Práctica 2: Limpieza y análisis de datos

Andoni Zengotitabengoa Fernandez & Lucas Farris

23 de mayo de 2020

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

Hemos elegido el dataset 'Wine Quality Data Set´ de la fuente: https://archive.ics.uci.edu/ml/datasets/wine+quality) que contiene datos de dos clases de vinos portugueses: vino verde (*white wine*) y vino tinto (*red wine*). Los datos incluyen variables fisicoquímicas (acidez, pH, densidad, volatilidad ácida, contenido en alcohol, calidad, etc.), y una variable objetivo sensorial, que representa la calidad del vino.

A la hora de elegir la base de datos hemos tenido en cuenta que es apropiada para responder al ejercicio es decir proceder a la limpieza de datos y realizar tres tipos de análisis de datos. En cuanto a la limpieza de datos el único inconveniente es la inexistencia de valores nulos, pero podemos aplicar otras técnicas como reducción de la dimensionalidad, análisis de valores extremos y escalar las variables.

En cuanto al análisis de datos pensamos que este es un *dataset* en el que podemos utilizar métodos supervisados, no supervisados y técnicas de regresión y/o estadísticas como contraste de hipótesis o análisis de la varianza. Así podemos unir las bases de datos para un tipo de análisis (clasificación, análisis de la varianza) y utilizarlas por separado para otro tipo (regresión).

1. **Problemas no supervisados:** Es decir cuando a partir de variables dependientes obtenemos información sobre la variable independiente (la cual disponemos para comprobaciones pero no para crear el modelo la cual nos da un plus de maniobrabilidad), ademas los atributos son apropiados para su uso con el algoritmo *k-means*. En este sentido el hecho de agrupar dos tipos de vinos con los mismos atributos nos permite realizar una agrupación sencilla sin perdernos en los detalles. Vamos a unir las dos bases de datos y vamos a intentar agrupar de forma que al vino blanco se le asigne un grupo y al vino tinto otro grupo utilizando el algoritmo *k_means*.
2. **Regresión lineal:** A partir de un conjunto de variables dependientes cercanas a una distribución normal, creamos un modelo que aproxima la curva de la variable independiente. En este modelo la hipotesis nula nos indica si hay una fuerte relación entre las variables fisicoquimicas y la calidad, y nos ayuda a entender como cada variable afecta la calidad.
3. **Contraste de hipotesis:** Como tenemos los datos de dos tipos de vinos (tinto y blanco) es interesante explorar si hay diferencia entre los valores de la variable sensorial (calidad) de cada tipo de vino. Para el test miraremos si las distribuciones son normales, y si podemos usar un test paramétrico. Si no usaremos un test no paramétrico para comparar la calidad de los tipos de vino.

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

En la primera parte cargamos los datos de ficheros CSV, eliminamos registros duplicados, y juntamos los diferentes conjuntos de datos.

```
set.seed(42)
# importamos los datos de los csv descargados
red_wine_data <- read.csv('winequality-red.csv', sep = ";", quote = "\"")
white_wine_data <- read.csv('winequality-white.csv', sep = ";", quote = "\"")
# añadimos el tipo de vino como una nueva variable categórica
```

```r
red_wine_data$type <- "red"
white_wine_data$type <- "white"
# comprobaremos cuantos registros duplicados hay para cada dataset
sum(duplicated(red_wine_data))
```

```
## [1] 240
```

```r
sum(duplicated(white_wine_data))
```

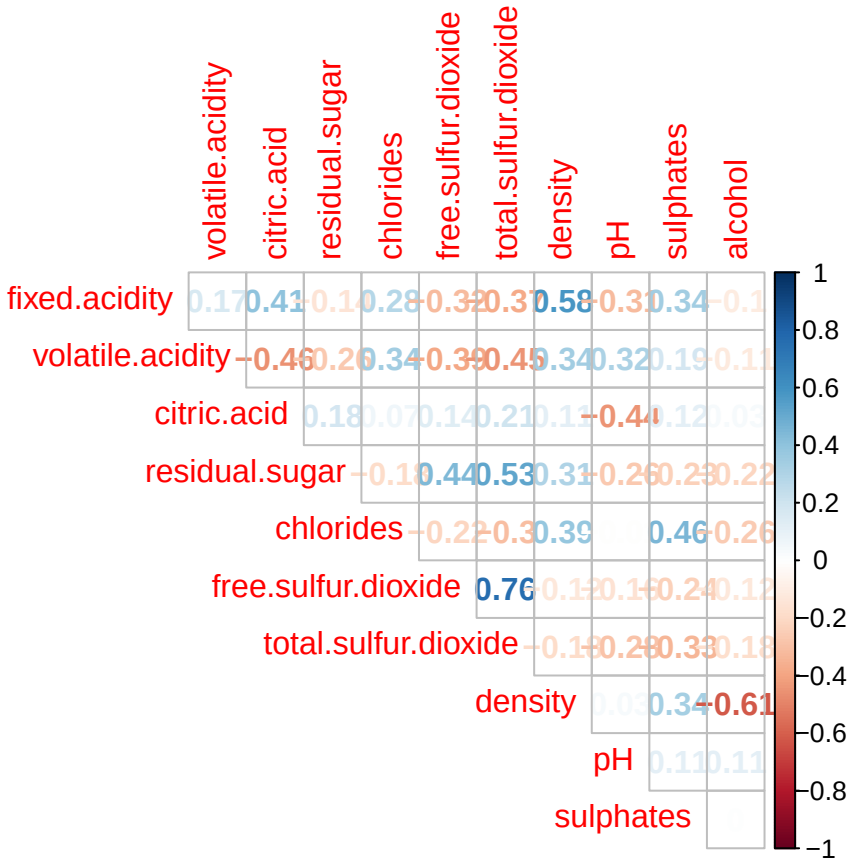
```
## [1] 937
```

```r
# eliminaremos los registros duplicados
red_wine_data <- red_wine_data[!duplicated(red_wine_data), ]
white_wine_data <- white_wine_data[!duplicated(white_wine_data), ]
# juntamos los datos seleccionando el mismo número de muestras para cada tipo de vino
white_wine_data <- white_wine_data[sample(nrow(x = white_wine_data),
                                          nrow(x = red_wine_data)),]
dataset <- rbind(red_wine_data, white_wine_data)
dataset$type <- as.factor(dataset$type)
# cantidad de registros disponibles para cada dataset
nrow(dataset)
```

```
## [1] 2718
```

Ahora hacemos el analisis de correlaciones de las variables.

```r
# miraremos ahora si hay variables numericas en los datos que tienen alta
# correlación para el data set conjunto y con las clases de vinos separadas
corrplot(
  cor(dataset[, 0:11], method = 'pearson'),
  type = "upper",
  method = 'number',
  diag = FALSE
)
```



La correlación más alta (0.76) fue entre las variables *free.sulfur.dioxide* y *total.sulfur.dioxide* pero no es suficientemente alta para retirar una de las variables con confianza. La correlación más baja (-0.6) fue entre las variables *density* y *alcohol* pero tampoco es suficientemente baja para eliminar una de las variables.

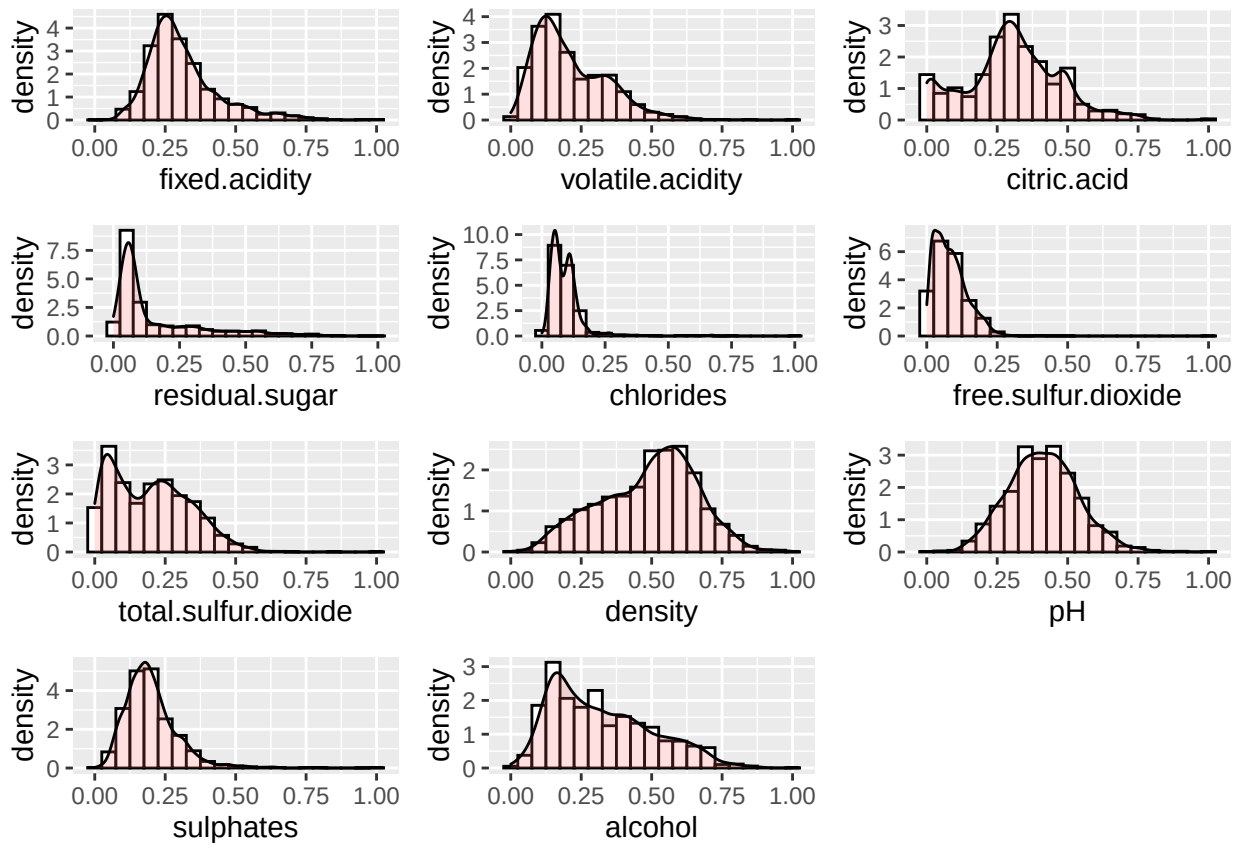
Para realizar algunos de los análisis (por ejemplo agrupación) devemos transformar las variables en una escala común (ver http://rpubs.com/ydmarinb/429761). Vamos a utilizar la función `rescale` que transforma el dominio de las variables en el intervalo $[0, 1]$ utilizando la siguiente aplicación: $x.escalada = \frac{x - min(x)}{max(x) - min(x)}$ Entonces:

```
dataset_scaled <- dataset
# para cada variable numerica
for (col in colnames(dataset)[0:11]) {
  # aplicamos la transformacion de la escala
  dataset_scaled[, col] <- rescale(dataset[, col])
}
```

Vamos a visualizar las variables:

```
grobs <- vector('list', 11)
i <- 1
# para cada variable numerica
for (col in colnames(dataset_scaled[0:11])) {
  # creamos su histograma y density plot
  grobs[[i]] <- ggplot(dataset_scaled, aes_string(x = col)) +
    geom_histogram(aes(y = ..density..), binwidth = .05,
                   colour = "black", fill = "white") +
    geom_density(alpha = .2, fill = "#FF6666")
  i <- i + 1
```

```
}
grid.arrange(grobs=grobs, nrow = 4, ncol=3)
```



3. Limpieza de los datos.

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

```
# comprobaremos si algun valor de nuestro dataset es vacío
any(is.na(dataset_scaled))
```

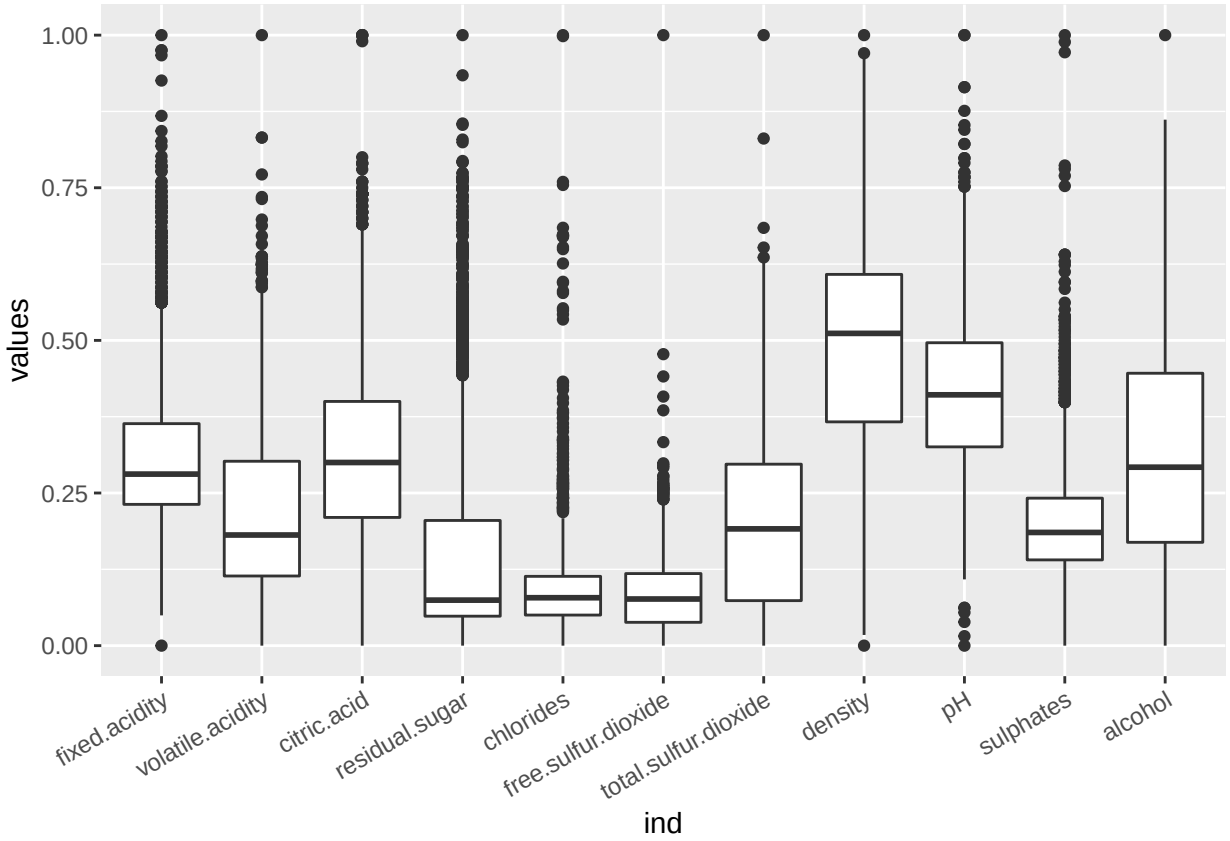
```
## [1] FALSE
```

En nuestro dataset no tenemos ningún caso de valores vacíos. Tenemos casos de valores ceros en la variable *citric.acid*, pero era esperado que algunos vinos no tendrían niguna cantidad de ácido cítrico una vez que es un corrector de acidez que se utiliza cuando es necesario. Si tuvieramos valores vacíos en nuestras variables numéricas, los podríamos reemplazar (por la media de la variable por ejemplo), o predecir con valores que tengan la máxima probabilidad de ser correctos (por ejemplo la tecnica *miss forest*).

3.2. Identificación y tratamiento de valores extremos.

Para examinar visualmente los valores extremos de las variables numéricas crearemos boxplots de cada una escalada para visualizar mejor las diferencias.

```
ggplot(stack(dataset_scaled[, 0:11]), aes(x = ind, y = values)) +
  geom_boxplot() + theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



El análisis de los valores extremos no es para nada trivial. Vemos que casi todos los atributos tienen valores extremos, pero no podemos retirarlos sin mas. Tenemos libertad para realizar tres analisis diferentes por tanto el tratamiento de los valores extremos será diferente dependiendo del analisis, a decir verdad el análisis de los valores extremos será determinante a la hora de escoger los análisis a realizar. Lo que realmente vamos a realizar es adaptarnos a la existencia de valores extremos y porque:

```
# funcion para eliminar registros usando los outliers de los plots
outlier_detection <- function(data) {
  lowerq <- quantile(data, na.rm = TRUE)[2]
  upperq <- quantile(data, na.rm = TRUE)[4]
  iqr <- upperq - lowerq
  threshold_upper <- (iqr * 1.5) + upperq
  threshold_lower <- lowerq - (iqr * 1.5)
  data > threshold_upper | data < threshold_lower
}
clean_dataset <-
  dataset_scaled[rowSums(sapply(dataset_scaled[, 0:11], outlier_detection),
                         na.rm = TRUE) == 0,]
# porcentaje de registros que hemos eliminado
(nrow(dataset_scaled) - nrow(clean_dataset)) / nrow(dataset_scaled)
```

```
## [1] 0.2251656
```

Porque vemos que si queremos eliminar los valores extremos nos tenemos que deshacer de 22 % de las muestras, en el caso de limpiarlos por separado la perdida sería menor pero continuaría siendo excesiva.

Esta gran cantidad de outliers no se puede atribuir a errores de medición y eliminarlos. Por ejemplo los valores de pH(acidez) ($pH = -log(H+)$) se encuentran entre 2 y 4 lo que es absolutamente normal. Sería de

extrañar un valor de 10-14 muy básico o de 0 muy ácido, ya 20 estaría fuera del rango de la escala de pH (0-14).

Analizando los boxplots no parece que sean tantos pero estos se acumulan atributo por atributo desintegrando el dataset. En este caso pensamos que el valor extremo en un atributo implica un handicap para el vino pero no determina de por si la calidad del vino, buena o mala (en ese caso un modelo de mineria de datos sería trivial) una vez que un porcentaje muy grande de vinos seleccionados tienen algún valor extremo en alguno de sus atributos.

Los valores extremos en densidad o contenido de alcohol podrían diferenciar un vino por la positiva para algunos paladares pero en estas dos variables existen muy pocos valores extremos. En estos dos casos conviene dejarlos simplemente por la cantidad de datos que existen y los tendremos en cuenta.

Estos son los tres análisis y el tratamiento de los valores extremos, teniendo en cuenta las limitaciones de tiempo y el encuadramiento de ejercicio:

1. Agrupación (*k-means*) utilizando `dataset_scaled` de los vinos blancos y tintos donde no es necesario ni tendría sentido retirar tantas entradas con valores extremos.
2. Regresión lineal para la calidad del vino (blanco y tinto) utilizando `dataset_scaled` donde no es necesaria la suposición de normalidad de las distribuciones por lo que en este caso no retiramos los valores extremos.
3. Test de hipótesis: utilizando la variable calidad dividiéndola en dos grupos (uno para cada tipo), no excluiremos datos de valores extremos para la análisis.

4. Análisis de los datos.

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

Estos son los tres análisis y el conjunto de datos necesario para cada uno:

1. Agrupación (*k-means*): utilizando el conjunto de las variables dependientes, y mediremos la efectividad del agrupamiento usando la variable `type`. Usaremos una técnica de balanceamento *under sampling* para garantizar que habrá la misma cantidad de vinos para cada tipo.
2. Regresión lineal: usaremos la variable objetivo y un subconjunto de las variables fisicoquímicas, que más se acerquen más a la normalidad, del *dataset* normalizado y balanceado.
3. Test de hipótesis: utilizaremos la variable *quality* y el tipo de vino, para el conjunto de datos normalizado y balanceado.

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

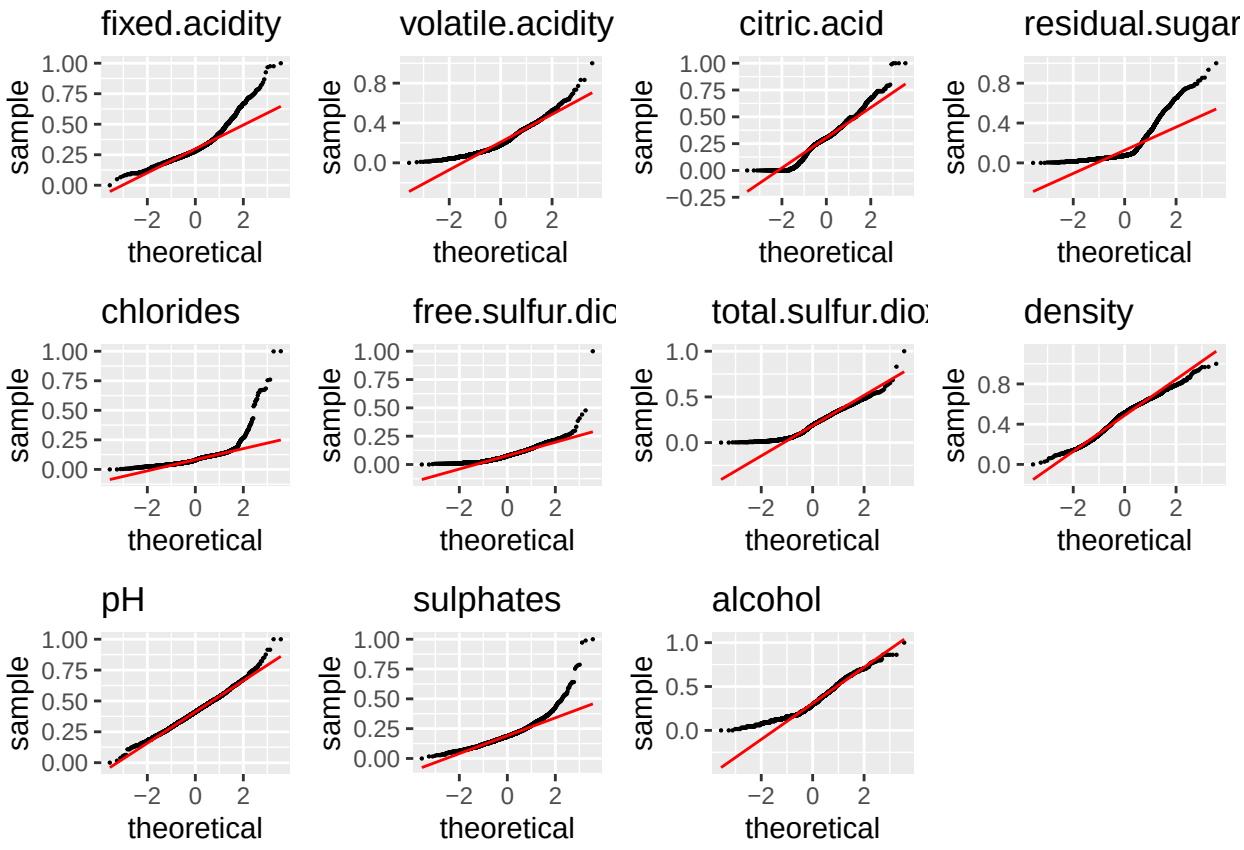
```
# usamos el test Shapiro-Wilk
for (col in colnames(dataset_scaled)[1:11]) {
  test_data <- dataset_scaled[,col]
  pvalue <- shapiro.test(test_data)$p
  if (pvalue < 0.05) {
    print(paste('La variable', col, 'no es normal, porque tiene valor p', pvalue))
  } else {
    print(paste('La variable', col, 'es normal, porque tiene valor p',
                round(pvalue, digits = 2)))
  }
}
```

```
## [1] "La variable fixed.acidity no es normal, porque tiene valor p 3.40115685746532e-38"
## [1] "La variable volatile.acidity no es normal, porque tiene valor p 3.46064913340437e-34"
```

```
## [1] "La variable citric.acid no es normal, porque tiene valor p 1.63898109926571e-19"
## [1] "La variable residual.sugar no es normal, porque tiene valor p 2.02002297158707e-54"
## [1] "La variable chlorides no es normal, porque tiene valor p 2.88909978997677e-60"
## [1] "La variable free.sulfur.dioxide no es normal, porque tiene valor p 1.70785798324673e-41"
## [1] "La variable total.sulfur.dioxide no es normal, porque tiene valor p 1.67650925422898e-29"
## [1] "La variable density no es normal, porque tiene valor p 8.97335804462957e-17"
## [1] "La variable pH no es normal, porque tiene valor p 1.62361293865772e-08"
## [1] "La variable sulphates no es normal, porque tiene valor p 5.42716344124033e-40"
## [1] "La variable alcohol no es normal, porque tiene valor p 5.43712753125272e-30"
```

También podemos comprobar visualmente la normalidad con Q-Q plots.

```
grobs <- vector('list', 11)
i <- 1
# para cada variable numerica
for (col in colnames(dataset_scaled[1:11])) {
  # creamos su histograma y density plot
  grobs[[i]] <-
    ggplot(dataset_scaled, aes_string(sample = col)) + stat_qq(size = 0.1) +
    stat_qq_line(color = 'red') + ggtitle(col)
  i <- i + 1
}
grid.arrange(grobs = grobs, nrow = 3, ncol = 4)
```



Según el test ninguna de las variables es normal. Las que tienen la distribución más cercana son *pH* y *density*. Como ninguna de las variables es normal, para comprobar la homocedasticidad usaremos el test de Fligner-Killeen. Comprobaremos la homogeneidad de la varianza entre los dos tipos de vino, para cada

variable.

```
for (col in colnames(dataset_scaled)[1:11]) {
  pvalue <- fligner.test(as.numeric(dataset_scaled$type), dataset_scaled[,col])$p.value
  if (pvalue < 0.05) {
    print(paste('La variable',col,
                'no presenta homocedasticidad entre los tipos de vino'))
  } else {
    print(paste('La variable', col,
                'presenta homocedasticidad entre los tipos de vino'))
  }
}
```

```
## [1] "La variable fixed.acidity no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable volatile.acidity no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable citric.acid no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable residual.sugar no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable chlorides no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable free.sulfur.dioxide no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable total.sulfur.dioxide no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable density presenta homocedasticidad entre los tipos de vino"
## [1] "La variable pH no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable sulphates no presenta homocedasticidad entre los tipos de vino"
## [1] "La variable alcohol presenta homocedasticidad entre los tipos de vino"
```

Según nuestro test las variables que presentan homogeneidad de la varianza según el tipo de vino son *density* y *alcohol*.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Método no supervisado para separación por grupos

Clustering (*k-means*):

```
# exploramos los cuartiles de las variables, divididas por tipo de vino
summary(dataset_scaled[dataset_scaled$type == 'red', 1:11])
```

```
##  fixed.acidity     volatile.acidity   citric.acid     residual.sugar
##  Min.   :0.06612   Min.   :0.02013   Min.   :0.0000   Min.   :0.008772
##  1st Qu.:0.27273   1st Qu.:0.20134   1st Qu.:0.0900   1st Qu.:0.052632
##  Median :0.33884   Median :0.28859   Median :0.2600   Median :0.065790
##  Mean   :0.37278   Mean   :0.29495   Mean   :0.2723   Mean   :0.079974
##  3rd Qu.:0.44628   3rd Qu.:0.36913   3rd Qu.:0.4300   3rd Qu.:0.083333
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :0.649123
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.00000      Min.   :0.1775
##  1st Qu.:0.09683   1st Qu.:0.02083     1st Qu.:0.03687      1st Qu.:0.5115
##  Median :0.11185   Median :0.04514     Median :0.07373      Median :0.5779
##  Mean   :0.12708   Mean   :0.05171     Mean   :0.09407      Mean   :0.5784
##  3rd Qu.:0.13189   3rd Qu.:0.06944     3rd Qu.:0.13134      3rd Qu.:0.6455
##  Max.   :1.00000   Max.   :0.24653     Max.   :0.65207      Max.   :1.0000
##        pH           sulphates         alcohol
##  Min.   :0.0155   Min.   :0.0618   Min.   :0.0000
##  1st Qu.:0.3798   1st Qu.:0.1854   1st Qu.:0.1692
##  Median :0.4574   Median :0.2247   Median :0.2769
```

```
##  Mean   :0.4572   Mean   :0.2465   Mean   :0.3127
##  3rd Qu.:0.5271   3rd Qu.:0.2865   3rd Qu.:0.4154
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
summary(dataset_scaled[dataset_scaled$type == 'white', 1:11])
```

```
##  fixed.acidity    volatile.acidity   citric.acid     residual.sugar
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.2066   1st Qu.:0.08725   1st Qu.:0.2700   1st Qu.:0.03947
##  Median :0.2479   Median :0.12081   Median :0.3200   Median :0.17544
##  Mean   :0.2499   Mean   :0.13054   Mean   :0.3357   Mean   :0.22928
##  3rd Qu.:0.2893   3rd Qu.:0.16107   3rd Qu.:0.3900   3rd Qu.:0.36404
##  Max.   :0.6612   Max.   :0.55034   Max.   :1.0000   Max.   :1.00000
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00000   Min.   :0.006944    Min.   :0.03456      Min.   :0.0000
##  1st Qu.:0.03840   1st Qu.:0.076389    1st Qu.:0.22811      1st Qu.:0.2687
##  Median :0.05008   Median :0.111111    Median :0.29032      Median :0.3786
##  Mean   :0.05742   Mean   :0.117532    Mean   :0.30117      Mean   :0.4004
##  3rd Qu.:0.06344   3rd Qu.:0.152778    3rd Qu.:0.36751      3rd Qu.:0.5160
##  Max.   :0.43239   Max.   :1.000000    Max.   :1.00000      Max.   :0.9227
##        pH           sulphates         alcohol
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2868   1st Qu.:0.1067   1st Qu.:0.1692
##  Median :0.3643   Median :0.1461   Median :0.3077
##  Mean   :0.3695   Mean   :0.1523   Mean   :0.3364
##  3rd Qu.:0.4419   3rd Qu.:0.1854   3rd Qu.:0.4769
##  Max.   :0.8527   Max.   :0.4831   Max.   :0.8615
```

Vemos que existen diferencias significativas en casi todos los atributos entre los dos tipos de vinos por lo que deveriamos lograr agruparlos en diferentes grupos lo que se puede comprobar visualmente en los boxplots reescalados

Antes de aplicar el algoritmo k_means Vamos a aplicar el método hopkins que nos da la probabilidad de un conjunto de datos tener estructura interna o no (estructura aleatoria)

```
# Estadístico H para el set de datos
# Modificar el valor de n dependiendo del número escogido para m
hopkins(data = dataset_scaled[, 1:11], n = 30)$H
```

```
## [1] 0.1103088
```

Con el valor de H intuimos que existe algun tipo de estructura.

```
set.seed(42)
# Utilizamos tecnicas de reducción del tamaño (selección aleatoria)
samples <- sample(nrow(dataset_scaled), 200)
X <- dataset_scaled[samples, 1:11]
# Vamos aplicar el algoritmo _k-means_ para 2 clusters
y <- kmeans(X, 2)$cluster
```

```
# Para visualizar los clústers podemos usar la funcion clusplot. Vemos la agrupación con 2 clústers
png(file="clusplot_k2.png",width=600, height=350)
clusplot(X, y, color=TRUE, shade=TRUE, labels=1, lines=0)
dev.off()
```

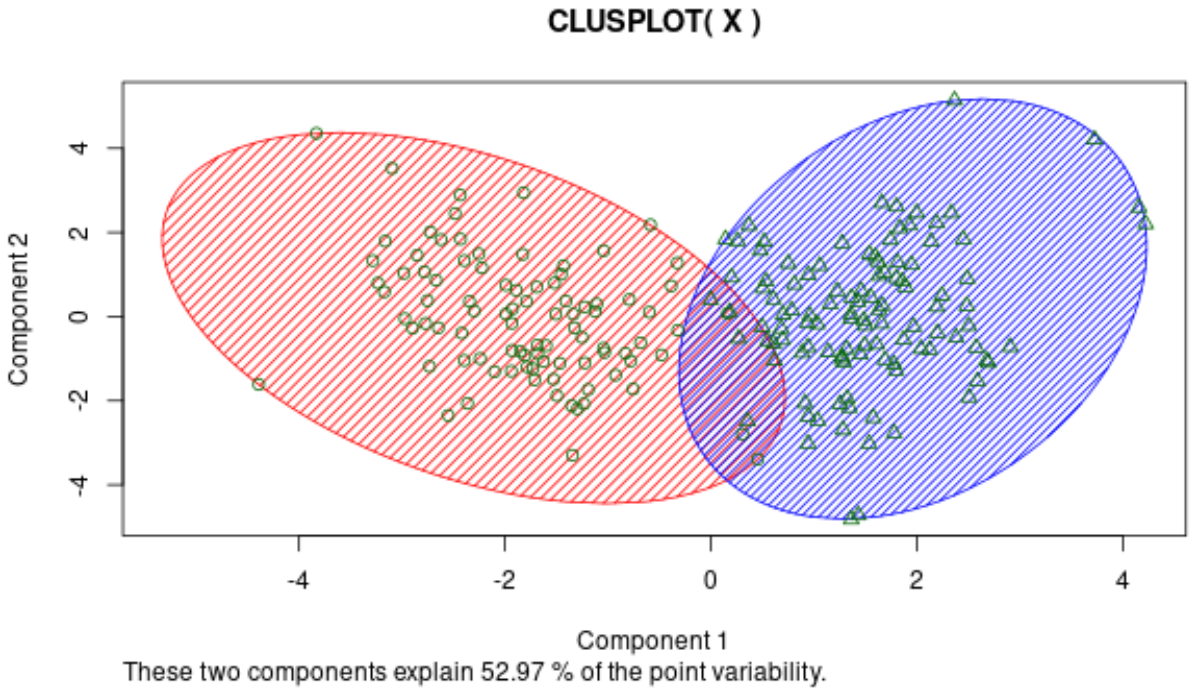


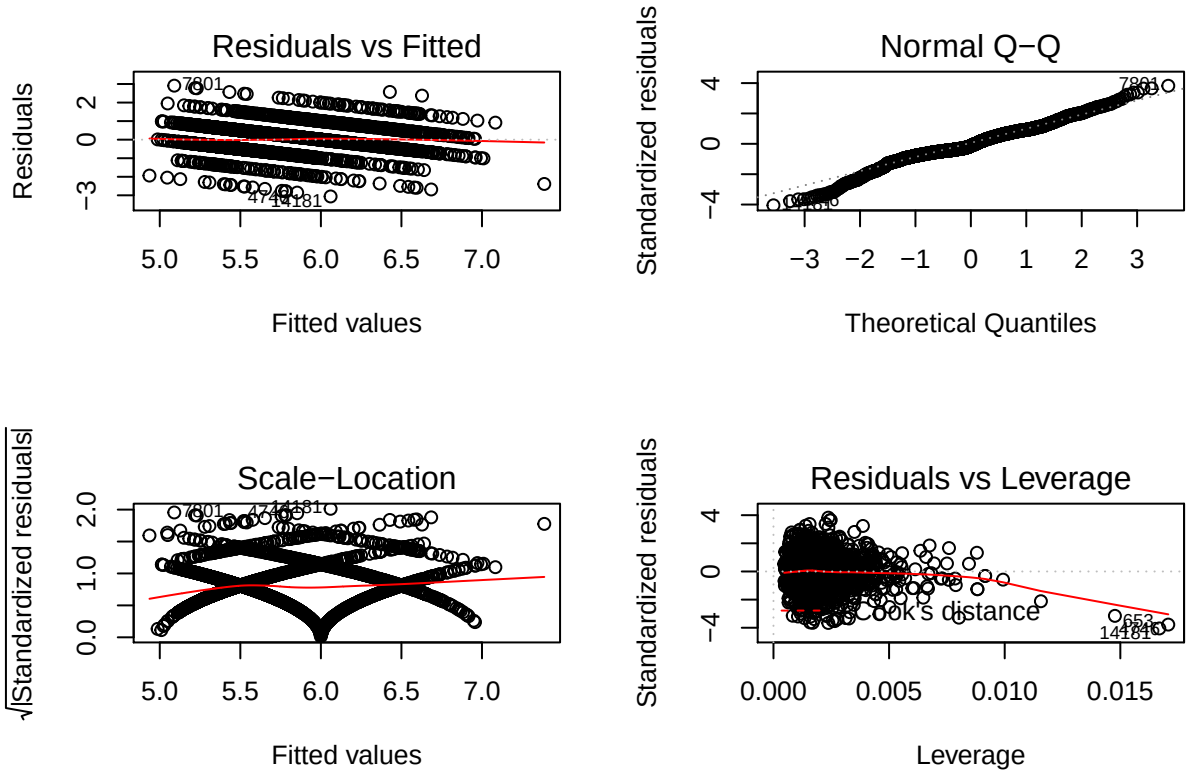
Figura 1: clusplot

Como se puede comprobar aunque las variables solo explican alrededor 52 % de la variabilidad con dos clusteres vemos que existen dos grupos diferenciados.

Regresión Lineal

Para la analisis usaremos las variables más proximas de la normalidad: *density*, *pH*, *total.sulfur.dioxide* y *alcohol*.

```
linear_reg <- lm(quality ~ density + pH + total.sulfur.dioxide + alcohol,
                 data=dataset_scaled)
par(mfrow = c(2, 2))
plot(linear_reg)
```



Podemos comprobar que los residuos son normales, eso nos indica que la regresión es significativa.

Contraste de hipotesis

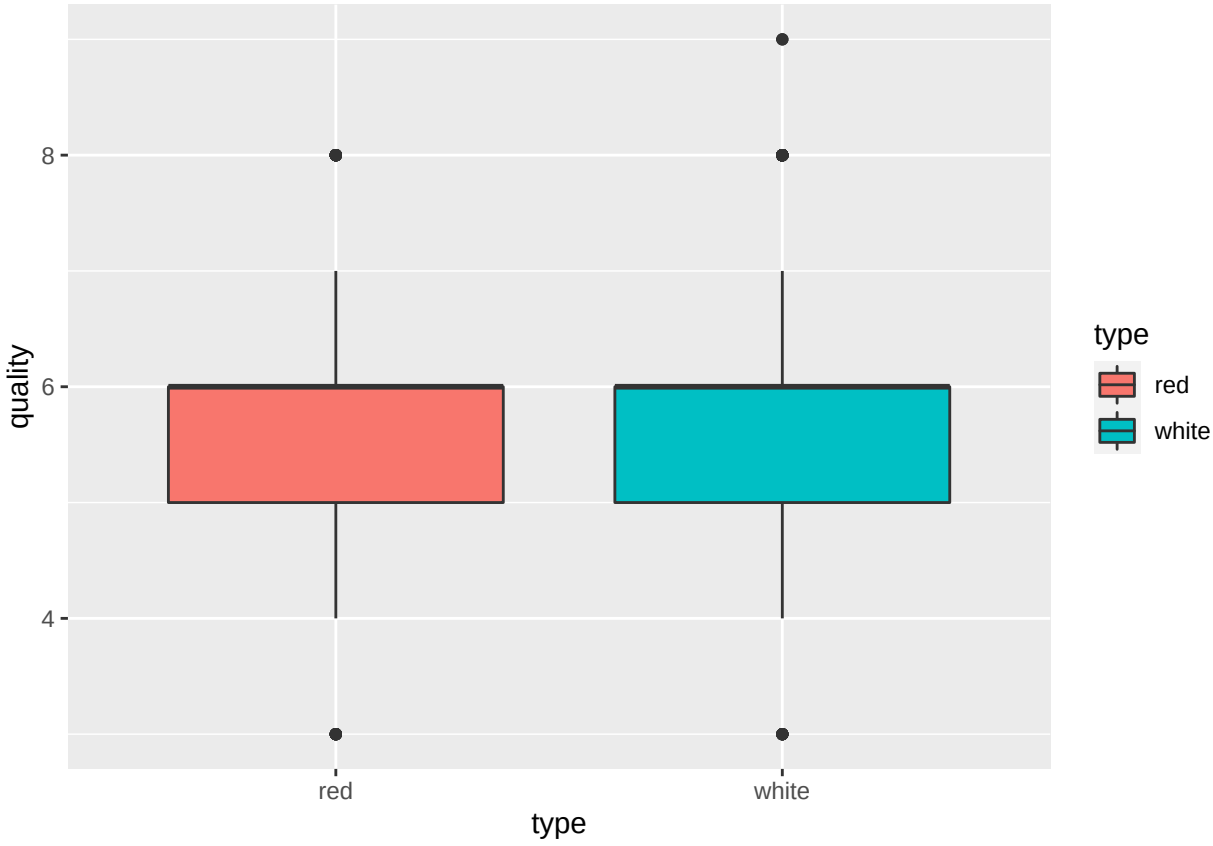
Ahora miraremos si hay diferencia en la calidad de los vinos tintos o blancos. Comprobaremos la normalidad de la variable *quality*.

```
shapiro.test(dataset_scaled$quality)$p
```

```
## [1] 2.920613e-41
```

Como el p-valor es pequeño, rechazamos la hipotesis que la variable es normal. Usaremos entonces el test no paramétrico de Mann–Whitney. Comprobaremos ahora visualmente la distribución de los dos grupos con un *box plot*:

```
quality_red <- dataset_scaled[dataset_scaled$type=='red','quality']
quality_white <- dataset_scaled[dataset_scaled$type=='white','quality']
ggplot(dataset_scaled, aes(x=type, y=quality, fill=type)) +
    geom_boxplot()
```



Las dos distribuciones se parecen mucho. Haremos el contraste para comprobar si los vinos rojos tienen la calidad inferior a los blancos:

```
contrast <- wilcox.test(quality_red, quality_white,
                        paired = FALSE, alternative="greater")
contrast
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  quality_red and quality_white
## W = 783276, p-value = 1
## alternative hypothesis: true location shift is greater than 0
```

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

Vamos a comparar la clasificación obtenida con una muestra aleatoria los datos reales de la clasificación de los vinos.

```
confusionMatrix(dataset_scaled[samples,'type'], factor(y, c(1,2),c('white','red')))
```

```
## Warning in confusionMatrix.default(dataset_scaled[samples, "type"], factor(y, :
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

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

```
## Prediction white red
##      white    88   0
##      red       3 109
##
##                Accuracy : 0.985
##                  95% CI : (0.9568, 0.9969)
##     No Information Rate : 0.545
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9697
##
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 0.9670
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9732
##              Prevalence : 0.4550
##          Detection Rate : 0.4400
##    Detection Prevalence : 0.4400
##       Balanced Accuracy : 0.9835
##
##        'Positive' Class : white
##
```

Visualmente vemos por ejemplo que para y la agrupación {1} Tinto {2} Blanco se corresponde con un indice de confianza muy alto. Nota: como no sabemos que número nos va asignar el algoritmo a cada conjunto (1 o 2) entenderemos como error para un resultado bajo y como acierto para un porcentaje alto.

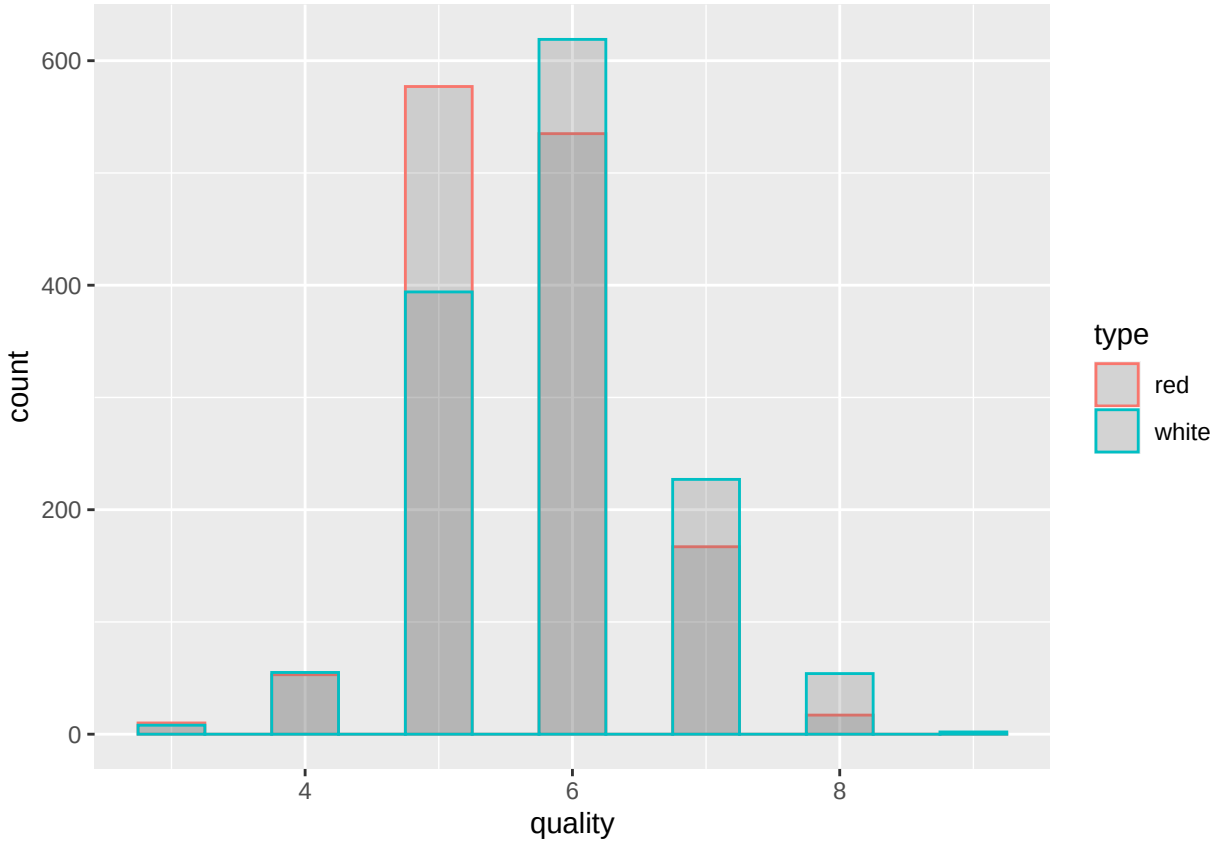
Ahora miraremos la significancia estadística de nuestra regresión.

```
summary(linear_reg)
```

```
##
## Call:
## lm(formula = quality ~ density + pH + total.sulfur.dioxide +
##     alcohol, data = dataset_scaled)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06026 -0.42942 -0.09613  0.52778  2.90997
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.1741     0.1109  46.648  < 2e-16 ***
## density               -0.1484     0.1169  -1.270  0.20417
## pH                    -0.3882     0.1218  -3.188  0.00145 **
## total.sulfur.dioxide   0.1880     0.1205   1.560  0.11885
## alcohol                2.3565     0.1116  21.113  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.762 on 2713 degrees of freedom
## Multiple R-squared:  0.2431, Adjusted R-squared:  0.242
## F-statistic: 217.9 on 4 and 2713 DF,  p-value: < 2.2e-16
```

En el contraste de hipotesis, el alto valor p ($p \simeq 1$) indica que no podemos rechazar la hipotesis nula H_0, entonces podemos decir que la calidad de los vinos tintos, percibida por los participantes, es de hecho inferior a la calidad de los vinos blancos. Podemos comprobar la diferencia con un histograma:

```
ggplot(dataset_scaled, aes(x=quality, color=type)) +
  geom_histogram(alpha=0.2, position="identity",binwidth=0.5)
```



Podemos comprobar visualmente que hay más vinos blancos con calidad 6, 7 y 8; y más vinos tintos con calidad 5.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Conclusiones: Los resultados han sido los esperados. Hemos explicado como funciona el algoritmo Hopkins, en este sentido es importante comprenderlo como un contraste de hipotesis, entre existe o no existe una posible estructura en los datos y utilizarlo antes de proceder a realizar agrupaciones también es importante entender que los grupos mejor definidos son aquellos que tienen una estructura aleatoria no uniforme, si las agrupaciones las realizamos en estructuras no aleatorias uniformes podemos obtener agrupaciones artificiales que no tengan sentido. No es el caso porque la asociación que hemos obtenido es real. Hemos seleccionado una base de datos con ciertos atributos uniéndola con otra con los mismos atributos, las hemos seleccionado y hemos escogido al azar una muestra. Es muy importante seleccionar al azar muestras de tamaño reducido en una base de datos con muchas entradas, porque obtenemos resultados que van mejorando según n aumenta (ley fuerte de los grandes números) y simplificamos el ejercicio ahorrando mucho tiempo de ejecución. Permitiendo mejores visualizaciones. Hemos preparado los datos para utilizar el algoritmo *k-means*, es muy importante normalizar con una escala apropiada los atributos para que el peso de algunos atributos no influya negativamente. Hemos visualizado los resultados para el valor de $k = 2$ y hemos comprobado que la agrupación es la que buscábamos. Para el calculo de la calidad del agrupamiento hemos comprobado la matriz de confusión y medidas de

precisión.

En la regresión lineal hemos elegido cuatro variables (densidad, pH, dióxido de azufre total y alcohol), y comparando los resultados hemos comprobado que las variables que más afectan la calidad fueron el nivel de alcohol y la acidez (pH). También hemos comprobado que la regresión fue significativa.

En el contraste de hipótesis hemos intentado descubrir si habría diferencia entre la calidad sensorial de los dos tipos de vino (tinto y blanco). Comprobamos que la calidad no tiene una distribución normal, y entonces usamos un test no paramétrico. Según el resultado de nuestro test, y la comprobación visual con un histograma, hemos comprobado que los vinos rojos en general tienen menos calidad que los blancos.