITY

Representación del boxplot y de los valores outliers:

```r
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$volatile.acidity,main="Volatile acidity")
```



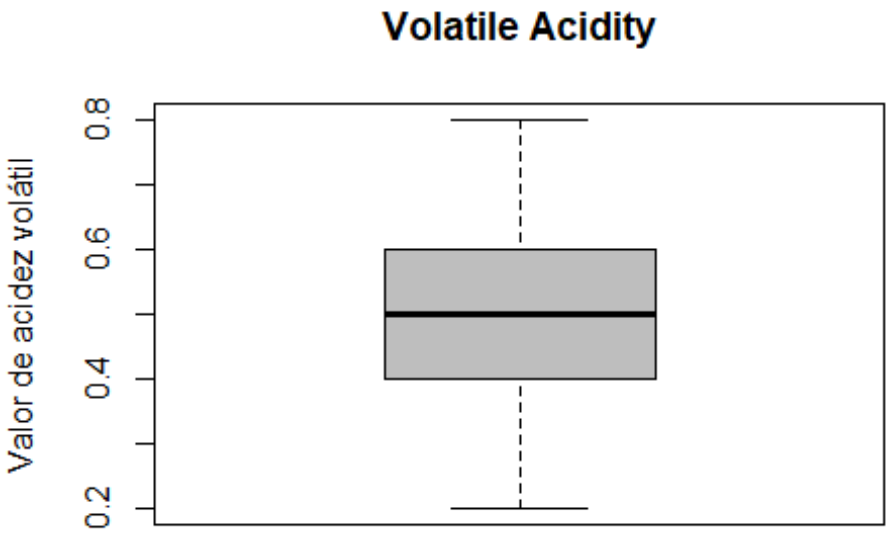

```
# Valor de los Outliers:
boxplot.stats(base.datos$volatile.acidity)$out
```

```
##  [1] 0.9 1.1 0.9 1.0 0.9 1.1 1.3 1.3 1.0 0.9 0.9 1.1 1.0 1.0 1.0 0.9 1.0
1.0 1.0
## [20] 0.9 0.9 0.9 0.9 0.9 0.9 1.2 1.0 1.2 0.9 1.0 1.0 1.0 1.1 0.9 0.9 1.0
1.0 0.9
## [39] 1.0 0.9 0.9 1.0 0.9 1.0 0.9 1.0 0.1 0.1 0.1 0.9 1.0 0.9 1.0 0.9 0.9
1.0 0.9
## [58] 0.9 1.0 1.6 0.9 1.2 0.9 0.9 1.0 0.9 0.9 1.0 1.0 0.9 0.9 0.9 0.9 0.9
0.9 0.9
```

Para Volatile acidity, de nuevo tratamos los outliers del mismo modo que anteriormente:

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_vol<- (base.datos$volatile.acidity > 0.1) &
(base.datos$volatile.acidity < 0.9)
# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$volatile.acidity[!condicion_outliers_vol] <- NA
# Renombro NA's por valor medio:
base.datos$volatile.acidity[is.na(base.datos$volatile.acidity)] <-
mean(base.datos$volatile.acidity, na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
outliers_vol <- boxplot.stats(base.datos$volatile.acidity)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$volatile.acidity, main = "Volatile Acidity",
```

```
        ylab = "Valor de acidez volátil",
        col = "grey")
```

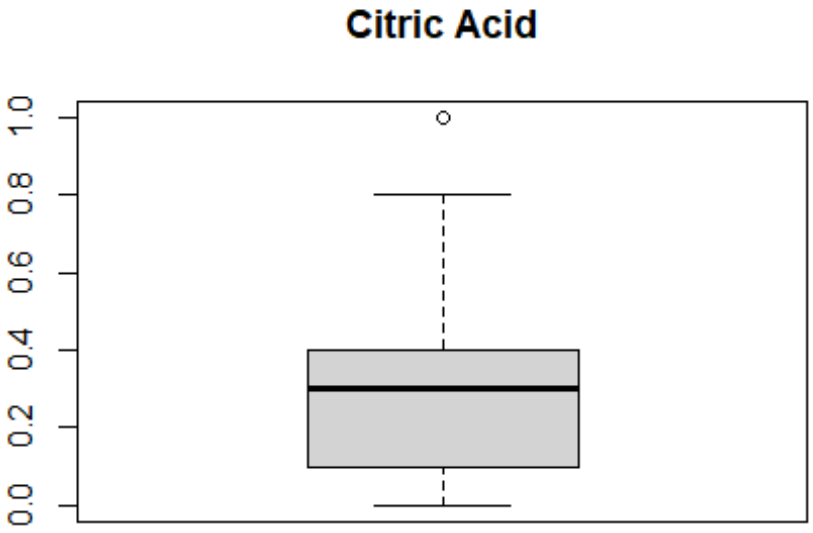


De nuevo, para la acidez volátil, observando el boxplot final tras la reestructuración vemos como el tratamiento de outliers ha sido correcto.

VARIABLE CITRIC ACID

Representación del boxplot y de los valores outliers:

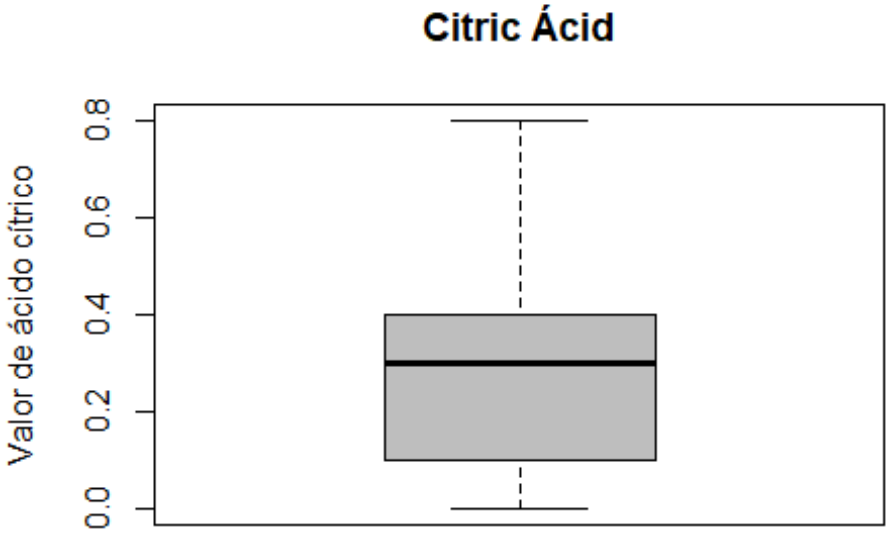
```
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$citric.acid,main="Citric Acid")
```




```
# Valor de los Outliers:
boxplot.stats(base.datos$citric.acid)$out

## [1] 1
```

Para citric acid, de nuevo tratamos los outliers del mismo modo que anteriormente:

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_cit<- base.datos$citric.acid >= 1
# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$citric.acid[condicion_outliers_cit] <- NA
# Renombro NA's por valor medio:
base.datos$citric.acid[is.na(base.datos$citric.acid)] <-
mean(base.datos$citric.acid, na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
outliers_cit <- boxplot.stats(base.datos$citric.acid)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$citric.acid, main = "Citric Ácid",
        ylab = "Valor de ácido cítrico",
        col = "grey")
```

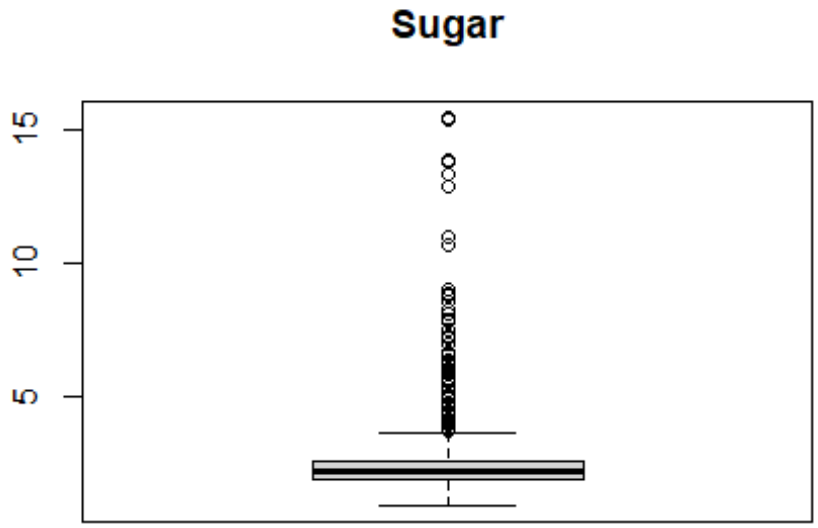


De nuevo, para la presencia de ácido cítrico, observando el boxplot, vemos como el tratamiento de outliers ha sido correcto.

VARIABLE AZÚCAR

Representación del boxplot y de los valores outliers:

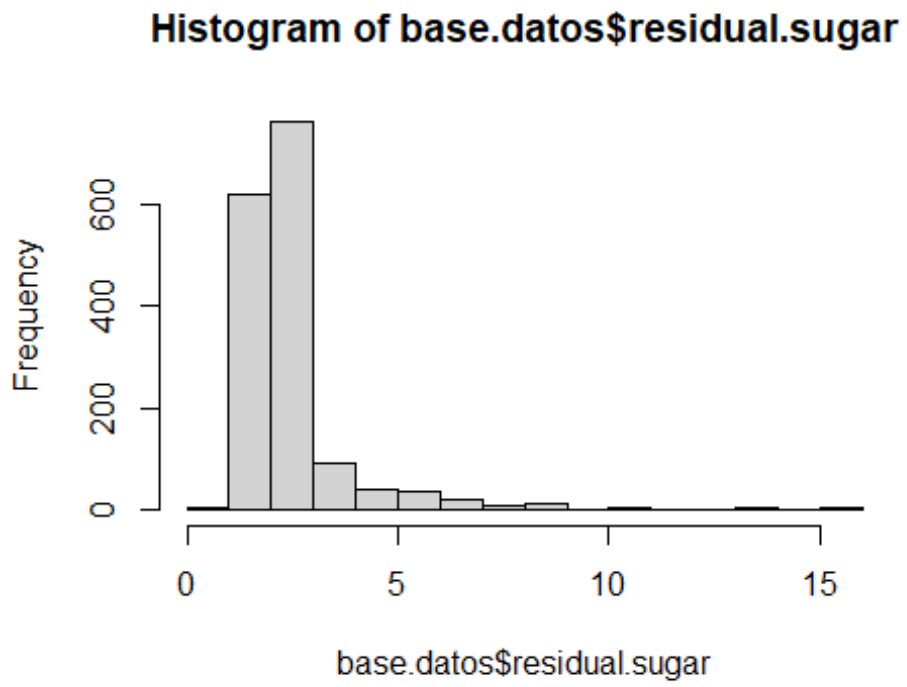
```
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$residual.sugar,main="Sugar")
```




```
# Valor de los Outliers:
boxplot.stats(base.datos$residual.sugar)$out
```

```
##   [1]  6.1  6.1  3.8  3.9  4.4 10.7  5.5  5.9  5.9  3.8  5.1  4.7  4.7
5.5  5.5
##  [16]  5.5  5.5  7.3  7.2  3.8  5.6  4.0  4.0  4.0  4.0  7.0  4.0  4.0
6.4  5.6
##  [31]  5.6 11.0 11.0  4.5  4.8  5.8  5.8  3.8  4.4  6.2  4.2  7.9  7.9
3.7  4.5
##  [46]  6.7  6.6  3.7  5.2 15.5  4.1  8.3  6.6  6.6  4.6  6.1  4.3  5.8
5.2  6.3
##  [61]  4.2  4.2  4.6  4.2  4.6  4.3  4.3  7.9  4.6  5.1  5.6  5.6  6.0
8.6  7.5
##  [76]  4.4  4.2  6.0  3.9  4.2  4.0  4.0  4.0  6.6  6.0  6.0  3.8  9.0
4.6  8.8
##  [91]  8.8  5.0  3.8  4.1  5.9  4.1  6.2  8.9  4.0  3.9  4.0  8.1  8.1
6.4  6.4
## [106]  8.3  8.3  4.7  5.5  5.5  4.3  5.5  3.7  6.2  5.6  7.8  4.6  5.8
4.1 12.9
## [121]  4.3 13.4  4.8  6.3  4.5  4.5  4.3  4.3  3.9  3.8  5.4  3.8  6.1
3.9  5.1
## [136]  5.1  3.9 15.4 15.4  4.8  5.2  5.2  3.8 13.8 13.8  5.7  4.3  4.1
4.1  4.4
## [151]  3.7  6.7 13.9  5.1  7.8
```

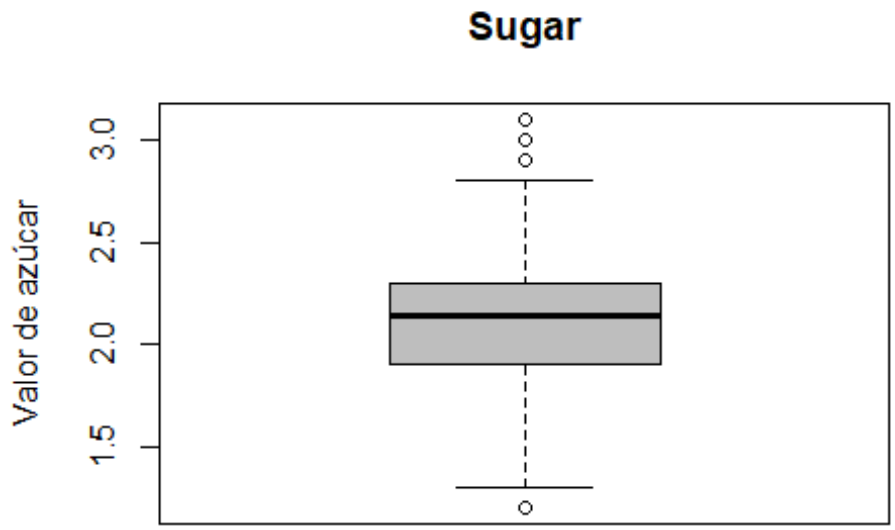
```
hist(base.datos$residual.sugar)
```



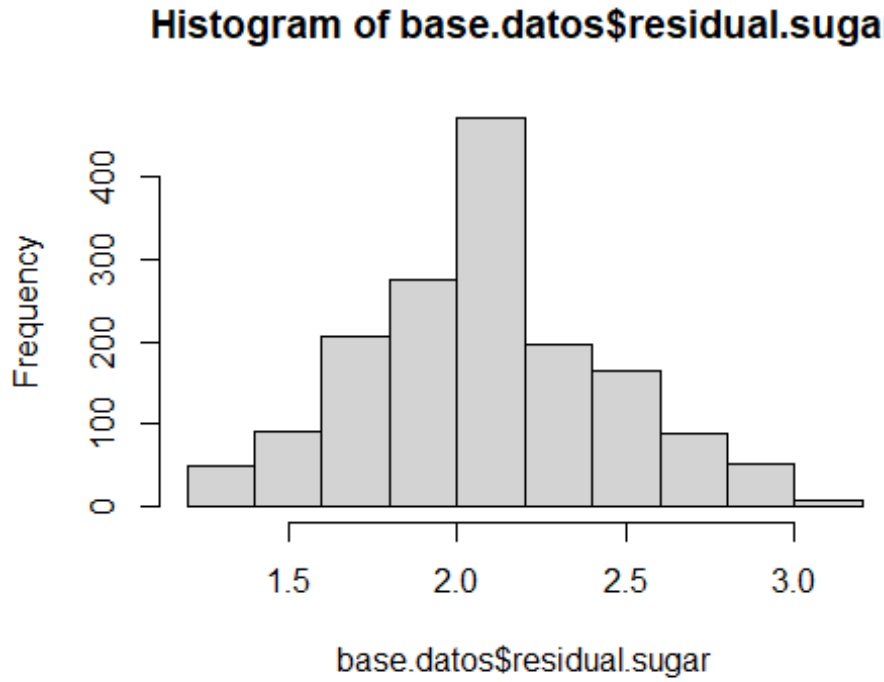
En este caso particular de azúcar, vemos como disponemos de muchos valores de outliers, por lo que hemos decidido representar el histograma de frecuencias para ver si los outliers se corresponden con una gran cantidad de valores o estos se encuentran distribuidos…

Tras observar el histograma, vemos que hay muy poca frecuencia de valores outliers por lo que vamos a proceder a tratarlos finalmente en la línea de los casos anteriores.

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_su<- (base.datos$residual.sugar> 0.9) &
(base.datos$residual.sugar < 3.2)
# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$residual.sugar[!condicion_outliers_su] <- NA
# Renombro NA's por valor medio:
base.datos$residual.sugar[is.na(base.datos$residual.sugar)] <-
mean(base.datos$residual.sugar, na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
outliers_su <- boxplot.stats(base.datos$residual.sugar)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$residual.sugar, main = "Sugar",
        ylab = "Valor de azúcar",
        col = "grey")
```




```
# Por ultimo, represento el histograma para ver la repartición de
frecuencias:
hist(base.datos$residual.sugar)
```

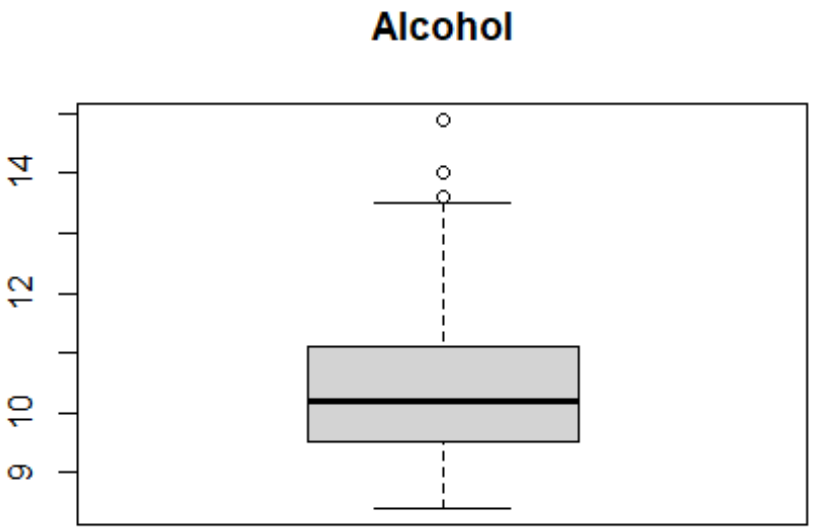


Para el azúcar, observando tanto el boxplot final tras la reestructuración como el histograma, vemos como el tratamiento de outliers ha sido correcto.

VARIABLE ALCOHOL

Representación del boxplot y de los valores outliers:

```
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$alcohol,main="Alcohol")
```




```
# Valor de los Outliers:
boxplot.stats(base.datos$alcohol)$out
```

```
##  [1] 14.0 14.0 14.0 14.0 14.9 14.0 13.6 13.6 13.6 14.0 14.0 13.6 13.6
```

Para el alcohol, tratamos los outliers del mismo modo que anteriormente, eliminando aquellos que se denotan exteriormente al boxplot:

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_al<- base.datos$alcohol >= 13.2

# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$alcohol[condicion_outliers_al] <- NA
# Renombro NA's por valor medio:
base.datos$alcohol[is.na(base.datos$alcohol)] <- mean(base.datos$alcohol,
na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
```

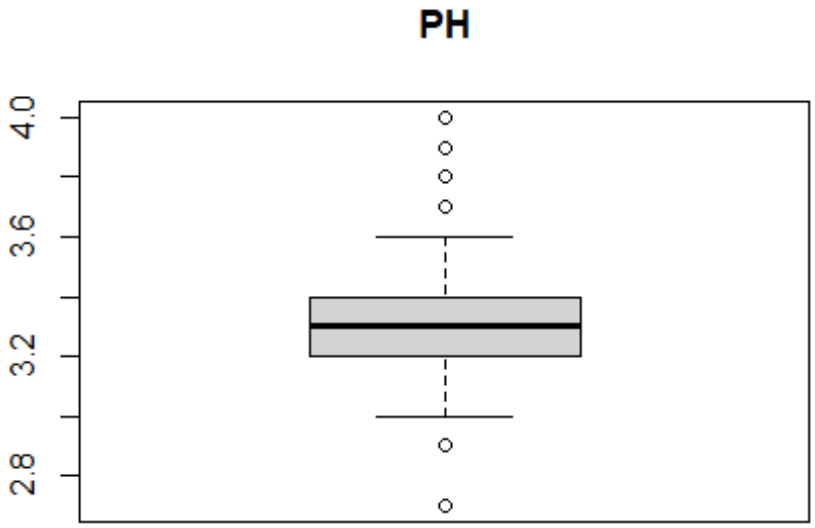
```
outliers_al <- boxplot.stats(base.datos$alcohol)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$alcohol, main = "Alcohol",
        ylab = "Alcohol",
        col = "grey")
```

Para el alcohol, observando el boxplot, vemos como el tratamiento de outliers ha sido correcto.

VARIABLE PH

Representación del boxplot y de los valores outliers:

```
# Definición de boxplot de muestra y obtención de valores atípicos:
boxplot(base.datos$pH,main="PH")
```

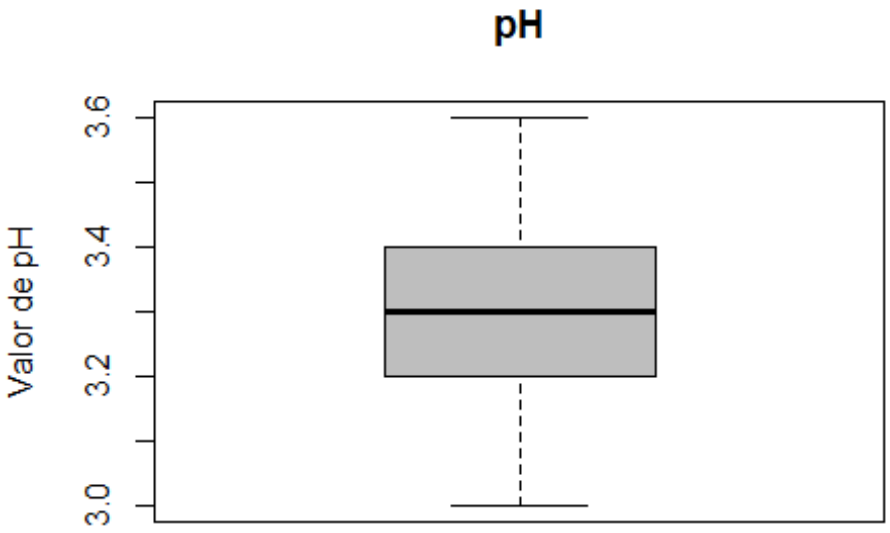



```
# Valor de los Outliers:
boxplot.stats(base.datos$pH)$out
```

```
##  [1] 3.9 2.9 2.9 2.9 3.8 3.9 3.7 3.7 2.7 3.7 3.7 3.7 3.7 3.7 2.9 2.9 3.7
2.9 2.9
## [20] 2.9 3.7 2.9 2.9 2.9 2.9 2.9 2.9 3.9 2.9 3.7 3.7 3.7 3.7 3.7 3.7 3.7
3.7 2.9
## [39] 2.9 3.8 2.9 3.7 3.7 3.7 3.8 4.0 2.9 4.0 3.7 2.9 3.7 3.7 3.7 3.7
```

Para pH, tratamos los outliers del mismo modo que anteriormente, eliminando aquellos que se denotan exteriormente al boxplot tanto los mínimos como los máximos:

```
# Establezco un límite el mínimo valor de outlier:
condicion_outliers_p<- (base.datos$pH> 2.9) & (base.datos$pH < 3.7)
# Asigno a aquellos valores que no cumplen la condición un valor NA:
base.datos$pH[!condicion_outliers_p] <- NA
# Renombro NA's por valor medio:
base.datos$pH[is.na(base.datos$pH)] <- mean(base.datos$pH, na.rm = TRUE)
# Muestro los outlietrs actuales una vez eliminados los casos anormales:
outliers_p <- boxplot.stats(base.datos$pH)$out
# Muestro el nuevo boxplot sin outliers anormales:
boxplot(base.datos$pH, main = "pH",
        ylab = "Valor de pH",
        col = "grey")
```



Para esta última variable, de nuevo, observando el boxplot, vemos como el tratamiento de outliers ha sido correcto.

Por último y antes de proceder con el análisis de datos, verificamos que después de la manipulación de outliers, no disponemos de ningún valor NA en nuestras distribuciones, ya que afectaría a la continuidad del proceso de análisis.

```
# Compruevo de nuevo que no dispongo de valores NA:
library(dplyr)
base.datos %>%
select(everything()) %>%
summarise_all(funs(sum(is.na(.))))

##   quality fixed.acidity volatile.acidity citric.acid residual.sugar
alcohol pH
## 1       0             0                0           0              0
0  0
```

El siguiente paso, tras haber tratado los datos en base a múltiples consideraciones (missings, outliers, homogeneidad en los datos, etc…) nos dispondremos a evaluar, de nuevo campo por campo, la distribución de estos y características relativas a su normalidad, varianza y relación entre variables. Debemos recordar, que la base del trabajo, tal y como presentamos en una primera instancia al inicio de este, se fundamenta en la composición de la calidad de un vino en base a la acidez de este y el contraste con otros aspectos como el dulzor de la uva (azúcar residual) y el alcohol del propio vino.

4. ANÁLISIS DE LOS DATOS

A lo largo de este apartado, vamos a tratar cada una de las variables de nuestro dataset en relación a su Normalidad y su Homocedasticidad para la varianza para posteriormente (apartado 4.3) poder llevar a cabo una serie de pruebas estadísticas en base a las variables que disponemos (regresiones, correlación, etc…). Para ello aplicaremos en una primera instancia diferentes técnicas tanto gráficas como estadísticas para el correcto tratamiento en base a estas dos consideraciones. Para esta serie de pruebas, sobre todo en el análisis de la varianza, asumimos que nuestro principal objetivo, tal y como ya se comentó al inicio de la actividad, es la relación calidad-acidez del vino, por lo que compararemos cada variable con calidad. Los análisis, se van a componer del siguiente orden de evaluación:

1.Representación del Histograma de cada distribución y comparación en base a una curva normal.

2.Representación gráfica de la normalidad-QQPlot.

3.Test Shapiro-Wilks para evaluar la normalidad. Dicho test basa su contraste de hipótesis en: H0: La distribución es normal H1: La distribución No es normal Evaluaremos cada hipótesis en función del p-value para un nivel de significancia a priori del 0.05.

Para comprobar la normalidad, también podemos aplicar el Test de Kolmogorov-Smirnov (KS). Test que compara dos distribuciones en base a una hipótesis nula de igualdad entre éstas y una hipótesis alternativa de diferencia. Para ello compararemos nuestras distribuciones con distribuciones normales.

El hecho de que usemos KS para ratificar el valor del test de Shapiro en algunos casos, por un lado es debido a que este último generalmente se usa para contrastar la normalidad en muestras ligeramente pequeñas. En R podemos usarlo con muestras con hasta 5000 elementos, pero es aconsejable y presenta mayor robustez para muestras realmente bajas en número de elementos. Por otro lado, no usamos únicamente el test de KS en una primera instancia ya que a pesar de que continuamente se alude a dicho test como válido para contrastar la normalidad, esto no es del todo cierto. KS asume que se conoce la media y varianza poblacional, cosa que en la mayoría de los casos no es posible, dotando por tanto de un test muy conservador y poco potente, pero que nos sirve para ratificar Shapiro.

Por ende, en caso de querer ir más allá en el análisis de normalidad, tras Shapiro y KS que se representan como tests clásicos en los análisis de normalidad, podemos hacer uso del test de Lilliefors. Este último, aunque no tan común, asume que la media y varianza son desconocidas, estando especialmente desarrollado para contrastar la normalidad.

4. Test de Levene para evaluar la Homocedasticidad. Dicho test basa su contraste de hipótesis en: H0: Homogeneidad en la varianza (Homocedasticidad) H1: NO Homogeneidad en la varianza (Heterocedasticidad) Evaluaremos de nuevo dicho test en base al p-value en un nivel de significancia del 0.05.

El test de Levene es especialmente útil para llevar a cabo el análisis de la varianza en torno a dos variables, en nuestro caso comparando siempre con el caso de estudio de la variable calidad, ya que no es sensible a que las distribuciones deban presentar normalidad.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Este apartado ya se ha llevado a cabo anteriormente debido a una prematura subselección de nuestro conjunto de datos por lo que en este punto ya disponemos de las variables que vamos a integrar a lo largo de nuestro análisis de datos y posteriormente en las diferentes pruebas estadísticas que vamos a llevar a cabo.

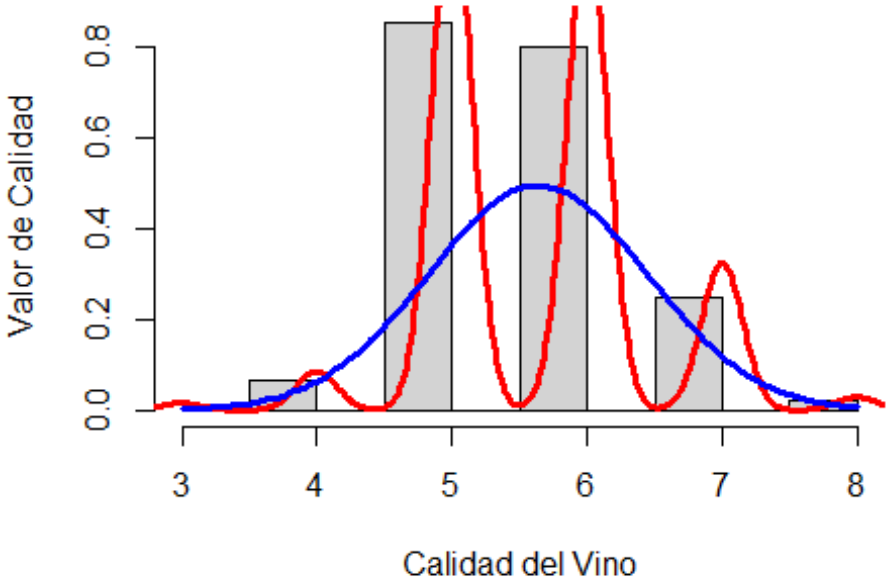
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

En este punto, procedemos con la evaluación de los tests y gráficos comentados para cada una de nuestras variables.

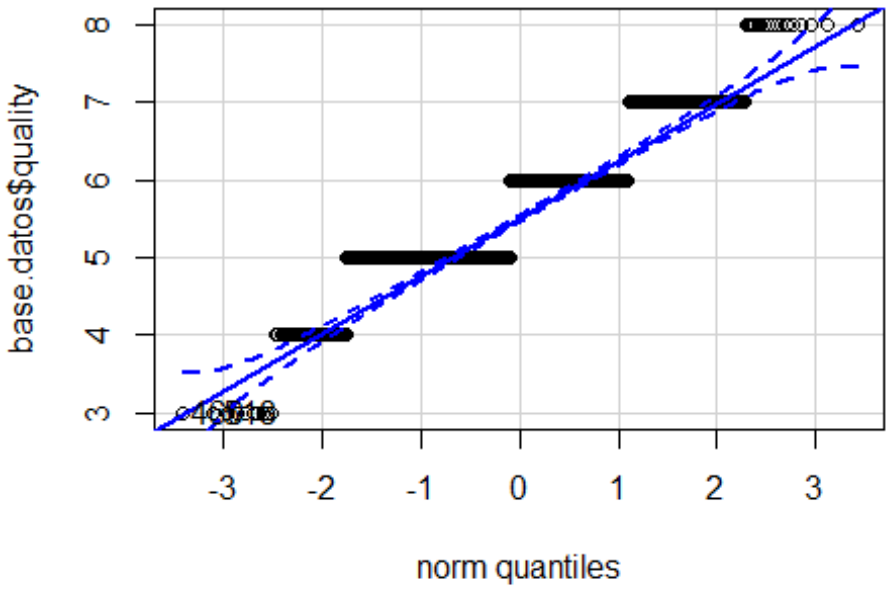
VARIABLE CALIDAD

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

## Warning: package 'car' was built under R version 4.0.5

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

# Representación de su distribución en base a una normal:
hist(base.datos$quality, freq = F,
     ylab = "Valor de Calidad",
     xlab = "Calidad del Vino", main = "")

dz1 <- density(base.datos$quality)
lines(dz1, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$quality), sd(base.datos$quality)),
col = "blue", lwd = 3, add = TRUE)
```




```r
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$quality)
```



```
## [1] 460 518

# Test de Shapiro. Normalidad:
shapiro.test(base.datos$quality)

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

# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$quality, base.datos$quality)

## Warning in leveneTest.default(base.datos$quality, base.datos$quality):
## base.datos$quality coerced to factor.

## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5
##       1593
```

Resultados variable calidad:

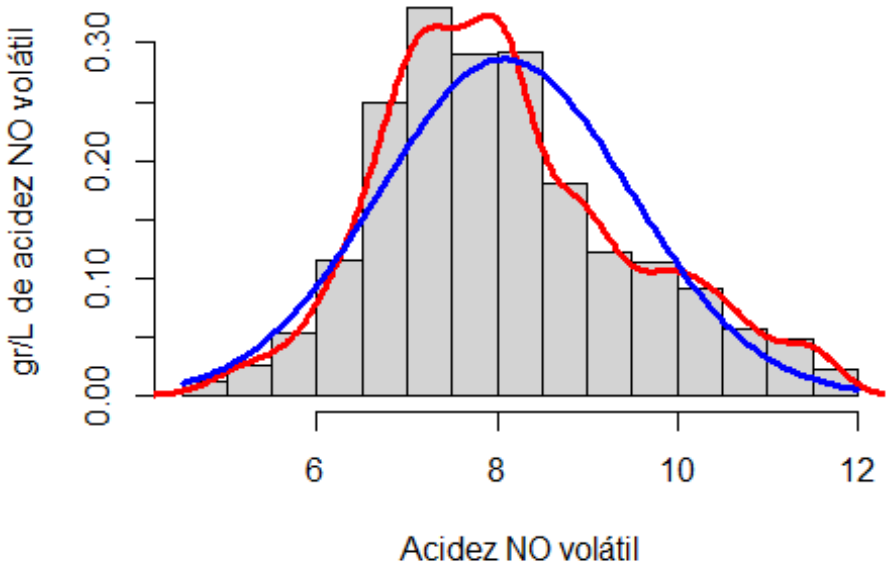
En base al análisis gráfico histograma vs normal, qqplot y al análisis mediante el Test de Shapiro, podemos concluir que la variable calidad no sigue una distribución Normal (p-value <<< 0 por lo que la hipótesis nula de Normalidad es falsa). En este caso no es necesario el uso del test KS para la ratificación de la No normalidad.

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso no aplica ya que estamos evaluando la propia variable Calidad.

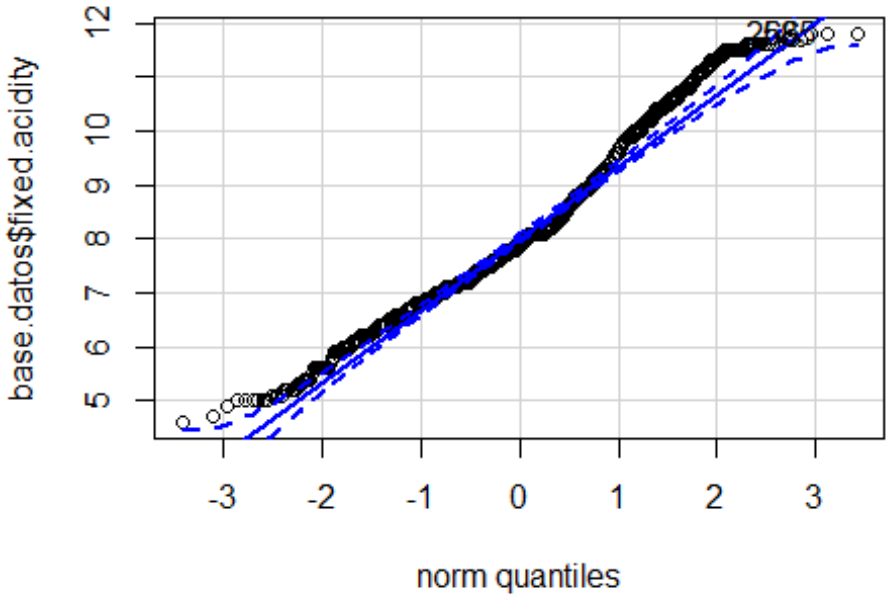
VARIABLE FIXED ACIDITY (acidez NO volátil)

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$fixed.acidity, freq = F,
     ylab = "gr/L de acidez NO volátil",
     xlab = "Acidez NO volátil", main = "")

dz2 <- density(base.datos$fixed.acidity)
lines(dz2, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$fixed.acidity), sd(base.datos$fixed.acidity)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$fixed.acidity)
```



```
## [1] 266 585
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$fixed.acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$fixed.acidity
## W = 0.97372, p-value < 2.2e-16
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$fixed.acidity,"pnorm", mean(base.datos$fixed.acidity),
sd(base.datos$fixed.acidity))
```

```
## Warning in ks.test(x = base.datos$fixed.acidity, "pnorm",
## mean(base.datos$fixed.acidity), : ties should not be present for the
Kolmogorov-
## Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$fixed.acidity
## D = 0.09837, p-value = 7.272e-14
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$fixed.acidity)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$fixed.acidity
## D = 0.09837, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$fixed.acidity, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$fixed.acidity,
base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  5.7282 3.022e-05 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Resultados variable fixed acidity:

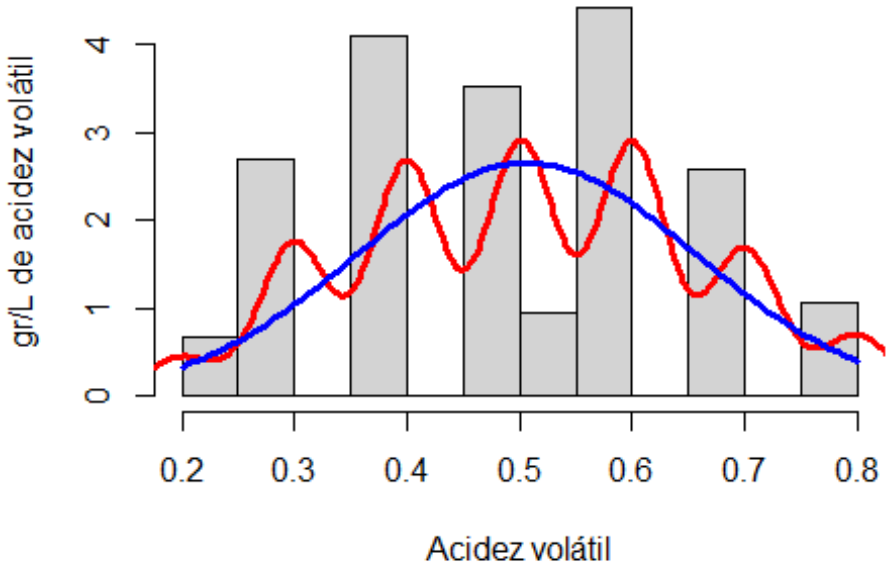
En base al análisis gráfico histograma vs normal y qqplot a priori podríamos decir que la distribución presenta cierto grado de normalidad, aunque no suficiente para ser significativo ni en el Test de Shapiro, ni en KS, ni el Lilliefors. Por tanto, no podemos concluir Normalidad en fixed acidity. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso tampoco nos presenta significancia y por ende Heterocedasticidad (p-value <<<<0).

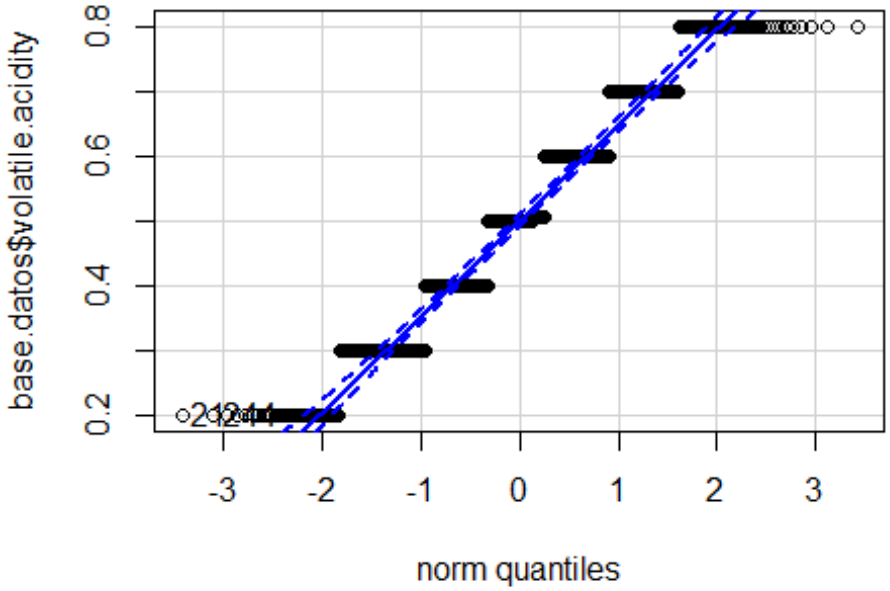
VARIABLE VOLATILE ACIDITY (Acidez volátil)

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$volatile.acidity, freq = F,
     ylab = "gr/L de acidez volátil",
     xlab = "Acidez volátil", main = "")

dz3 <- density(base.datos$volatile.acidity)
lines(dz3, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$volatile.acidity),
sd(base.datos$volatile.acidity)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$volatile.acidity)
```



```
## [1]  21 244
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$volatile.acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$volatile.acidity
## W = 0.95087, p-value < 2.2e-16
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$volatile.acidity,"pnorm",
mean(base.datos$volatile.acidity), sd(base.datos$volatile.acidity))
```

```
## Warning in ks.test(x = base.datos$volatile.acidity, "pnorm",
## mean(base.datos$volatile.acidity), : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$volatile.acidity
## D = 0.13614, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$volatile.acidity)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$volatile.acidity
## D = 0.13614, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$volatile.acidity, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$volatile.acidity,
base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  0.8927 0.4851
##       1593
```

Resultados variable Volatile acidity:

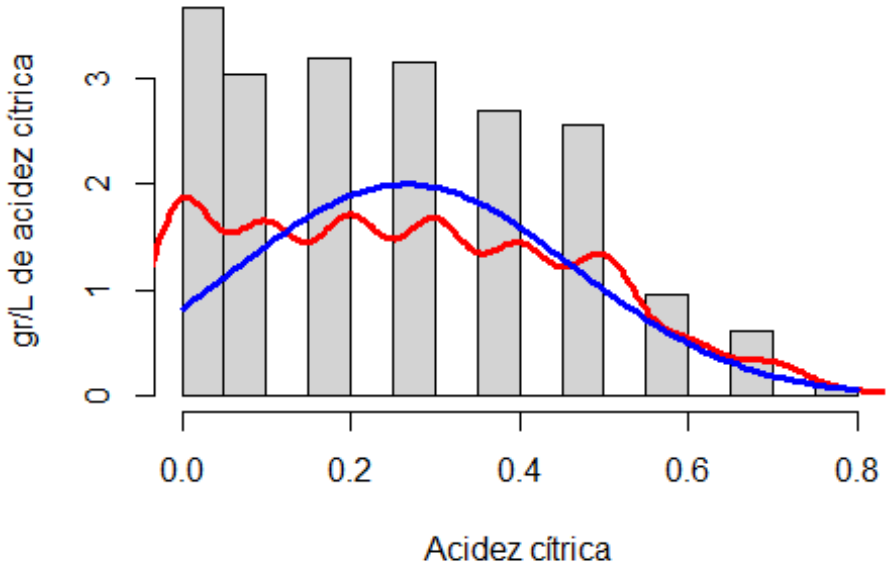
En base al análisis gráfico histograma vs normal y qqplot podemos observar como dicha variable no presenta signos de normalidad. Este hecho lo confirman los tests de Normalidad; Test de Shapiro, KS y Lilliefors. Por tanto no podemos concluir Normalidad en volatile acidity. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso sí que nos indica significancia y por ende Homocedasticidad en la varianza en la comparación de la distribución con la variable quality. (p-value = 0.48 por lo que no podemos rechazar la hipótesis nula de homogeneidad en la varianza).

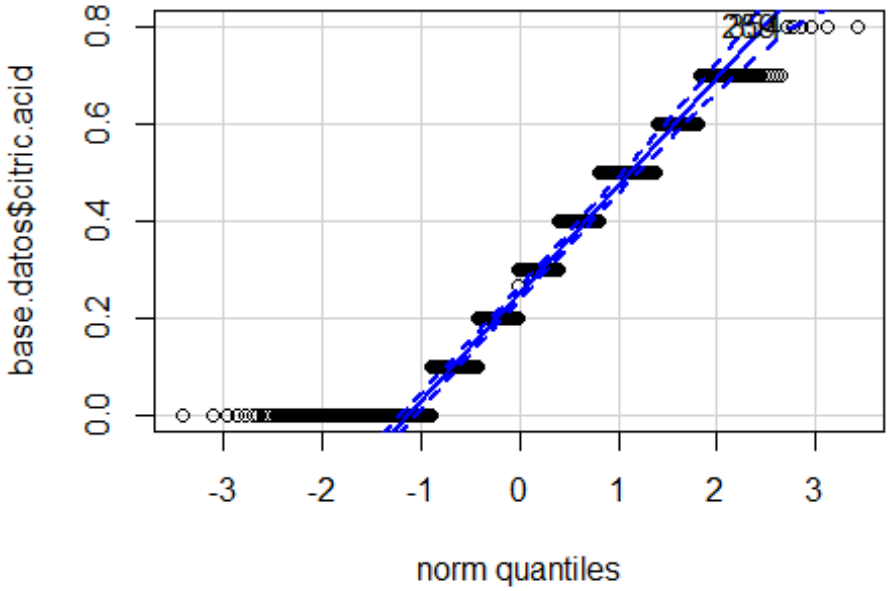
VARIABLE CITRIC ACID

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$citric.acid, freq = F,
     ylab = "gr/L de acidez cítrica",
     xlab = "Acidez cítrica", main = "")

dz4 <- density(base.datos$citric.acid)
lines(dz4, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$citric.acid), sd(base.datos$citric.acid)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$citric.acid)
```



```
## [1] 259 354
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$citric.acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$citric.acid
## W = 0.934, p-value < 2.2e-16
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$citric.acid,"pnorm", mean(base.datos$citric.acid),
sd(base.datos$citric.acid))
```

```
## Warning in ks.test(x = base.datos$citric.acid, "pnorm",
## mean(base.datos$citric.acid), : ties should not be present for the
Kolmogorov-
## Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$citric.acid
## D = 0.13382, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$citric.acid)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$citric.acid
## D = 0.13382, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$citric.acid, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$citric.acid, base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value  Pr(>F)
## group    5  3.0678 0.00923 **
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Resultados variable citric acid:

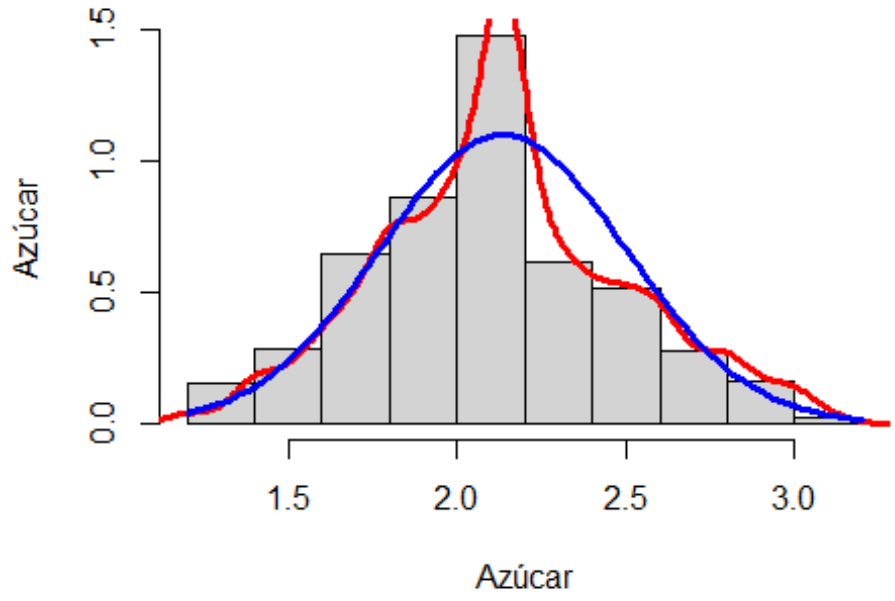
En base al análisis gráfico histograma vs normal y qqplot podemos observar como dicha variable no presenta signos de normalidad. Este hecho lo confirman los tests de Normalidad; Test de Shapiro, KS y Lilliefors. Por tanto no podemos concluir Normalidad en citric acid. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso tampoco nos presenta suficiente significancia y por ende Heterocedasticidad (p-value = 0.00923), aunque en este caso, el p-value está cerca del valor 0.05 y por ende de haber podido asumir homocedasticidad.

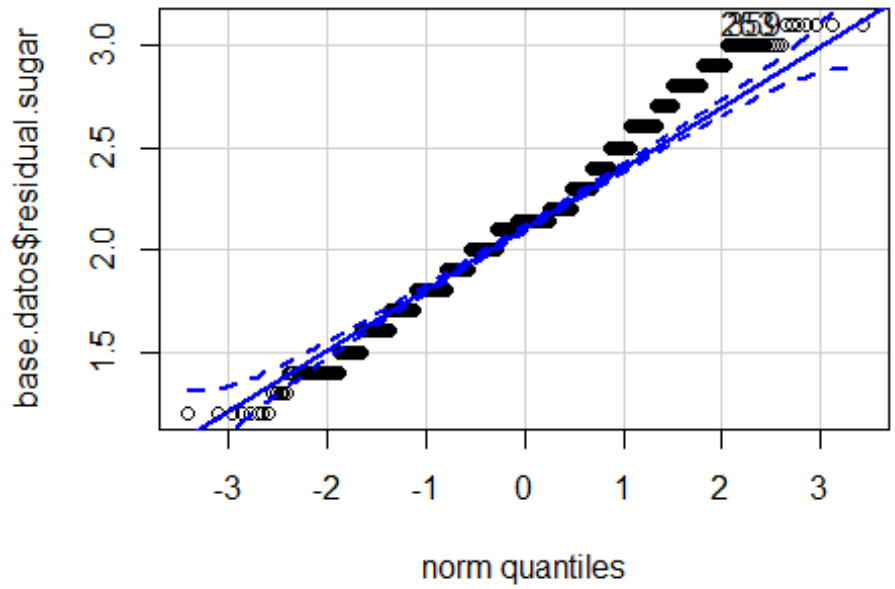
VARIABLE RESIDUAL SUGAR

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$residual.sugar, freq = F,
     ylab = "Azúcar",
     xlab = "Azúcar", main = "")

dz5 <- density(base.datos$residual.sugar)
lines(dz5, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$residual.sugar),
sd(base.datos$residual.sugar)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$residual.sugar)
```



```
## [1] 253 359
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$residual.sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$residual.sugar
## W = 0.9833, p-value = 1.119e-12
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$residual.sugar,"pnorm",
mean(base.datos$residual.sugar), sd(base.datos$residual.sugar))
```

```
## Warning in ks.test(x = base.datos$residual.sugar, "pnorm",
## mean(base.datos$residual.sugar), : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$residual.sugar
## D = 0.11336, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$residual.sugar)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$residual.sugar
## D = 0.11336, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$residual.sugar, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$residual.sugar,
base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5  1.2215 0.2965
##       1593
```

Resultados variable residual sugar:

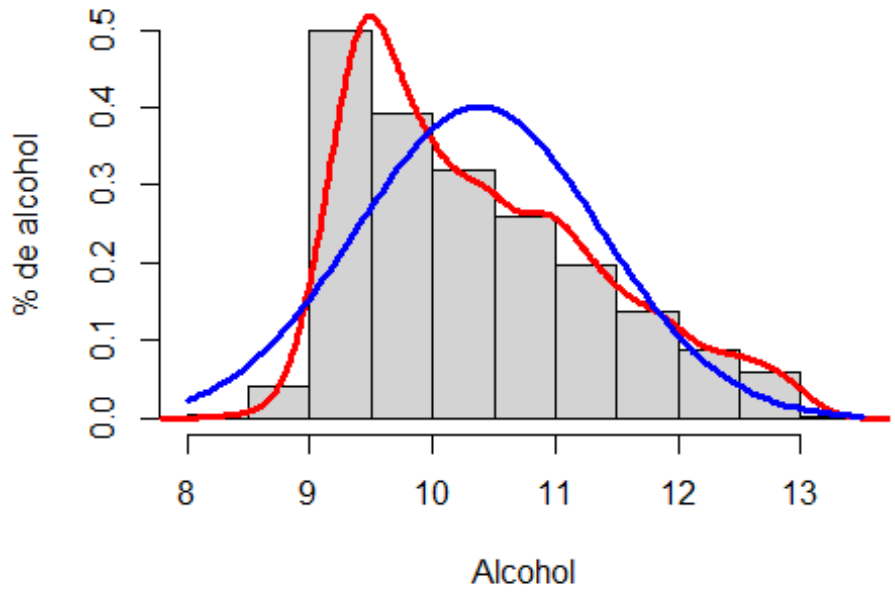
En base al análisis gráfico histograma vs normal y qqplot a priori podríamos decir que la distribución presenta cierto grado de normalidad, aunque no suficiente para ser significativo ni en el Test de Shapiro, ni en KS, ni el Lilliefors. Por tanto no podemos concluir Normalidad en residual sugar. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso sí que nos indica significancia y por ende Homocedasticidad en la varianza en la comparación de la distribución con la variable quality. (p-value = 0.29 por lo que no podemos rechazar la hipótesis nula de homogeneidad en la varianza).

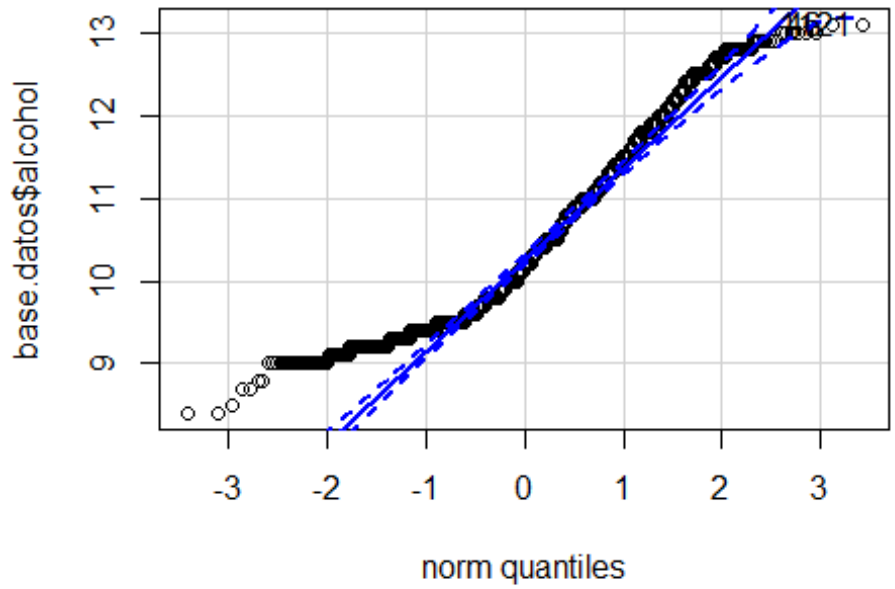
VARIABLE ALCOHOL

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$alcohol, freq = F,
     ylab = "% de alcohol",
     xlab = "Alcohol", main = "")

dz5 <- density(base.datos$alcohol)
lines(dz5, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$alcohol), sd(base.datos$alcohol)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$alcohol)
```



```
## [1]   46 1121
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$alcohol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$alcohol
## W = 0.93588, p-value < 2.2e-16
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$alcohol,"pnorm", mean(base.datos$alcohol),
sd(base.datos$alcohol))
```

```
## Warning in ks.test(x = base.datos$alcohol, "pnorm",
mean(base.datos$alcohol), :
## ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$alcohol
## D = 0.1169, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$alcohol)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$alcohol
## D = 0.1169, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$alcohol, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$alcohol, base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    5  22.281 < 2.2e-16 ***
##       1593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Resultados variable alcohol:

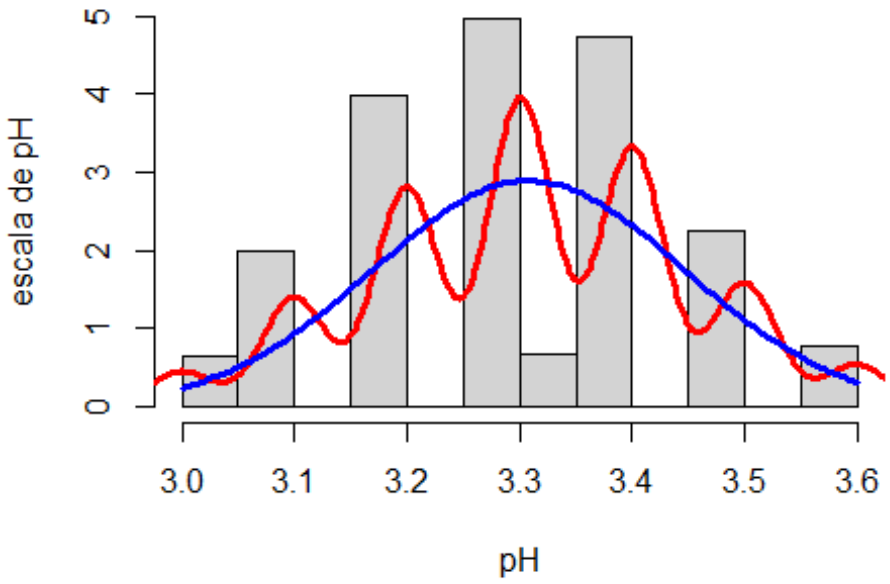
En base al análisis gráfico histograma vs normal y qqplot podemos observar como dicha variable no presenta signos de normalidad. Este hecho lo confirman los tests de Normalidad; Test de Shapiro, KS y Lilliefors. Por tanto no podemos concluir Normalidad en alcohol. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas tampoco nos indica significancia y por ende Homocedasticidad en la varianza en la comparación de la distribución con la variable quality. (p-value <<<<0 por lo que no podemos asumir la hipótesis nula de homogeneidad en la varianza).

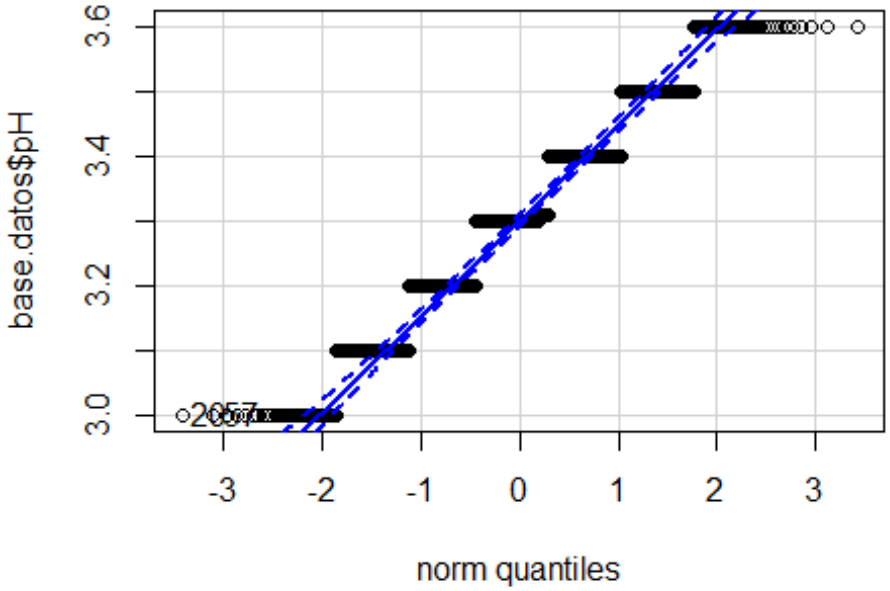
VARIABLE pH

Evaluación:

```
# Definición de la librería Car:

#install.packages("car")
library(car)

# Representación de su distribución en base a una normal:
hist(base.datos$pH, freq = F,
     ylab = "escala de pH",
     xlab = "pH", main = "")

dz5 <- density(base.datos$pH)
lines(dz5, col = "red", lwd = 3)

curve(dnorm(x, mean(base.datos$pH), sd(base.datos$pH)),
col = "blue", lwd = 3, add = TRUE)
```




```
# Representación de su QQ-PLOT:
library("car")
qqPlot(base.datos$pH)
```



```
## [1] 20 57
```

```
# Test de Shapiro. Normalidad:
shapiro.test(base.datos$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  base.datos$pH
## W = 0.95186, p-value < 2.2e-16
```

```
# Test Kolmogorov-Smirnov para la evaluación de normalidad:
ks.test(x = base.datos$pH,"pnorm", mean(base.datos$pH), sd(base.datos$pH))
```

```
## Warning in ks.test(x = base.datos$pH, "pnorm", mean(base.datos$pH),
## sd(base.datos$pH)): ties should not be present for the Kolmogorov-Smirnov
test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  base.datos$pH
## D = 0.1447, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Test de Lilliefors Normalidad:
# install.packages("nortest") # lo instalo si no dispongo del paquete
library("nortest")
lillie.test(base.datos$pH)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  base.datos$pH
## D = 0.1447, p-value < 2.2e-16
```

```
# Test de Levene. Varianza con respecto a calidad:
leveneTest(base.datos$pH, base.datos$quality)
```

```
## Warning in leveneTest.default(base.datos$pH, base.datos$quality):
## base.datos$quality coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    5   0.736 0.5964
##       1593
```

Resultados variable residual sugar:

En base al análisis gráfico histograma vs normal y qqplot a priori podríamos decir que la distribución presenta cierto grado de normalidad, aunque no suficiente para ser significativo ni en el Test de Shapiro, ni en KS, ni el Lilliefors. Por tanto no podemos concluir

Normalidad en la variable pH. (p-value <<< 0 en todos los casos por lo que la hipótesis nula de Normalidad es falsa).

Por otro lado, el test de Levene para la evaluación de la igualdad de Varianzas en este caso sí que nos indica significancia y por ende Homocedasticidad en la varianza en la comparación de la distribución con la variable quality. (p-value = 0.59 por lo que no podemos rechazar la hipótesis nula de homogeneidad en la varianza).

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

En este apartado, en función de los datos y el objetivo del estudio se van a aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. en el desarrollo del análisis de la calidad del vino en torno a múltiples variables de acidez, azúcar y alcohol.

Análisis de Correlación multivariable

En términos genéricos, la correlación cuantifica el grado de relación entre dos variables. El cálculo de la correlación entre dos variables es independiente del orden o asignación de cada variable a X e Y, midiendo únicamente la relación entre ambas sin considerar dependencias. Generalmente y siguiendo en la línea en la que vamos a proceder, el análisis de correlación lineal precede a la generación del modelo de regresión lineal. Por tanto, en este punto vamos a estudiar/observar si las variables de nuestro dataset se encuentran correlacionadas en un análisis multivariante.

El análisis de la correlación entre dos variables se basa en el cómputo de la covarianza estandarizada generando diferentes coeficientes de correlación. Existen diferentes coeficientes en función del tipo de variables de las que dispongamos (Pearson, Spearman, Kendal, etc…).

Para poder elegir el coeficiente de correlación adecuado, se tiene que analizar el tipo de variables y la distribución que presentan. Este paso ya lo hemos llevado a cabo anteriormente, obteniendo como resultado que ninguna distribución presenta un comportamiento normal. Este hecho excluye la posibilidad de utilizar el coeficiente de Pearson, uno de los más típicos, dejando como alternativas el de Spearman o Kendall. Sin embargo, como el coeficiente de Pearson tiene cierta robustez, a fines prácticos podemos usarlo siempre y cuando se tenga en cuenta este hecho en los resultados.

En este punto, procedemos con el cómputo de la correlación mediante la obtención del coeficiente de correlación de Pearson para la totalidad de las variables. Asumiendo que este puede presentar cierta variabilidad al no disponer de normalidad en los datos, para aquellos valores de correlación más significantes, se llevará a cabo una ratificación con Speerman. Spearman es adecuado cuando la distribución de los datos no sigue una ley normal y son de naturaleza ordinal.

Procedemos en una primera instancia:

```
# Evaluación de la correlación entre las 3 variables:
# install.packages("GGally") # instalo la librería en caso de no tenerla
```

```
# install.packages("ggplot2")
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.5
```

```
## Loading required package: ggplot2
```

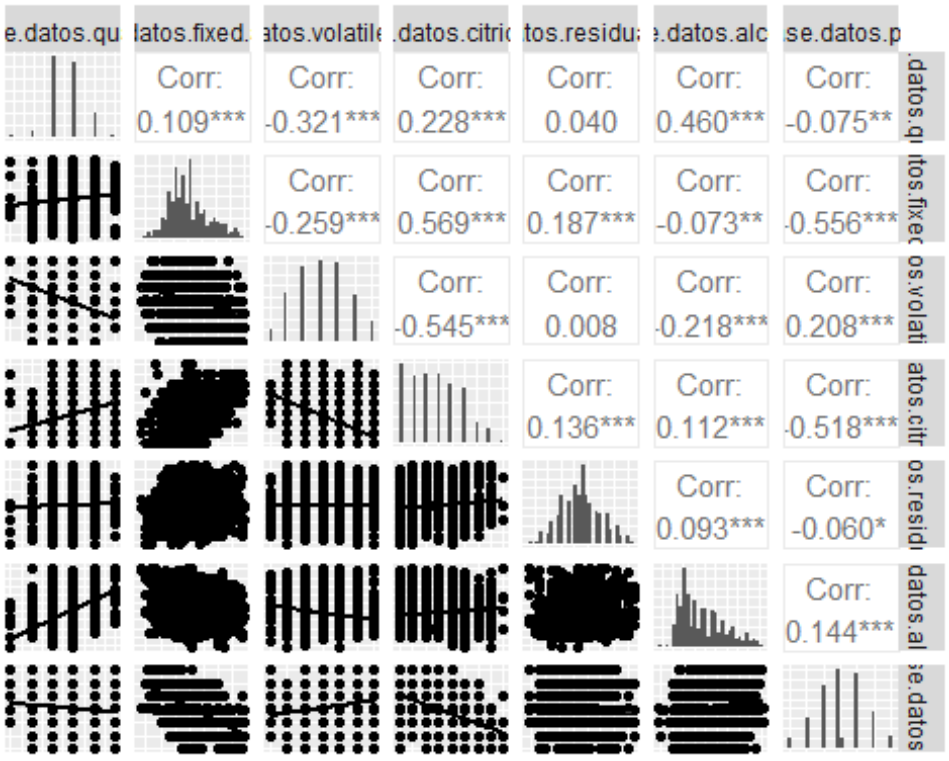
```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
new_data <- data.frame(base.datos$quality, base.datos$fixed.acidity,
base.datos$volatile.acidity, base.datos$citric.acid,
base.datos$residual.sugar, base.datos$alcohol, base.datos$pH)

ggpairs(new_data , lower = list(continuous = "smooth"),
        diag = list(continuous = "barDiag"), axisLabels = "none")
```

```
## `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`.
```



Valores a considerar de correlación:

alcohol y quality = 0.46 positiva fixed acidity y citric acid = 0.569 positiva fixed acidity y pH = -0.556 negativa citric acid y volatile acid = -0.545 negativa citric acid y pH = -0.518 negativa

Ratificación con Speerman:

```
# Alcohol vs quality:
cor.test(x = base.datos$alcohol, y = base.datos$quality, method = "spearman")

## Warning in cor.test.default(x = base.datos$alcohol, y =
base.datos$quality, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  base.datos$alcohol and base.datos$quality
## S = 363333975, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4667731

# Fixed acidity vs citric acid:
cor.test(x = base.datos$fixed.acidity, y = base.datos$citric.acid, method =
"spearman")

## Warning in cor.test.default(x = base.datos$fixed.acidity, y =
## base.datos$citric.acid, : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  base.datos$fixed.acidity and base.datos$citric.acid
## S = 279013831, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5905209

# Fixed acidity vs pH:
cor.test(x = base.datos$fixed.acidity, y = base.datos$pH, method =
"spearman")

## Warning in cor.test.default(x = base.datos$fixed.acidity, y =
base.datos$pH, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
```

```
##
## data:  base.datos$fixed.acidity and base.datos$pH
## S = 1093487074, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6047954
```

```
# Citric acid vs volatile acid:
cor.test(x = base.datos$citric.acid, y = base.datos$volatile.acidity, method
= "spearman")
```

```
## Warning in cor.test.default(x = base.datos$citric.acid, y =
## base.datos$volatile.acidity, : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  base.datos$citric.acid and base.datos$volatile.acidity
## S = 1081138856, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5866733
```

```
# Citric acid vs pH:
cor.test(x = base.datos$citric.acid, y = base.datos$pH, method = "spearman")
```

```
## Warning in cor.test.default(x = base.datos$citric.acid, y = base.datos$pH,
:
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  base.datos$citric.acid and base.datos$pH
## S = 1032179534, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5148208
```

En todos los casos, vemos como la rho del test de Speerman presenta un valor significativo de correlación de acuerdo a los valores obtenidos con Pearson

Por ende, concluimos esencialmente tras dicho análisis de correlación que la calidad está alineada con el grado de alcohol. Esto generalmente será debido a que ambas variables presentan una escala (de 0 a 10 en el caso de quality y una escala porcentual con un máximo de 15% en el caso de alcohol), por lo que ambas crecen de manera similar en magnitud.

Por otro lado, vemos como la variable fixed acidity presenta correlación con citric acid, hecho que podemos explicar, ya que como hemos visto antes, esta última variable se encuentra en el interior de los ácidos no volátiles del vino y por ende de fixed acidity. Fixed acidity, al igual que citric, presentan una buena correlación con el pH (escala de acidez), siguiendo un poco la línea comentada anteriormente en relación al ácido cítrico.

Por último, comentar que citric y volatile acid, presentan un grado de correlación negativa considerable a pesar de ir contenidos en grupos de acidez distintos (volátil y no volátil).

Por ende en relación a este análisis de la correlación, necesitaremos llevar a cabo un modelo de regresión lineal para tratar de dar explicación en relación a las acideces en la calidad del vino, ya que tras dicho análisis de correlación, no disponemos de significancia para considerar tal hecho.

Modelo de regresión lineal para calidad

Tras haber llevado a cabo un análisis de correlación, vamos a ver cómo describe la acidez y cómo afecta la acidez en la explicación de la calidad. En este punto, vamos a construir un modelo de regresión lineal mediante variables predictoras cuantitativas. Los modelos de regresión lineal, basan su fundamento en la predicción basada en la siguiente ecuación:

$$Yi = (\beta 0 + \beta 1X1i + \beta 2X2i + \cdots + \beta\, nXni) + ei$$

Donde Yi es el valor real observado del que disponemos, el conjunto (β0 + β1X1i + β2X2i + ··· +βnXni) se corresponde con el término predicho por nuestro modelo e ei (el error), se asocia con el error cometido en el establecimiento de la propia regresión en una cuestión puramente geométrica del eje y.

El procedimiento de generación del modelo va a ser el siguiente. En una primera instancia vamos a tratar la aproximación de la calidad (variable dependiente), mediante predictores de acidez (fixed acidity, volatile acidity y citric acid). Para ello y tras el propio análisis de correlación, evitaremos introducir precisamente citric acid y fixed acidity en el modelo conjuntamente ya que al estar el primero contenido en la magnitud del segundo, podría presentar un sesgo en el modelo por la dependencia entre dichas variables.

procedemos por tanto con la creación de un primer modelo para explicar la calidad en base a la acidez volátil y no volátil:

```
modelo_1 <- lm(base.datos$quality ~  base.datos$volatile.acidity +
base.datos$fixed.acidity)

summary(modelo_1)

##
## Call:
## lm(formula = base.datos$quality ~ base.datos$volatile.acidity +
##     base.datos$fixed.acidity)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.85353 -0.49645 -0.01747  0.52069  2.89959
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.35999    0.14840  42.857   <2e-16 ***
## base.datos$volatile.acidity -1.68806    0.13179 -12.808   <2e-16 ***
## base.datos$fixed.acidity     0.01623    0.01422   1.142    0.254
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7649 on 1596 degrees of freedom
## Multiple R-squared:  0.1041, Adjusted R-squared:  0.1029
## F-statistic: 92.68 on 2 and 1596 DF,  p-value: < 2.2e-16
```

Tras la generación del modelo, observamos las siguientes consideraciones:

La parte volátil de la acidez explica bien en términos de significancia la calidad (p-value). Por otro lado, el nivel de significancia de la acidez no volátil cae por encima de 0.05, un valor elevado que nos indica que dicha variable no acaba de explicar bien la calidad. En este punto y dado que citric acid presenta un grado de dependencia con fixed acidity, vamos a componer un nuevo modelo con citric acid en lugar de fixed acidity para ver si de este modo el primero presenta una mayor significancia en el modelo en términos del p-value.

Por otro lado, debemos tener en cuenta que el valor del coeficiente de dependencia que hemos obtenido por parte del coeficiente de significancia R^2 es muy bajo, entorno a un 10%, pero esto lo trataremos posteriormente cuando lleguemos al modelo que más explique la calidad.

En este punto, se lleva a cabo la eliminación de fixed acidity y la introducción de citric acid:

```
modelo_2 <- lm(base.datos$quality ~  base.datos$volatile.acidity +
base.datos$citric.acid)

summary(modelo_2)

##
## Call:
## lm(formula = base.datos$quality ~ base.datos$volatile.acidity +
##     base.datos$citric.acid)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83745 -0.52077 -0.01854  0.52475  2.88694
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.31908    0.09864  64.059  < 2e-16 ***
## base.datos$volatile.acidity -1.50752    0.15159  -9.945  < 2e-16 ***
## base.datos$citric.acid        0.30344    0.11426   2.656  0.00799 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7635 on 1596 degrees of freedom
## Multiple R-squared:  0.1073, Adjusted R-squared:  0.1062
## F-statistic: 95.89 on 2 and 1596 DF,  p-value: < 2.2e-16
```

Tras observar el resultado que nos presenta summary() en base a nuestro modelo de regresión, Citric acid y volatile acidity en este caso presentan significancia en este punto en términos del p-value descrito.

A partir de aquí, y dado que el valor R^2 sigue siendo entorno al 10%, vamos a tratar de añadir variables para ajustar nuestro modelo en términos del coeficiente de determinación R^2 y por ende llevar la bondad de ajuste de nuestro modelo.

```
modelo_3 <- lm(base.datos$quality ~  base.datos$volatile.acidity +
base.datos$citric.acid + base.datos$alcohol  + base.datos$pH)

summary(modelo_3)

##
## Call:
## lm(formula = base.datos$quality ~ base.datos$volatile.acidity +
##     base.datos$citric.acid + base.datos$alcohol + base.datos$pH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.72610 -0.40400 -0.09497  0.48495  2.48111
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.04578    0.52398   7.721 2.02e-14 ***
## base.datos$volatile.acidity -1.04142    0.13978  -7.451 1.51e-13 ***
## base.datos$citric.acid       0.13972    0.11942   1.170  0.24217
## base.datos$alcohol           0.34506    0.01823  18.932  < 2e-16 ***
## base.datos$pH               -0.45365    0.15070  -3.010  0.00265 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.69 on 1594 degrees of freedom
## Multiple R-squared:  0.2717, Adjusted R-squared:  0.2699
## F-statistic: 148.7 on 4 and 1594 DF,  p-value: < 2.2e-16
```

Modelo de regresión logística:

En este punto, tras el modelo de regresión lineal creado y debido a la poca significancia de este en la explicación de la calidad del vino, hemos decidido trasladar el enfoque de la calidad a un caso logístico. Partiendo de las diferencias entre ambas tipologías de regresiones sobre todo en términos del enfoque inicial del modelo, en este caso pasamos de la voluntad de predecir/explicar un valor de calidad con respecto a los predictores anteriores a la voluntad de clasificar, en función de los mismos predictores (o similares una

vez creemos el modelo y veamos la significación que cada uno aporta al modelo) el valor de calidad en términos de probabilidad, en base a inputs de las diferentes variables predictoras.

El modelo logístico se establece de la siguiente manera:

$$\text{logit(calidad)} = \beta 0 + \beta 1 X1i + \beta 2 X2i + \cdots + \beta \, nXni$$

Por ende, la probabilidad, la podemos definir con la siguiente expresión:

$$P(\text{calidad}) = \exp(\beta 0 + \beta 1 X1i + \beta 2 X2i + \cdots + \beta \, nXni) \,/\, 1 + \exp(\beta 0 + \beta 1 X1i + \beta 2 X2i + \cdots + \beta \, nXni)$$

En este punto, ya solo nos queda en un primer paso establecer/crear una variable dicotómica binomial en función de los valores de calidad para poder dotar la creación del modelo logístico. Para ello crearemos la variable dicotómica para calidad "calidad_BM" en base a los diferentes valores originarios de "quality". la manipulación de la variable se hará de la siguiente manera:

Vino bueno. Rango [5, 10], recibirá un valor 1.

Vino Malo; Rango [0, 5), recibirá un valor 0.

El procedimiento es el siguiente:

```
# Creo la nueva columna con valores iguales a quality:
base.datos$calidad_BM <- base.datos$quality

# Defino 1 y 0 en función del valor del quality:
for (i in 1:nrow(base.datos)){
  if (base.datos$calidad_BM[i] < 5){
    base.datos$calidad_BM[i] = 0
  }
  if (base.datos$calidad_BM[i] >= 5){
    base.datos$calidad_BM[i] = 1
  }
}

# Defino el factor:
base.datos$calidad_BM <- factor(base.datos$calidad_BM)

# Mostramos algunas líneas de como se ha transformado la variable:
head(base.datos$calidad_BM)

## [1] 1 1 1 1 1 1
## Levels: 0 1

# Mostramos como se han repartido los niveles 1 y 0:
summary(base.datos$calidad_BM)

##    0    1
##   63 1536
```

Tras la creación de la variable dicotómica, es importante verificar la independencia con respecto a las variables que introduciremos en el modelo como predictoras. En este caso, haremos uso del test Chi cuadrado para llevar a cabo una prueba de independencia entre cada una de las variables y de este modo observar si existe asociación entre las variables.

Test $\chi 2$ de Pearson:

Ho: No hay asociación entre las variables A|B (Las variables son independientes)

H1: Si hay asociación entre las variables A|B (Las variables no son independientes)

```
# Test Chi-Cuadrado:

# Variable dicotómica vs fixed acidity:
chisq.test(x = table(base.datos$calidad_BM,base.datos$fixed.acidity))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM,
## base.datos$fixed.acidity)): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$fixed.acidity)
## X-squared = 91.683, df = 72, p-value = 0.05876
```

```
# Variable dicotómica vs volatile acidity:
chisq.test(x = table(base.datos$calidad_BM,base.datos$volatile.acidity))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM,
## base.datos$volatile.acidity)): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$volatile.acidity)
## X-squared = 112.29, df = 7, p-value < 2.2e-16
```

```
# Variable dicotómica vs citric acid:
chisq.test(x = table(base.datos$calidad_BM,base.datos$citric.acid))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM,
base.datos$citric.acid)):
## Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$citric.acid)
## X-squared = 58.055, df = 9, p-value = 3.172e-09
```

```
# Variable dicotómica vs residual sugar:
chisq.test(x = table(base.datos$calidad_BM,base.datos$residual.sugar))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM,
## base.datos$residual.sugar)): Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$residual.sugar)
## X-squared = 45.466, df = 20, p-value = 0.0009535
```

```
# Variable dicotómica vs alcohol:
chisq.test(x = table(base.datos$calidad_BM,base.datos$alcohol))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM,
base.datos$alcohol)):
## Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$alcohol)
## X-squared = 67.055, df = 46, p-value = 0.02298
```

```
# Variable dicotómica vs pH:
chisq.test(x = table(base.datos$calidad_BM,base.datos$pH))
```

```
## Warning in chisq.test(x = table(base.datos$calidad_BM, base.datos$pH)):
Chi-
## squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  table(base.datos$calidad_BM, base.datos$pH)
## X-squared = 17.947, df = 7, p-value = 0.01221
```

Al llevar a cabo el test Chi-Squared en una prueba de independencia de cada una de las variables predictoras con respecto a calidad_BM, observamos como en la mayoría de los casos el p-value obtenido es inferior a un nivel de significancia 0.05 (excepto en fixed acitidy donde roza dicho nivel y si se puede identificar una No dependencia con respecto a calidad_BM) lo que nos lleva a determinar que dichas variables predictoras están estadísticamente asociadas con las variable calidad_BM ya que podemos rechazar la hipótesis nula (H0) en casi todos los casos.

Modelo de regresión logística cuantitativo:

```
# Definición del modelo:
modelo_glm_1 <- glm(base.datos$calidad_BM ~ base.datos$volatile.acidity +
base.datos$citric.acid + base.datos$fixed.acidity + base.datos$alcohol  +
base.datos$pH + base.datos$residual.sugar, family = binomial (link = logit))
```

```
# Representación del modelo:
summary(modelo_glm_1)

##
## Call:
## glm(formula = base.datos$calidad_BM ~ base.datos$volatile.acidity +
##     base.datos$citric.acid + base.datos$fixed.acidity + base.datos$alcohol
+
##     base.datos$pH + base.datos$residual.sugar, family = binomial(link =
logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1579   0.1736   0.2456   0.3161   0.6403
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   12.7033     4.9607   2.561   0.0104 *
## base.datos$volatile.acidity  -1.2277     1.1200  -1.096   0.2730
## base.datos$citric.acid        2.5420     1.1463   2.218   0.0266 *
## base.datos$fixed.acidity     -0.2563     0.1457  -1.759   0.0785 .
## base.datos$alcohol            0.1379     0.1492   0.924   0.3553
## base.datos$pH                -3.2013     1.2836  -2.494   0.0126 *
## base.datos$residual.sugar     0.9227     0.3888   2.373   0.0176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.97  on 1598  degrees of freedom
## Residual deviance: 496.29  on 1592  degrees of freedom
## AIC: 510.29
##
## Number of Fisher Scoring iterations: 6
```

Tras obtener y observar el resultado del modelo, observamos como volatile acidity y alcohol nos presentan niveles de significancia que se salen de los márgenes de buena aproximación para nuestro modelo. (valores de p value superiores a 0.05). En este punto, decidimos eliminar dichos predictores del modelo y quedarnos con el resto en un modelo nuevo:

```
# Definición del modelo:
modelo_glm_2 <- glm(base.datos$calidad_BM ~ base.datos$citric.acid +
base.datos$fixed.acidity  + base.datos$pH + base.datos$residual.sugar, family
= binomial (link = logit))

# Representación del modelo:
summary(modelo_glm_2)
```

```
##
## Call:
## glm(formula = base.datos$calidad_BM ~ base.datos$citric.acid +
##     base.datos$fixed.acidity + base.datos$pH + base.datos$residual.sugar,
##     family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2228   0.1818   0.2467   0.3194   0.6113
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 12.7875     4.8269   2.649 0.008068 **
## base.datos$citric.acid       3.3968     0.9850   3.449 0.000563 ***
## base.datos$fixed.acidity    -0.2909     0.1422  -2.045 0.040810 *
## base.datos$pH               -2.9593     1.2690  -2.332 0.019700 *
## base.datos$residual.sugar    0.9045     0.3834   2.359 0.018317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.97  on 1598  degrees of freedom
## Residual deviance: 498.65  on 1594  degrees of freedom
## AIC: 508.65
##
## Number of Fisher Scoring iterations: 6
```

Los resultados que obtenemos en este nuevo modelo son favorables en términos del nivel de significancia individuales de cada una de las variables Aún y así, a expensas de verificar la validez de dicho modelo en la predicción para la probabilidad de la calidad, vamos a llevar a cabo la evaluación del modelo en base a los siguientes tests de verificación:

Likelihood ratio:

```
# Diferencia de residuos
dif_residuos <- modelo_glm_2$null.deviance - modelo_glm_2$deviance

# Grados libertad
df <- modelo_glm_2$df.null - modelo_glm_2$df.residual
# p-value
p_value <- pchisq(q = dif_residuos,df = df, lower.tail = FALSE)

paste("Diferencia de residuos:", round(dif_residuos, 4))

## [1] "Diferencia de residuos: 32.3232"

paste("Grados de libertad:", df)

## [1] "Grados de libertad: 4"
```

```
paste("p-value:", p_value)
```

```
## [1] "p-value: 1.64309868965872e-06"
```

El p-value obtenido (p-value = 1.64309868965872e-06) nos indica que el modelo entendido en su conjunto es significativo. Todo esto en base a la totalidad de los predictores aglutinados en base a la variable dependiente y a diferencia del summary() del modelo donde se considera la significatividad separada por cada variable.

Por otro lado y de manera más robusta en esta caso que el test de Likelihood, podemos evaluar la bondad de ajuste de nuestro modelo de regresión logística mediante el test de Hosmer-Lemeshow.

Dicho test, se usa para comparar los valores previstos (esperados) por el modelo con los valores observados basándose en el siguiente contraste de hipótesis:

HO: No hay diferencias entre los valores observados y los pronosticados. EL modelo está bien ajustado!

H1: Hay diferencias entre los valores observados y los pronosticados. El modelo NO está bien ajustado.

La hipótesis nula del test de Hosmer-Lemeshow es que no hay diferencias entre los valores observados y los valores pronosticados (el rechazo este test indicaría que el modelo no está bien ajustado).

```
#install.packages("ResourceSelection") # Instalo el paquete si no dispongo de él
library(ResourceSelection)
```

```
## Warning: package 'ResourceSelection' was built under R version 4.0.5
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
# Definición del test:
y <- base.datos$calidad_BM
hoslem.test(modelo_glm_2$y, fitted(modelo_glm_2))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  modelo_glm_2$y, fitted(modelo_glm_2)
## X-squared = 7.2232, df = 8, p-value = 0.5128
```

El p-value obtenido es de 0.51, y por ende superior a un valor típico de significación de 0.05, lo que nos indica que el modelo está bien ajustado ya que no podemos rechazar la hipótesis nula!

Por tanto y como conclusión, encontramos que el modelo logístico formado por las variables predictoras cuantitativas base.datos$citric.acid + base.datos$fixed.acidity + base.datos$pH + base.datos$residual.sugar aproxima bien la calidad en términos de

probabilidad y por ende, dados inputs de estas variables, podremos determinar en términos probabilísticos (entre 0 y 1) que calidad tenemos en un vino.

Ejemplo de aplicación. Predicción del modelo logístico:

```
# Coeficientes del modelo :
coefficients(modelo_glm_2)
```

```
##               (Intercept)    base.datos$citric.acid
base.datos$fixed.acidity
##                12.7875333                 3.3968270
-0.2908968
##             base.datos$pH base.datos$residual.sugar
##                -2.9592687                 0.9045245
```

El resultado que obtenemos lo podemos ver de la siguiente forma:

logit(calidad_BM)= 12.7875333 - 0.2908968 * fixed.acidity + 0.9045245 * residual.sugar + 3.3968270 * citric.acid -2.9592687 * pH

Por ende, la probabilidad, la podemos definir con la siguiente expresión:

P(delay_sfp) = exp(12.7875333 - 0.2908968 * fixed.acidity + 0.9045245 * residual.sugar + 3.3968270 * citric.acid -2.9592687 * pH) / 1 + exp(12.7875333 - 0.2908968 * fixed.acidity + 0.9045245 * residual.sugar + 3.3968270 * citric.acid -2.9592687 * pH)

Donde operando para diferentes valores de cada una de las variables como ejemplo cogidos de los puntos más centrales a la media independiente de cada uno de ellos, obtenemos el valor de probabilidad:

```
fixed <- 8
sugar <- 2.1
citric <- 0.2
ph <- 3

exponente <- exp(12.7875333 + (-0.2908968 * fixed) + 0.9045245 * sugar +
3.3968270 * citric  + (-2.9592687 * ph))

P_calidad = ((exponente) / (1 + exponente))
paste("La probabilidad de obtener un vino de calidad con los valores
introducidos es de : ", P_calidad * 100, " %")
```

```
## [1] "La probabilidad de obtener un vino de calidad con los valores
introducidos es de :  98.4653761151453  %"
```

En este punto, vemos que con valores medios de cada una de las variables del modelo, aplicando nuestro modelo de regresión logística, observamos cómo podemos obtener una calidad en el vino.

Dicha predicción, se entiende como la probabilidad de que valores introducidos como variables predictoras, muestran un valor 1 en la variable “calidad_BM”, es decir P(X = x| 1),

siendo X la variable aleatoria que define el conjunto total de Predictores introducidos y 1 el valor que toma la varible “Calidad_BM” en la predicción llevada a cabo.

Aún y así, aunque la predicción haya sido buena en términos de probabilidad obtenida y la bondad de ajuste por Holmer y LikeliHood nos determinan un modelo ajustado en la predicción de la calidad del vino (en función de dicha variable dicotómica establecida) a partir de los predictores: fixed_acidity, residual_sugar, citric_acidity y pH, vamos a contrastar y ratificar dicha predicción mediante un contraste de hipótesis sobre los valores que hemos establecido en la variable Calidad_BM.

CONTRASTE DE HIPÓTESIS

Siguiendo con la línea expuesta desde el inicio del análisis, a la que vamos a sumar las variables significativas que acabamos de observar a lo largo de los modelos de regresión, vamos a tratar la creencia de que los vinos de calidad (Calidad_BM = 1 o quality >= 5) tienen una mayor acidez que los vinos de calidad baja (Calidad_BM = 0 o quality < 5). Para ello, veremos si los vinos de calidad disponen de mejor acidez NO volátil (fixed.acidity), ácido cítrico (citric.acid) que los vinos de una calidad baja.

En este apartado se va a llevar a cabo un contraste de hipótesis de dos muestras independientes sobre el valor medio de fixed.acidity/citric.acid/volatile.acidity de una muestra de vinos de calidad y de una muestra de vinos de baja calidad con varianzas desconocidas por lo que deberemos, previamente a la realización del contraste de hipótesis, llevar a cabo un Test de varianzas para observar si las varianzas de ambas muestras son iguales o no con el fin de aplicar el estadístico correcto en el posterior contraste de hipótesis sobre la media de las variables.

El primer paso a llevar a cabo previo a cualquier tipo de contraste de hipótesis en este caso es la obtención de las muestras de los vinos que presentan un valor más alto y más bajo de calidad, en función del valor de calidad indistintamente:

```
# Establezco una condición de calidad:
condicion_calidad <- base.datos$quality >= 5

# Aplico la condición de calidad sobre la muestra inicial y obtengo las dos
muestras independientes:
alta_calidad <- base.datos[condicion_calidad, ]
baja_calidad <- base.datos[!condicion_calidad, ]

paste("La muestra de ALTA calidad contiene n = ",nrow(alta_calidad), "
campos")

## [1] "La muestra de ALTA calidad contiene n =  1536  campos"

paste("La muestra de BAJA calidad contiene n = ",nrow(baja_calidad), "
campos")

## [1] "La muestra de BAJA calidad contiene n =  63  campos"
```

Una vez disponemos de las muestras y del valor n de cada una de ellas, procedemos con la pregunta de investigación.

La estructura de dichas preguntas cobra importancia ya que van a dar pié al establecimiento de las hipótesis (nula y alternativa) correspondientes y de fijar en cierta manera qué tipología de contraste (unilateral, bilateral) vamos a tener. Las preguntas son por tanto:

Variable fixed.acidity: ¿El valor medio de la variable fixed.acidity de los vinos de calidad es significativamente mayor al valor medio de la variable fixed.acidity de los vinos de baja calidad?

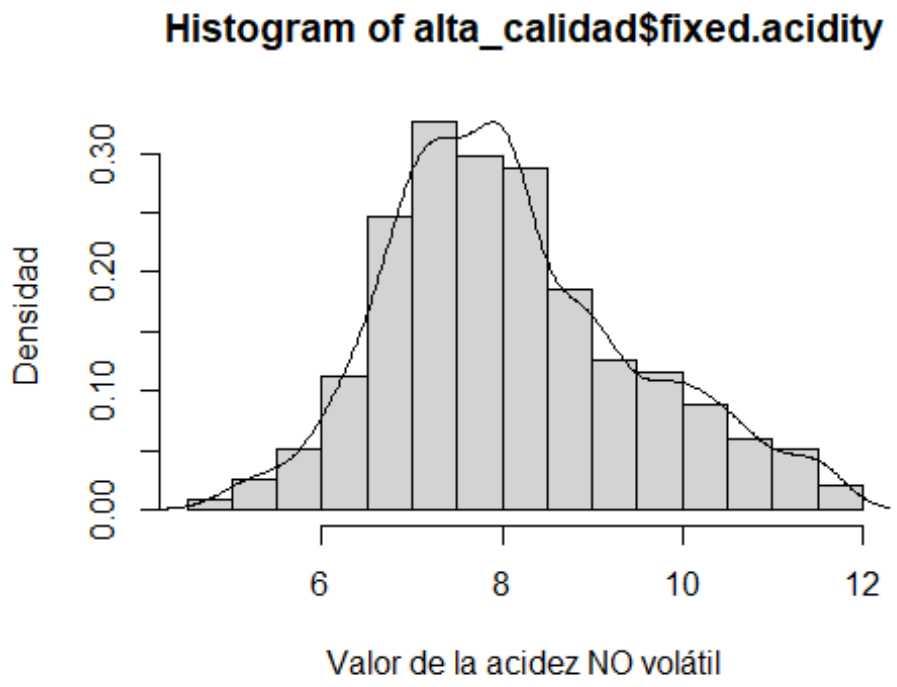
Variable citric.acid: ¿El valor medio de la variable citric.acid de los vinos de calidad es significativamente mayor al valor medio de la variable citric.acid de los vinos de baja calidad?

Variable volatile.acidity: ¿El valor medio de la variable volatile.acidity de los vinos de calidad es significativamente mayor al valor medio de la variable volatile.acidity de los vinos de baja calidad?

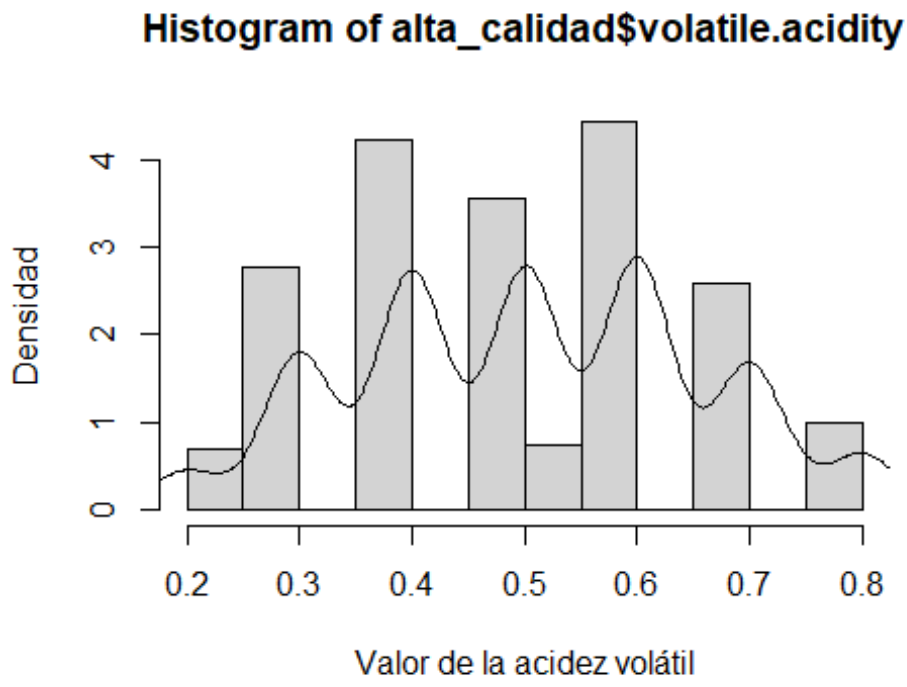
Queremos observar en cada caso, si el valor medio de cada variable para la muestra de buenos de calidad es mayor al valor medio de cada variable para la muestra de vinos de baja calidad, lo que nos llevará a establecer un contraste de hipótesis Unilateral por la derecha como veremos más adelante.

El siguiente punto sería la observación de la distribución de frecuencias de cada una de las variables en cada una de las muestras (alta_calidad y baja_calidad respectivamente) para observar si podemos asumir normalidad en las distribuciones y aplicar el test estadístico correspondiente y/o asumir la teoría del Teorema del Límite central para evitar llevar a cabo un test no paramétrico en caso de no normalidad.

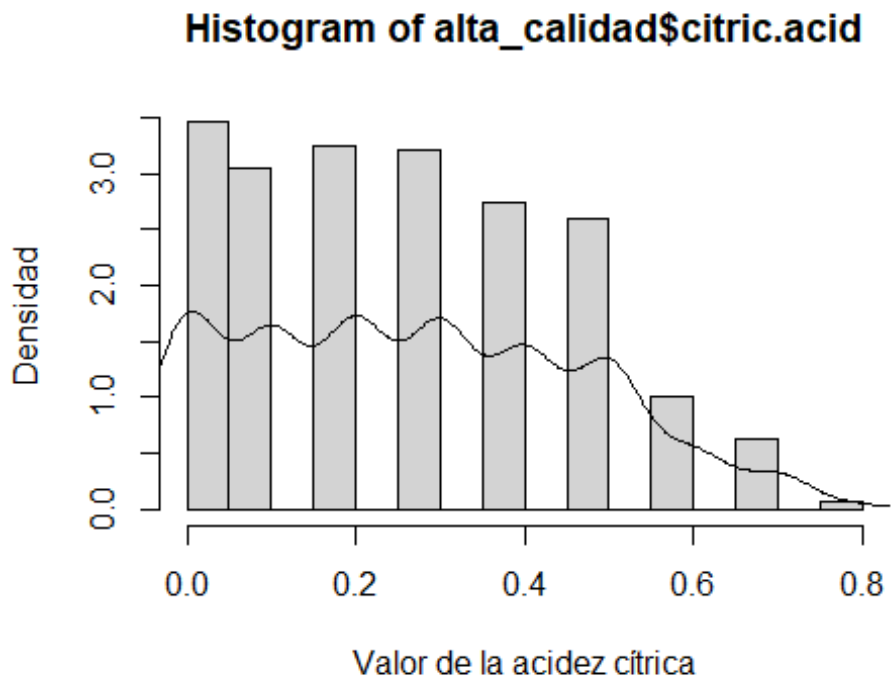
```
# ALTA CALIDAD:

# fixed.acidity
hist(alta_calidad$fixed.acidity, freq = F, ylab = "Densidad", xlab = "Valor
de la acidez NO volátil")
lines(density(alta_calidad$fixed.acidity))
```



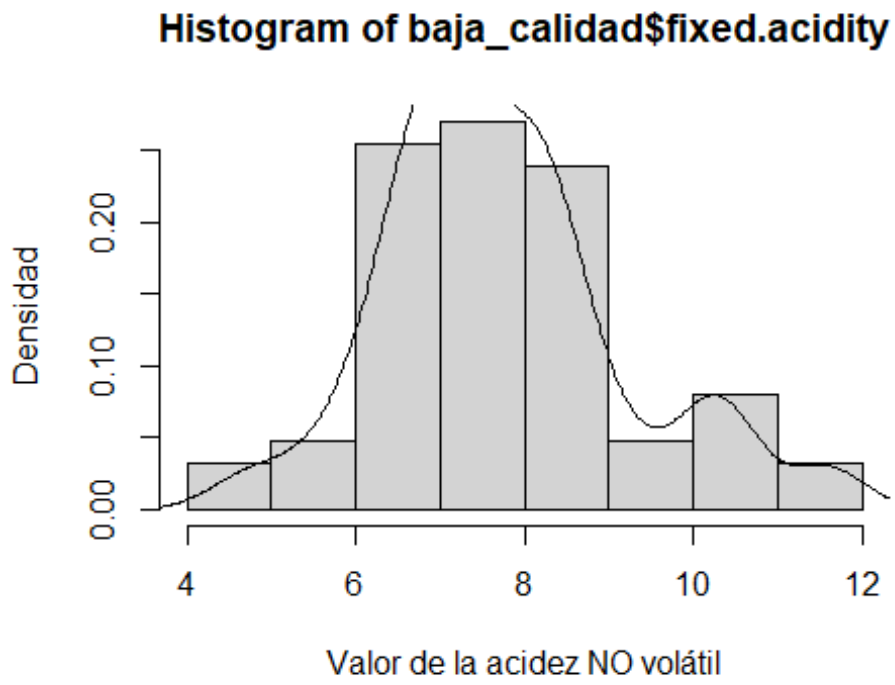

```r
# volatile.acidity
hist(alta_calidad$volatile.acidity, freq = F, ylab = "Densidad", xlab =
"Valor de la acidez volátil")
lines(density(alta_calidad$volatile.acidity))
```



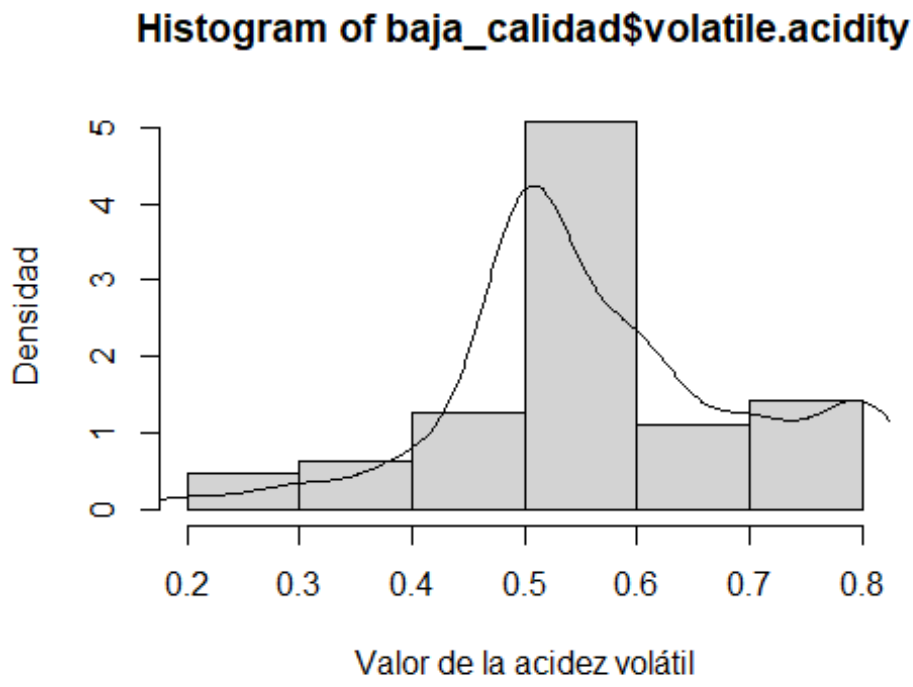

```
# citric.acid
hist(alta_calidad$citric.acid, freq = F, ylab = "Densidad", xlab = "Valor de
la acidez cítrica")
lines(density(alta_calidad$citric.acid))
```



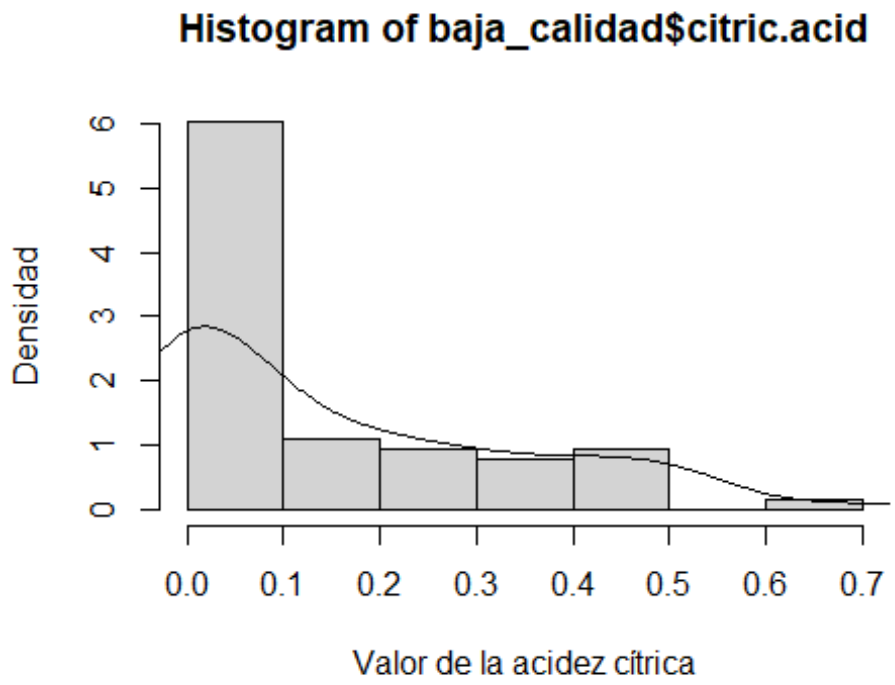

```
# BAJA CALIDAD:

# fixed.acidity
hist(baja_calidad$fixed.acidity, freq = F, ylab = "Densidad", xlab = "Valor
de la acidez NO volátil")
lines(density(baja_calidad$fixed.acidity))
```




```
# volatile.acidity
hist(baja_calidad$volatile.acidity, freq = F, ylab = "Densidad", xlab =
"Valor de la acidez volátil")
lines(density(baja_calidad$volatile.acidity))
```




```r
# citric.acid
hist(baja_calidad$citric.acid, freq = F, ylab = "Densidad", xlab = "Valor de la acidez cítrica")
lines(density(baja_calidad$citric.acid))
```



A pesar de únicamente haber mostrado los diagramas de frecuencias, podemos observar como un valor de Shapiro.test o cualquier otro test de normalidad, nos conlleva a identificar una distribución NO normal…

Para evitar la realización de Tests no paramétricos para llevar a cabo el contraste de hipótesis y todo lo que ello comporta, haremos cita de y uso del TEOREMA DEL LÍMITE CENTRAL. Dicho teorema establece que el contraste de hipótesis sobre la media de una muestra (que es precisamente el hecho que estamos evaluando en nuestro caso con las muestras de alta_calidad y baja_calidad) se aproxima a una distribución normal aunque la población original no siga dicha distribución (pues lo hemos visto con la representación de la densidad), siempre que el tamaño de la muestra "n" sea suficientemente grande, esto es, superior a 30 elementos.

En nuestro caso, las muestras disponen de unos valores de n de 63 y 1536 respectivamente para baja_calidad y alta_calidad tal y como ya se ha mostrado con anterioridad. Por tanto, la aplicación del Teorema del Límite Central es válida y asumimos que las distribuciones se pueden aproximar a una distribución Normal para el contraste. Esto nos facilita el proceso significativamente…

Una vez tenemos claras y definidas las muestras, las distribuciones que siguen las variables que vamos a analizar dentro de la muestra y las preguntas de investigación para cada variable sobre la que vamos a constatar en contraste de hipótesis, procedemos con el establecimiento de las hipótesis Nula y alternativa.

H0: Media de los valores de fixed.acidity/volatile.acidity/citric.acid en alta_calidad = Media de los valores de fixed.acidity/volatile.acidity/citric.acid en baja_calidad

H1: Media de los valores de fixed.acidity/volatile.acidity/citric.acid en alta_calidad > Media de los valores de fixed.acidity/volatile.acidity/citric.acid en baja_calidad

Como vemos, nos encontramos frente un contraste unitaleral por la derecha. En este punto, procedemos con el contraste de hipótesis.

Llegados a este punto, vamos a proceder con el dicho contraste de hipótesis bajo las consideraciones que hemos ido justificado en los pasos previos de este ejercicio. Antes, debido al desconocimiento de la varianza de las muestras, debemos proceder con un Test de Varianzas para cada muestra. Para aplicar el estadístico adecuado, hay que comprobar si las varianzas de las dos poblaciones son iguales o no. Para ello, aplicamos primero el test de igualdad de varianzas.

En este punto vamos a suponer una hipótesis nula y alternativa sobre el valor de la varianza de cada población sobre cada una de las 3 variables:

H0: (varianza alta_calidad[fixed.acidity/volatile.acidity/citric.acid]) = (varianza baja_calidad[fixed.acidity/volatile.acidity/citric.acid])

H1: (varianza alta_calidad[fixed.acidity/volatile.acidity/citric.acid]) != (varianza baja_calidad[fixed.acidity/volatile.acidity/citric.acid])

En este punto procederemos con el Test de Varianzas para cada una de las variables anteriores sobre ambas muestras:

```
funcion_varianzas <- function(alta, baja){

   # Definición de variables necesarias para el cómputo:
   alfa = 0.05
   mean1 <- mean(alta) # media de la variable en la muestra alta calidad
   mean2 <- mean(baja) # media de la variable en la muestra baja calidad
   n1 <- length(alta)
   n2 <- length(baja)
   s1 <- sd(alta) # d.estándar
   s2 <- sd(baja) # d.estándar

   # Muestro por pantalla las variables anteriores:

print("Mean_Alta|Mean_Baja|S_Alta|S_Baja|Total_muestra_Alta|Total_muestra_Baj
a")
   print(c(mean1, mean2, s1, s2, n1, n2))


   # Calculo estadístico y valores de intérvalo inferior y superior:
   fobs = s1^2/s2^2
   fcritL <- qf(alfa/2, df1=n1-1, df2=n2-2)
   fcritU <- qf(1-alfa/2, df1=n1-1, df2=n2-2)
```

```
   pvalue <- min(pf(fobs, df1=n1-1, df2=n2-2, lower.tail = FALSE), pf(fobs,
df1=n1-1, df2=n2-2))*2

   # Muestro el resultado por pantalla:
   print("Fobs|LOW|UPPER|P_VALUE")
   print(c( fobs, fcritL, fcritU, pvalue))

print("##############################################################################
######################")

}


# EJECUCIÓN DE LA FUNCIÓN PARA CADA VARIABLE:

# FIXED ACIDITY:
funcion_varianzas(alta_calidad$fixed.acidity, baja_calidad$fixed.acidity)

## [1]
"Mean_Alta|Mean_Baja|S_Alta|S_Baja|Total_muestra_Alta|Total_muestra_Baja"
## [1]    8.102953    7.739321    1.390090    1.443420 1536.000000
63.000000
## [1] "Fobs|LOW|UPPER|P_VALUE"
## [1] 0.9274704 0.7161723 1.4853285 0.6390661
## [1]
"##############################################################################
###############"

# VOLATILE ACIDITY:
funcion_varianzas(alta_calidad$volatile.acidity,
baja_calidad$volatile.acidity)

## [1]
"Mean_Alta|Mean_Baja|S_Alta|S_Baja|Total_muestra_Alta|Total_muestra_Baja"
## [1]    0.5040221    0.5702540    0.1503907    0.1343978 1536.0000000
## [6]   63.0000000
## [1] "Fobs|LOW|UPPER|P_VALUE"
## [1] 1.2521543 0.7161723 1.4853285 0.2632244
## [1]
"##############################################################################
###############"

# CITRIC ACID:
funcion_varianzas(alta_calidad$citric.acid, baja_calidad$citric.acid)

## [1]
"Mean_Alta|Mean_Baja|S_Alta|S_Baja|Total_muestra_Alta|Total_muestra_Baja"
## [1]    0.2704427    0.1566025    0.1987225    0.1860324 1536.0000000
## [6]   63.0000000
## [1] "Fobs|LOW|UPPER|P_VALUE"
## [1] 1.1410815 0.7161723 1.4853285 0.5203119
```

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

En todos los casos el valor del estadístico fobs cae dentro del intérvalo de confianza del 95% además de que el valor del valor p (p value) en todos los casos es mayor que el nivel de significancia. Estos hechos nos llevan a aceptar la hipótesis nula y por tanto a disponer de varianzas iguales - HOMOCEDASTICIDAD.

En este punto nos disponemos a realizar el contraste de hipótesis ciertamente sobre el valor de las variables mediante la función t.test() una vez hemos conocido que debemos tratar con varianzas desconocidas diferentes (var.equal = TRUE) y por ende con el estadístico que este supone implementar de acuerdo al contraste unilateral (greater) que deseamos implementar.

```
# Definición de los contrastes de hipótesis:

# Contraste acidez NO volátil:
t.test(alta_calidad$fixed.acidity,
baja_calidad$fixed.acidity,alternative="greater", var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  alta_calidad$fixed.acidity and baja_calidad$fixed.acidity
## t = 2.0319, df = 1597, p-value = 0.02116
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.06909517        Inf
## sample estimates:
## mean of x mean of y
##  8.102953  7.739321

# Contraste acidez volátil:
t.test(alta_calidad$volatile.acidity,
baja_calidad$volatile.acidity,alternative="greater", var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  alta_calidad$volatile.acidity and baja_calidad$volatile.acidity
## t = -3.4395, df = 1597, p-value = 0.9997
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.09792419         Inf
## sample estimates:
## mean of x mean of y
## 0.5040221 0.5702540
```

```r
# Contraste acidez cítrica:
t.test(alta_calidad$citric.acid,
baja_calidad$citric.acid,alternative="greater", var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  alta_calidad$citric.acid and baja_calidad$citric.acid
## t = 4.4672, df = 1597, p-value = 4.241e-06
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.07189911        Inf
## sample estimates:
## mean of x mean of y
## 0.2704427 0.1566025
```

En este punto y tras obtener el resultados de los contrastes podemos concluir con lo siguiente:

Para un nivel de significancia del 0.05, las variables fixed acidity (p-value = 0.02) y citric acid (p-value <<<< 0) como vemos presentan un p-value menor al nivel de significancia (muy considerable para citric acid) que nos lleva a rechazar la hipótesis nula y por ende a disponer que el valor medio de los valores de fixed.acidity y citric.acid en la muestra de alta_calidad es superior al valor medio de fixed.acidity y citric.acid en la muestra baja_calidad.

Esta conclusión, ratifica la consideración de estas dos variables en la predicción llevada a cabo mediante el modelo de regresión logístico anterior àra la predicción de la calidad del vino, donde ya vimos, que estas dos variables aportan significancia al modelo y por tanto eran buenos predictores de calidad.

Por otro lado, el p-value del contraste para la variable volatile.acidity, es superior al nivel de significancia, lo que nos lleva a aceptar la hipótesis nula y descartar que el valor medio de volatile.acidity es superior en vinos de buena calidad en comparación con vinos de una calidad inferior.

Como conclusión tras todas las pruebas llevadas a cabo, podemos interpretar como la variable fixed.acidity y la variable citric.acid (esta última contenida dentro del mismo valor de acidez NO volátil del vino y por ende de fixed.acidity) se asocian con la interpretación de la calidad del vino, pudiendo influir significativamente en esta y ratificando la idea general presentada en la introducción de la actividad cuando se expuso acerca de la calidad del vino.

5. REPRESENTACIÓN DE LOS RESULTADOS

A continuación vamos a mostrar nuestro conjunto de datos en este punto de la actividad una vez llevados a cabo la totalidad de la manipulación de este para las diferentes pruebas estadísticas.

Nuestro Dataset al completo, en términos de tipología, valor tratado de las variables y representación de la frecuencia de cada una de ellas es el siguiente:

```
# Observación de nuestro Dataset final:
head(base.datos, 30)
```

```
##    quality fixed.acidity volatile.acidity citric.acid residual.sugar
alcohol
## 1        5           7.4        0.7000000         0.0       1.900000
9.4
## 2        5           7.8        0.5066316         0.0       2.600000
9.8
## 3        5           7.8        0.8000000         0.0       2.300000
9.8
## 4        6          11.2        0.3000000         0.6       1.900000
9.8
## 5        5           7.4        0.7000000         0.0       1.900000
9.4
## 6        5           7.4        0.7000000         0.0       1.800000
9.4
## 7        5           7.9        0.6000000         0.1       1.600000
9.4
## 8        7           7.3        0.7000000         0.0       1.200000
10.0
## 9        7           7.8        0.6000000         0.0       2.000000
9.5
## 10       5           7.5        0.5000000         0.4       2.136429
10.5
## 11       5           6.7        0.6000000         0.1       1.800000
9.2
## 12       5           7.5        0.5000000         0.4       2.136429
10.5
## 13       5           5.6        0.6000000         0.0       1.600000
9.9
## 14       5           7.8        0.6000000         0.3       1.600000
9.1
## 15       5           8.9        0.6000000         0.2       2.136429
9.2
## 16       5           8.9        0.6000000         0.2       2.136429
9.2
## 17       7           8.5        0.3000000         0.6       1.800000
10.5
## 18       5           8.1        0.6000000         0.3       1.700000
9.3
## 19       4           7.4        0.6000000         0.1       2.136429
9.0
## 20       6           7.9        0.3000000         0.5       1.800000
9.2
## 21       6           8.9        0.2000000         0.5       1.800000
9.4
```

```
## 22       5           7.6         0.4000000         0.3       2.300000
9.7
## 23       5           7.9         0.4000000         0.2       1.600000
9.5
## 24       5           8.5         0.5000000         0.1       2.300000
9.4
## 25       6           6.9         0.4000000         0.1       2.400000
9.7
## 26       5           6.3         0.4000000         0.2       1.400000
9.3
## 27       5           7.6         0.4000000         0.2       1.800000
9.5
## 28       5           7.9         0.4000000         0.2       1.600000
9.5
## 29       5           7.1         0.7000000         0.0       1.900000
9.4
## 30       6           7.8         0.6000000         0.0       2.000000
9.8
##     pH calidad_BM
## 1  3.5          1
## 2  3.2          1
## 3  3.3          1
## 4  3.2          1
## 5  3.5          1
## 6  3.5          1
## 7  3.3          1
## 8  3.4          1
## 9  3.4          1
## 10 3.4          1
## 11 3.3          1
## 12 3.4          1
## 13 3.6          1
## 14 3.3          1
## 15 3.2          1
## 16 3.2          1
## 17 3.3          1
## 18 3.1          1
## 19 3.4          0
## 20 3.0          1
## 21 3.4          1
## 22 3.5          1
## 23 3.2          1
## 24 3.2          1
## 25 3.4          1
## 26 3.3          1
## 27 3.3          1
## 28 3.2          1
## 29 3.5          1
## 30 3.4          1
```

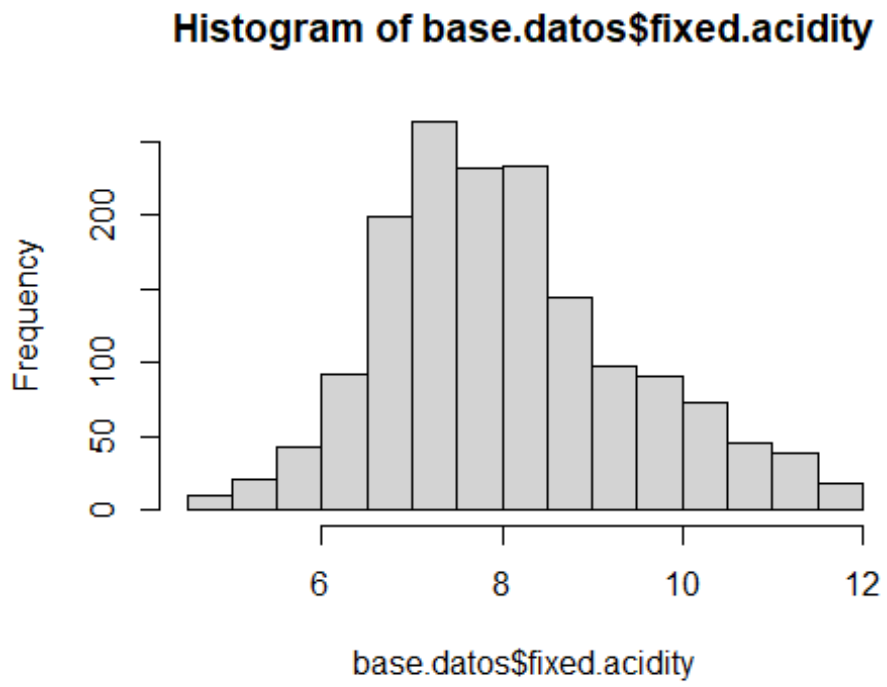
```r
# Tipología de datos de cada una de las variables:
str(base.datos)
```

```
## 'data.frame':    1599 obs. of  8 variables:
##  $ quality         : num  5 5 5 6 5 5 5 7 7 5 ...
##  $ 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 0.507 0.8 0.3 0.7 ...
##  $ citric.acid     : num  0 0 0 0.6 0 0 0.1 0 0 0.4 ...
##  $ residual.sugar  : num  1.9 2.6 2.3 1.9 1.9 ...
##  $ alcohol         : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ pH              : num  3.5 3.2 3.3 3.2 3.5 3.5 3.3 3.4 3.4 3.4 ...
##  $ calidad_BM      : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

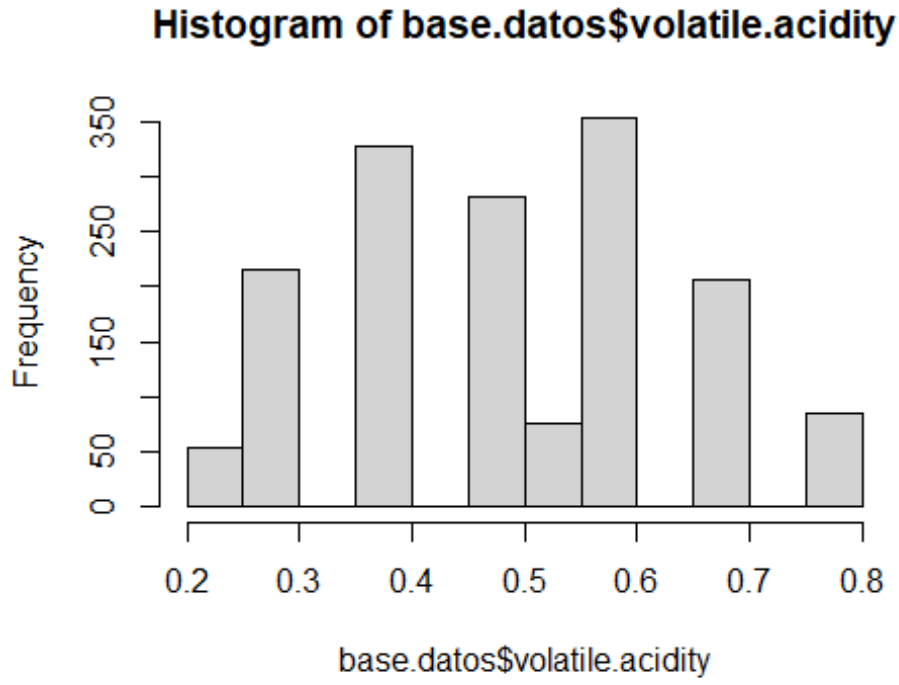
```r
# Summary de la información estadíscica bácica de cada variable:
summary(base.datos)
```

```
##     quality      fixed.acidity    volatile.acidity  citric.acid
##  Min.   :3.000   Min.   : 4.600   Min.   :0.2000   Min.   :0.000
##  1st Qu.:5.000   1st Qu.: 7.100   1st Qu.:0.4000   1st Qu.:0.100
##  Median :6.000   Median : 7.900   Median :0.5000   Median :0.300
##  Mean   :5.636   Mean   : 8.089   Mean   :0.5066   Mean   :0.266
##  3rd Qu.:6.000   3rd Qu.: 8.900   3rd Qu.:0.6000   3rd Qu.:0.400
##  Max.   :8.000   Max.   :11.800   Max.   :0.8000   Max.   :0.800
##  residual.sugar     alcohol           pH        calidad_BM
##  Min.   :1.200   Min.   : 8.40   Min.   :3.000   0:  63
##  1st Qu.:1.900   1st Qu.: 9.50   1st Qu.:3.200   1:1536
##  Median :2.136   Median :10.20   Median :3.300
##  Mean   :2.136   Mean   :10.38   Mean   :3.308
##  3rd Qu.:2.300   3rd Qu.:11.00   3rd Qu.:3.400
##  Max.   :3.100   Max.   :13.10   Max.   :3.600
```

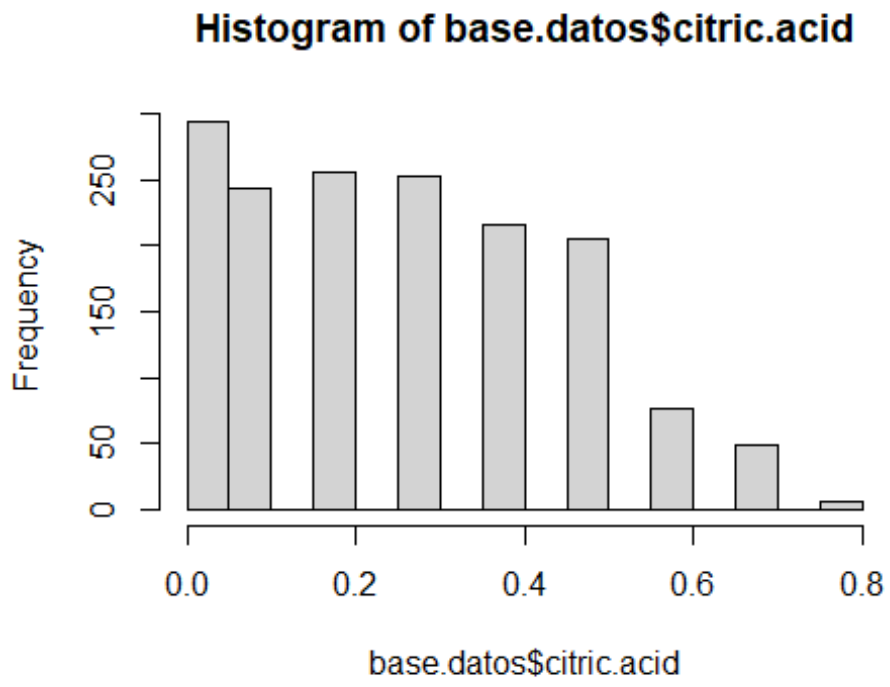
```r
# Representación en frecuencias (histogramas) de cada una de las variables
que numéricas conforman el dataset :
hist(base.datos$fixed.acidity)
```



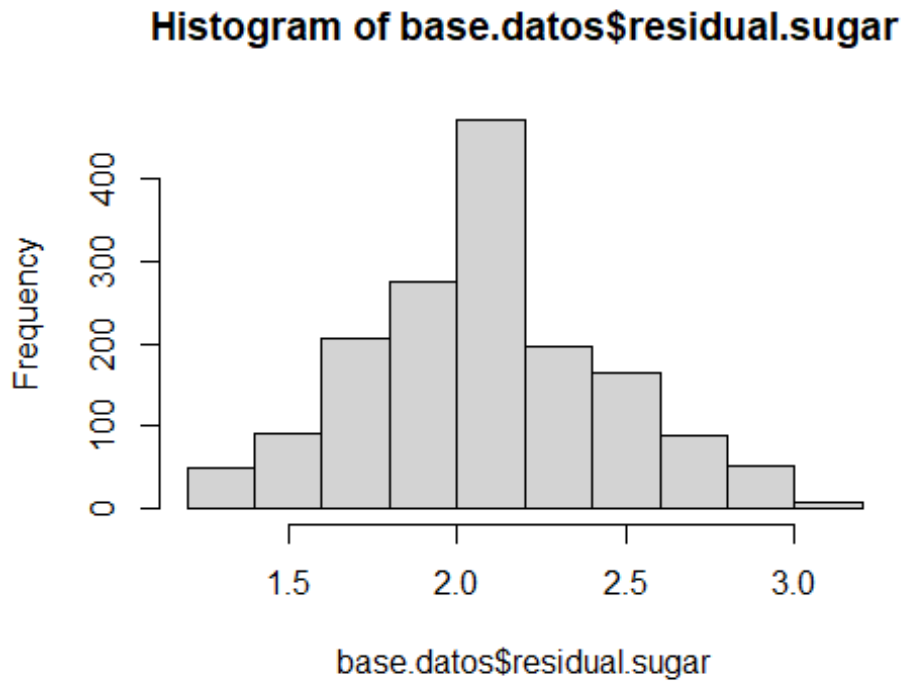

```
hist(base.datos$volatile.acidity)
```



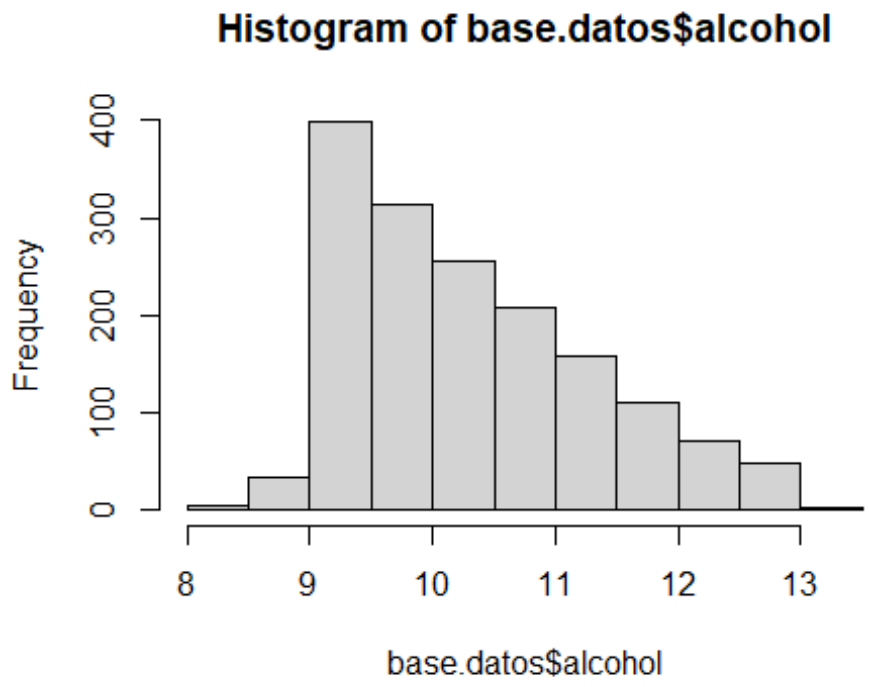

```
hist(base.datos$citric.acid)
```



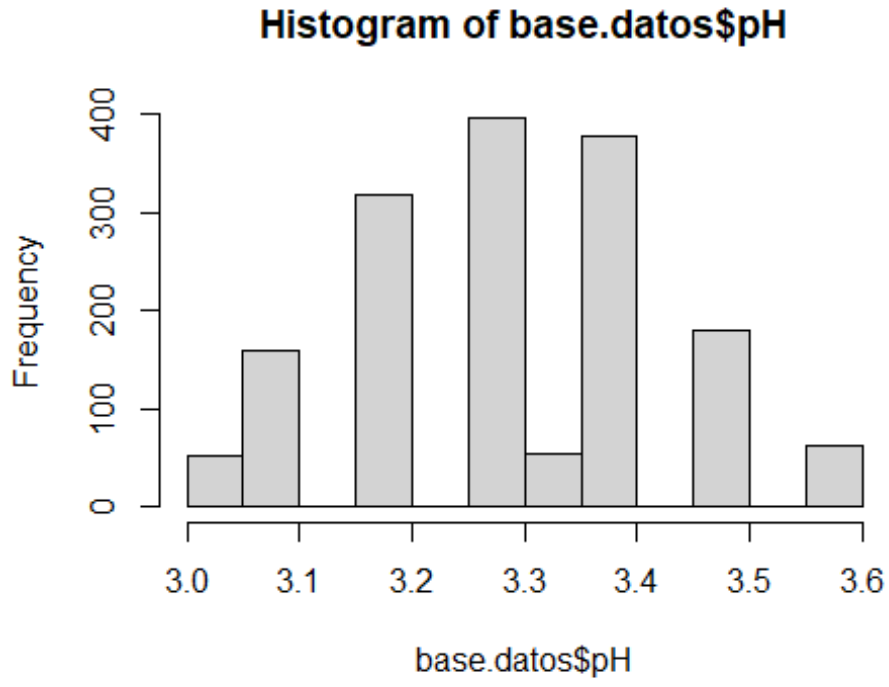

```
hist(base.datos$residual.sugar)
```



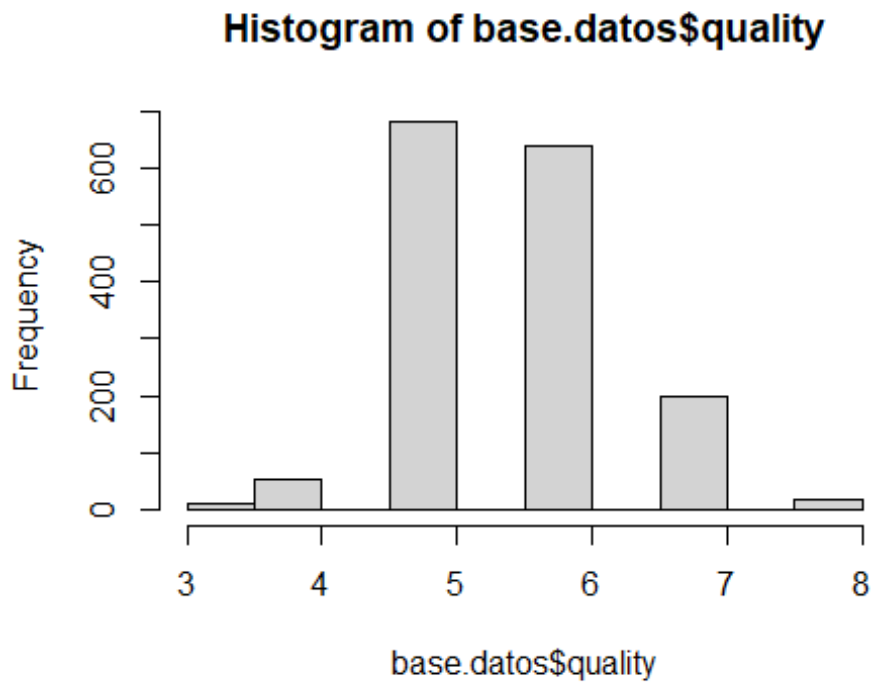

```
hist(base.datos$alcohol)
```




```
hist(base.datos$pH)
```




```
hist(base.datos$quality)
```




```
# Representación de la variable calidad_BM dicotómica por niveles:
table(base.datos$calidad_BM)

##
##    0    1
##   63 1536
```

A lo largo del análisis también se ha ido mostrando allí donde son necesarios otros plots/gráficos y documentación tanto visual como numérica de las diferentes variables que conforman la totalidad de nuestro dataset, por lo que realmente, la representación visual de la totalidad del Dataset no puede limitarse únicamente a dicho apartado, siendo este únicamente a modo resumen después del análisis llevado a cabo y las consideraciones pertinentes.

6. RESOLUCIÓN DEL PROBLEMA. CONCLUSIONES

En línea con lo que se ha ido considerando y verificando a lo largo de la totalidad del documento y bajo las consideraciones inicialmente planteadas en torno a la calidad del vino según fuentes vitivinícolas fiables y de interés, presentamos la siguiente conclusión:

En diversas fuentes, encontramos como la acidez en un vino se puede asociar con la calidad de este. “Un vino debe su calidad en gran parte a su acidez”. A lo largo de toda la actividad, se ha trabajado en este concepto en base a las variables de las que disponíamos en nuestro conjunto de datos que podían influir en el problema de investigación.

Concretamente, se escogió un subconjunto del mismo dataset para llevar a cabo la verificación y el análisis estadístico, no solo formado por las variables descriptoras de la acidez (fixed.acidity, volatile.acidity y citric.acid) sinó también por las variables pH, residual.sugar estos últimos parámetros se especulaba a través de considerables fuentes que también influyen en la calidad de un vino.

Una vez dispusimos de dichas variables, las tratamos en términos de Datos para poder llevar a cabo 4 pruebas estadísticas básicas para poder referirnos a nuestro problema inicial de investigación.

Primeramente, evaluamos la correlación total entre la totalidad de las variables de nuestro subconjunto, donde pudimos ver el grado de asociación de estas, obteniendo pares de variables emparejadas y preparando el análisis para la imposición de un modelo de regresión lineal capaz de explicar la calidad del vino.

El modelo lineal resultó con apenas una explicación de la calidad de un 27 % según el ajuste del modelo, lo que corresponde un valor bajo en términos de bondad de ajuste para un modelo pudiendo concluir que a priori, no podemos explicar calidad “quality” con la significancia moderada y introducida ni por las variables ácidas, ni por las demás añadidas posteriormente (según las consideraciones alcohol-acidez y azúcar-acidez que se comentaron inicialmente en la actividad y que podrían influir en la calidad de los vinos), en un modelo lineal múltiple.

En este punto, nos dispusimos a dicotomizar la variable quality en un nivel binario mediante la instauración de la variable calidad_BM. En este punto, pudimos llevar a cabo un modelo logístico, que tratase de indicarles la probabilidad, en este caso, de obtener un vino de calidad (valor 1 de calidad_BM) en función de una serie de predictores, en línea a los que consideramos para formar el modelo lineal…

Tras la evaluación logística y la bondad de ajuste de modelo en términos de significancia, encontramos un modelo donde las variables fixed.acidity, citric.acid, residual.sugar y pH nos proporcionaban una correcta aproximación de la calidad de un vino. En este punto, nos acercamos más a la consideración de que la calidad en un vino venía asociada, en buena parte, a la calidad de este según las variable de acidez presentes en el modelo…

En este punto y focalizados en rebatir y/o ratificar las consideraciones iniciales tomadas desde diversas fuentes vitivinícolas en relación a la relación acidez-calidad de un vino y el resultado de la predicción logística, nos hemos dispuesto a evaluar, en una última instancia, un contraste de hipótesis sobre muestras referentes a alta calidad y baja calidad donde hemos visto y refutar hipótesis en relación a las 3 variables de acidez que disponíamos en nuestro dataset. Como resultado, hemos podido ver, en línea con la bondad del modelo logístico, como las variables de acidez fixed.acidity y citric.acid (esta última contenida dentro del mismo valor de acidez NO volátil del vino y por ende de fixed.acidity) se asocian con la interpretación de la calidad del vino, pudiendo influir significativamente en esta y ratificando la idea general presentada en la introducción de la actividad cuando se expuso acerca de la calidad del vino.

Por otro lado, no podemos concluir con la misma idea para el nivel de acidez relativo a la acidez volátil de un vino (ácidos lácticos, acéticos, etc…), donde no hemos obtenido suficiente significancia para relacionar su justificación en la calidad de un vino, a priori, con los datos de los que hemos dispuesto de nuestro Dataset.

7. GUARDADO DEL FICHERO LIMPIO (CLEAN)

En este punto y por último, nos disponemos a guardar nuestro fichero base.datos debidamente modificado a lo largo de la actividad:

```
# Defino la sentencia de guardado del documento en el directorio actual:
write.csv(base.datos, file = './WINE_CLEAN.csv')
```