Analisis_No_Supervisado

Belen Paredes y Cesar Velasco

25/01/2021

INTRODUCCIÓN

En este análisis no supervisado se utiliza un dataset de las variantes tintas del vino portugués Vinho[1], donde se desea optener grupos optimos a partir de las caracteristicas fisicas y quimicas

Introducción teórica

El clustering es una tarea que tiene como finalidad principal lograr el agrupamiento de conjuntos de objetos no etiquetados, para lograr construir subconjuntos de datos conocidos como Clusters.

Métodos de clusterización de datos a utilizar:

K-means: también utiliza las distancias entre puntos. Y agrupo los clusters según lo cerca que están de los centros de gravedad de los clusters. A priori, tienes que saber cuántos clusters vas a crear para que el algoritmos sepa dónde colocar cada punto. Gaussian mixture models (GMM): utiliza modelos gaussianos o modelos de distribuciones normales de diferentes formas para que puedas crear grupos en forma de elipses. Para cada punto se le calcula la probabilidad de pertenencia a cada modelo gaussiano

GIT: https://github.com/CAVA1611/fid-analisis

```
chooseCRANmirror(graphics=FALSE, ind=1)
knitr::opts_chunk$set(echo = TRUE)

#Paquetes vistos en clases

install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x7l7820pg2h6vnmmm0000gn/T//RtmpPIaOUm/downloaded_packages
```

```
library(dplyr)
```

```
##
## 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
```

```
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
```

```
##   /var/folders/9c/qjx5f99x7l7820pg2h6vnmmm0000gn/T//RtmpPIaOUm/downloaded_packages
```

```
library(ggplot2)

#Paquetes  consultados

install.packages("cluster")
```

```
##
## The downloaded binary packages are in
##   /var/folders/9c/qjx5f99x7l7820pg2h6vnmmm0000gn/T//RtmpPIaOUm/downloaded_packages
```

```
library(cluster)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(mclust)
```

```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(FactoMineR)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
#lectura del dataset winequality-red.csv

wine <- read.csv("winequality-red.csv")
```

Atributos

-fixed.acidity:
-Volatile.acidity: -Citric.acid:
-Residual.sugar:
-Chorides -Free.sulfr: -Total.sulur:
-Density:
-Ph:
-Sulphates:
-Alcohol:
-Quality

Se realiza la correlacion entre columnas para verificar cuales nos ayudaran a determinar de mejor manera la calidad de vino

```
#Empezamos el algoritmo observando las variables con un matrixplot:

cor(wine$quality, wine$density)
```

```
## [1] -0.1749192
```

```r
cor(wine$quality, wine$alcohol)
```

```
## [1] 0.4761663
```

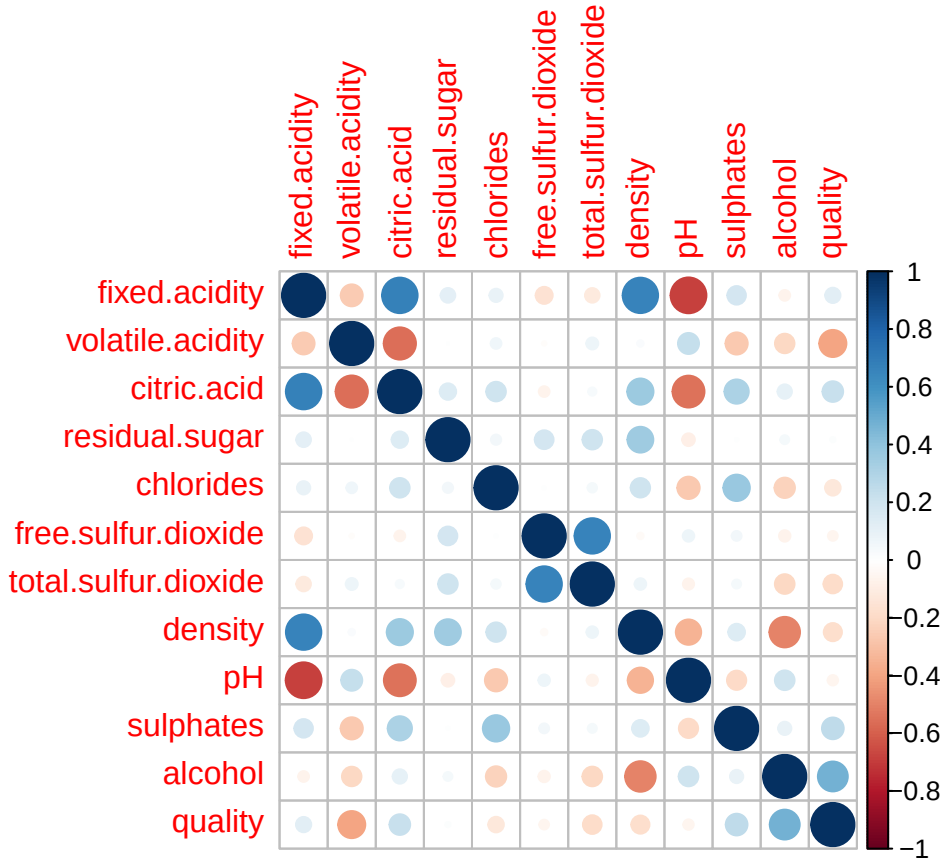
```r
cor(wine$quality, wine$pH)
```

```
## [1] -0.05773139
```

```r
cor(wine$quality, wine$sulphates)
```

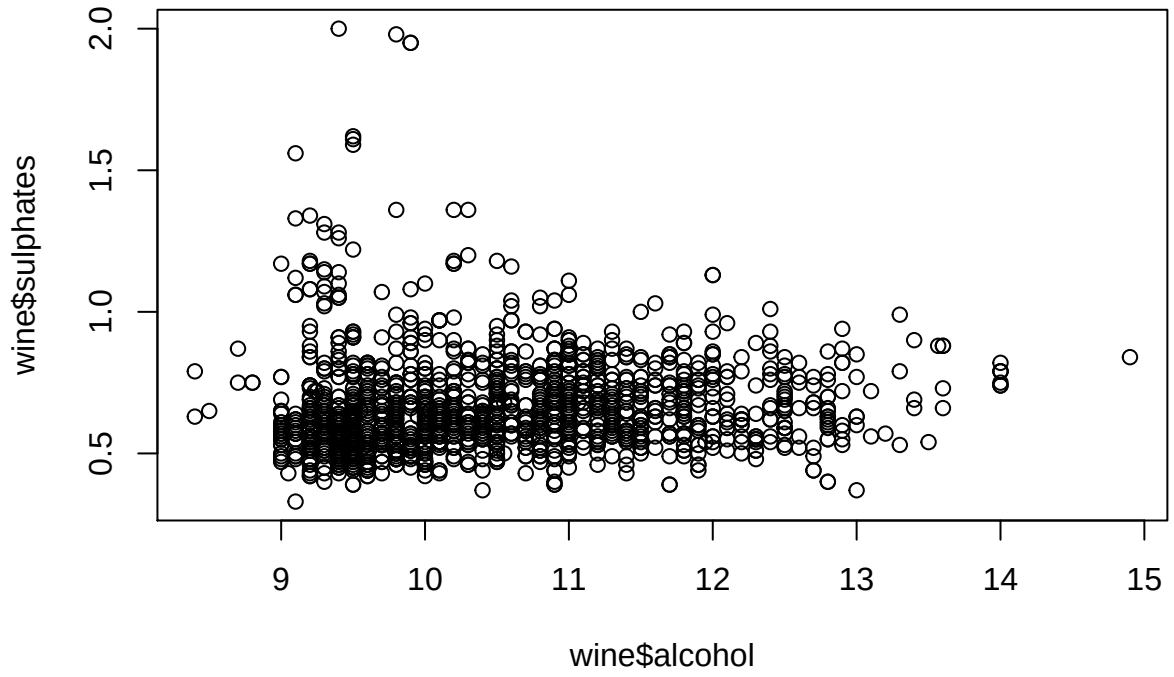
```
## [1] 0.2513971
```

```r
# grafica de correlacion
correlations <- cor(wine) # correlation matrix
corrplot(correlations, method="circle") # corrplot
```



Segun se observa en la grafica, exite una mayor correlacion para determinar la mayor calidad del vino con los datos de la columna alcohol y sulphates

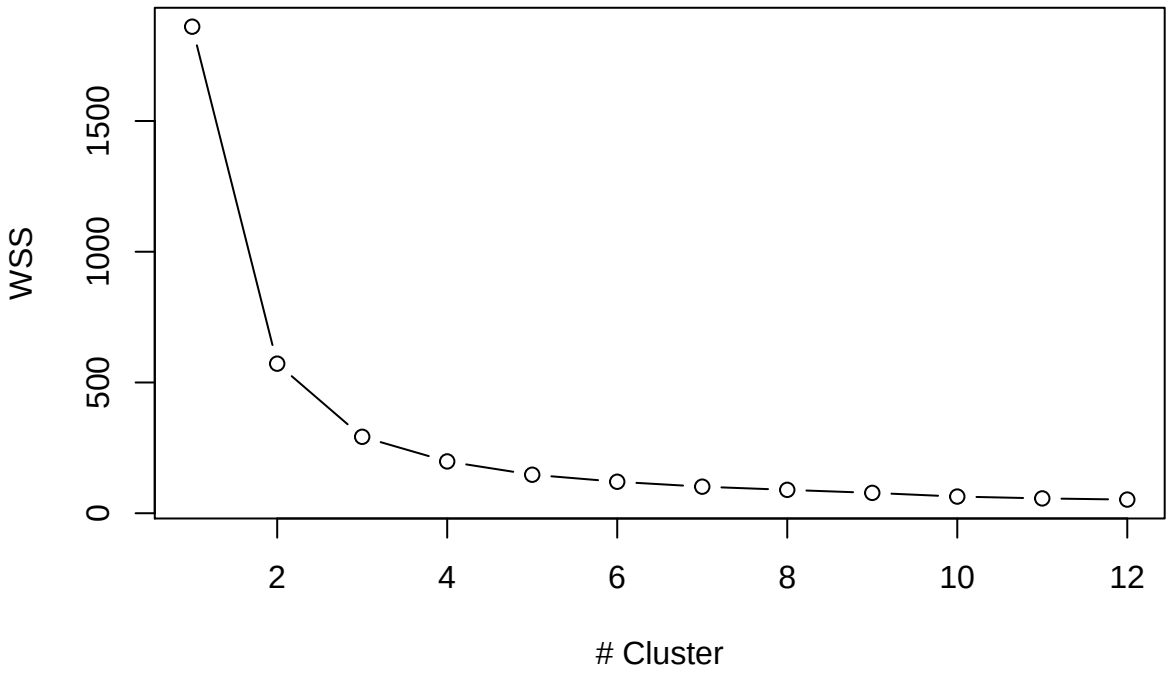
```r
plot(x=wine$alcohol, y=wine$sulphates)
```



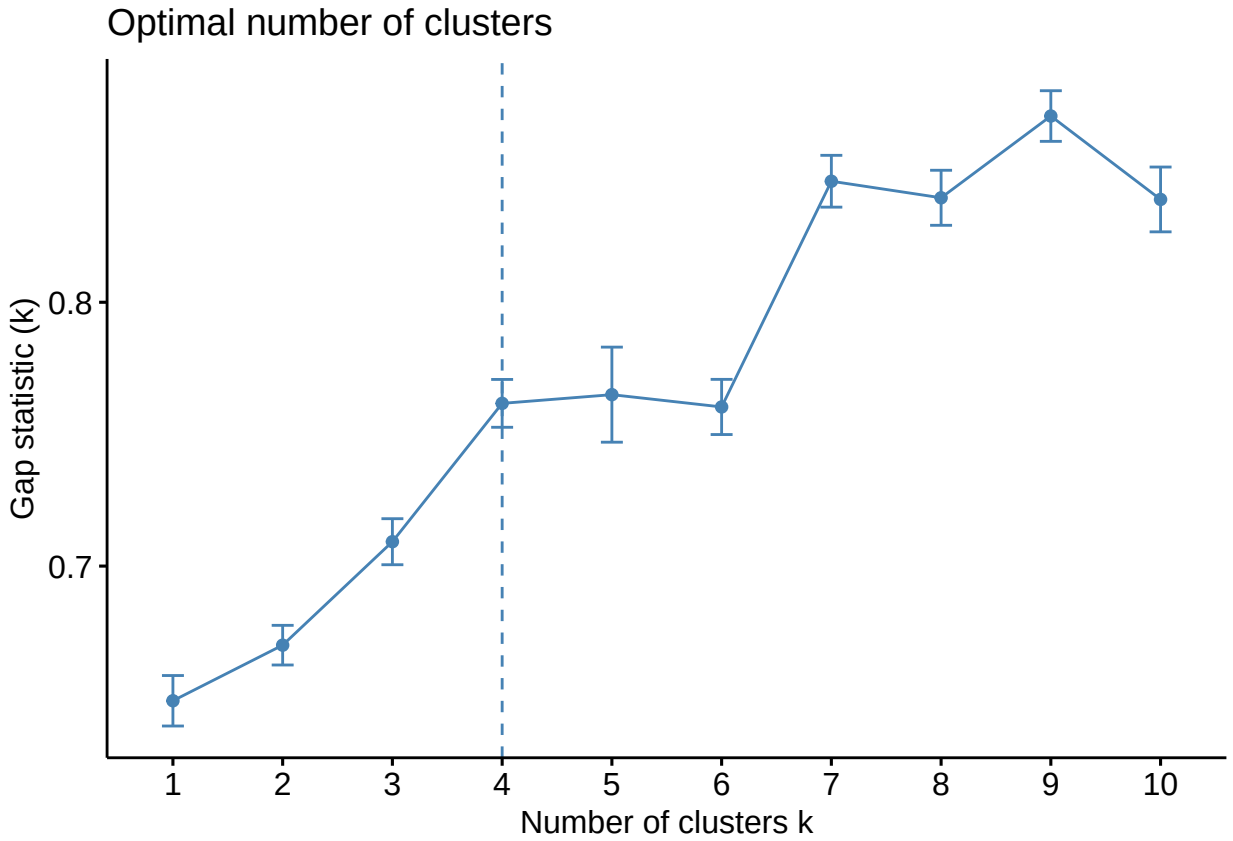

```
#Seleccion de las columnas alcohol y sulphates

wine_s_a <- wine[,10:11]
```

Para determinar el valor optimo de k usaremos dos metodos:

```
# #determinar el numero ideal de cluster grafica con el metodo  Metodo WSS


wss<- 0
for (i in 1:12) {
  out <- kmeans(wine_s_a, centers = i, nstart = 20)
  wss[i] <- out$tot.withinss
}

plot(1:12, wss,type = "b", xlab = "# Cluster", ylab = "WSS")
```

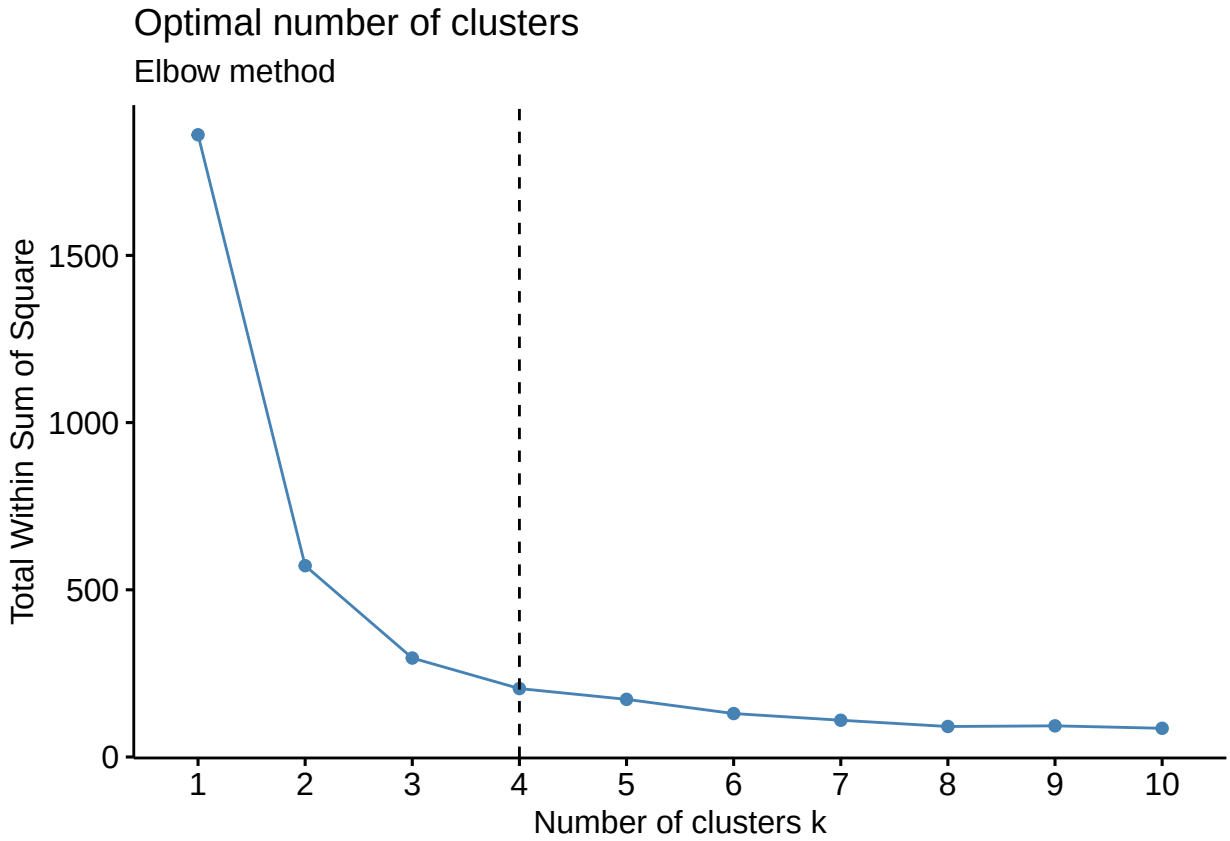



```
#determinar el numero ideal de cluster grafica con el metodo "gap_stat"

fviz_nbclust(wine_s_a, kmeans, method = "gap_stat")
```



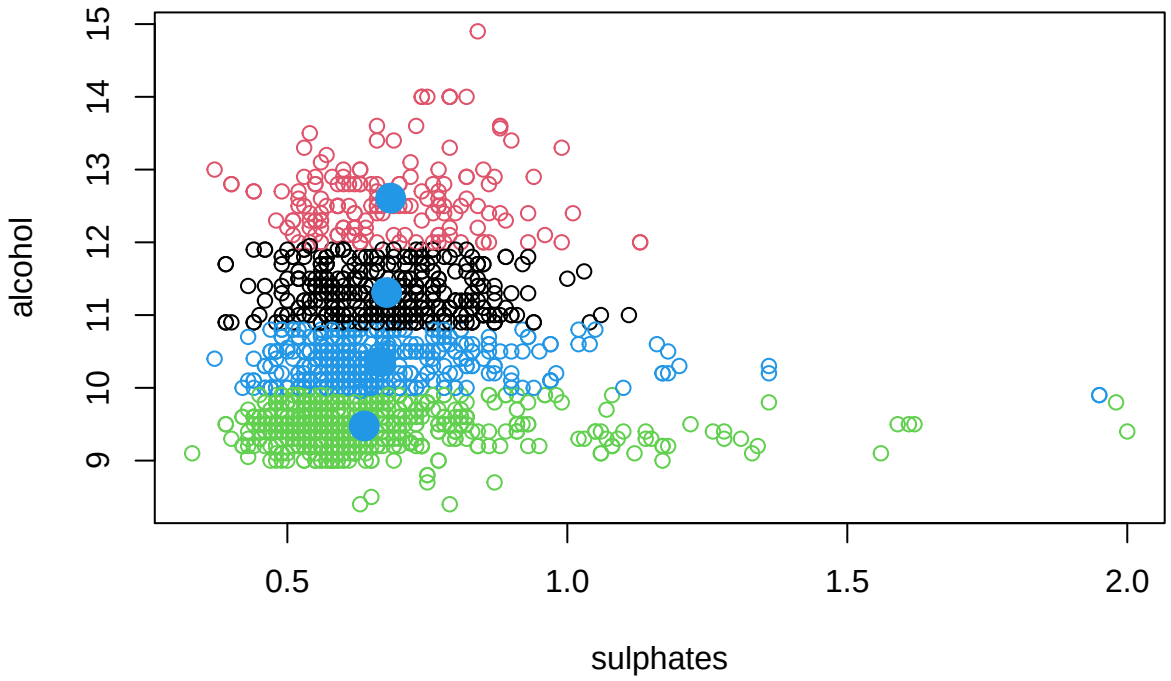

```
#Verificacion del numero de k con la funcion fviz_nbclust de la libreria library(factoextra)
```

```
fviz_nbclust(wine_s_a, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```

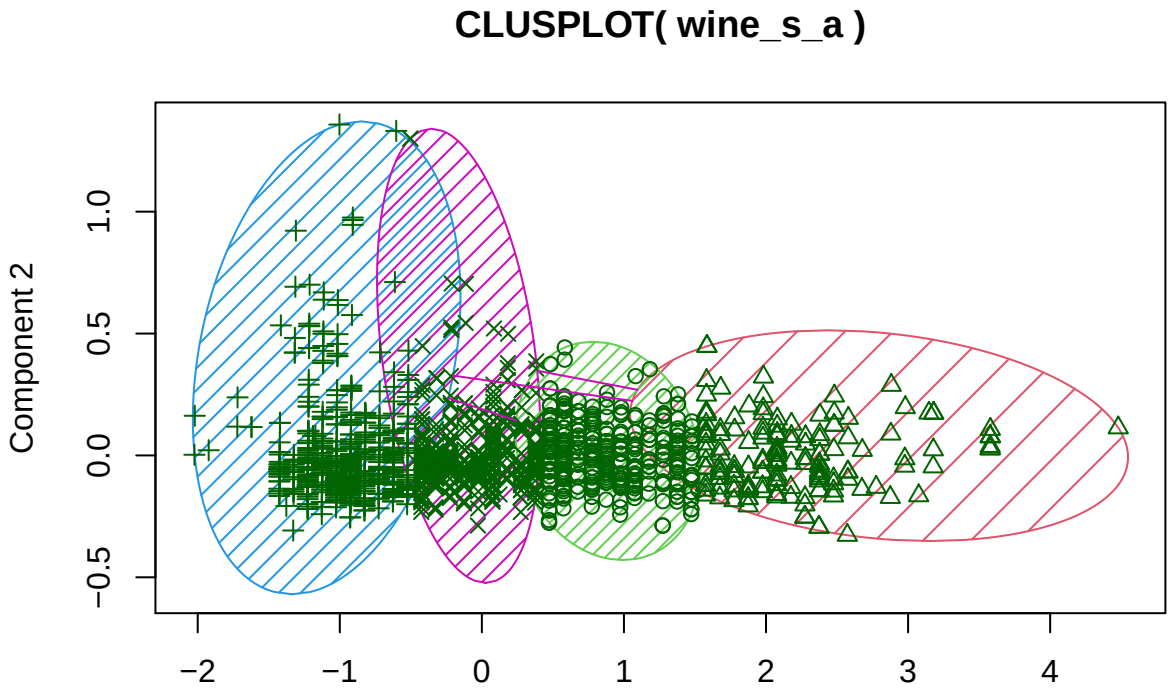


Por medio de la grafica se verifica que el numero optimo para K es 4

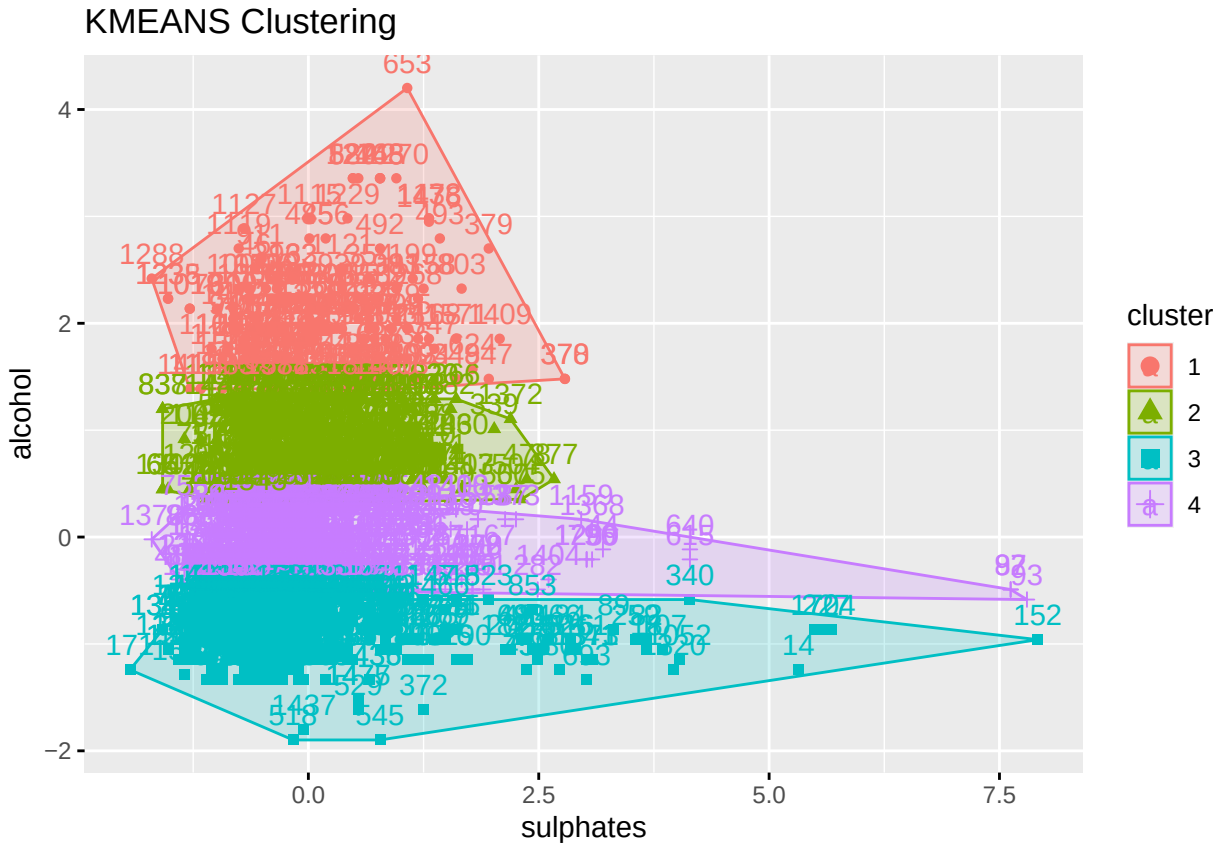
```
# k-means
wine_clus <- kmeans(wine_s_a, center= 4, nstart = 20)
plot(wine_s_a, col=wine_clus$cluster)
points(wine_clus$centers, cex=2, col=12, pch=19)
```




```
clusplot(wine_s_a, wine_clus$cluster, color=TRUE, shade=TRUE)
```




```
eclust(wine_s_a, "kmeans", k = 4, nstart = 20, graph = TRUE)
```




```
## K-means clustering with 4 clusters of sizes 183, 376, 629, 411
##
## Cluster means:
##   sulphates   alcohol
## 1 0.6777596 12.526867
## 2 0.6786968 11.218395
## 3 0.6338315  9.444197
## 4 0.6678345 10.256488
##
## Clustering vector:
##    [1] 3 3 3 3 3 3 3 4 3 4 3 4 4 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 3 3 3 3 2
##   [38] 3 3 4 4 3 4 4 3 1 3 3 3 3 3 3 3 3 4 3 3 3 4 3 3 3 3 3 2 2 3 4 4 4 3 3 3 3
##   [75] 3 4 4 4 4 3 3 3 3 3 4 4 4 3 3 3 3 4 4 3 4 1 4 3 3 3 4 4 3 3 3 3 3 3 3 3 3
##  [112] 3 3 3 3 4 4 3 2 3 3 2 3 3 3 3 2 2 4 3 3 1 1 3 4 3 3 3 3 3 3 3 1 3 1 3 4 3
##  [149] 4 4 4 3 4 4 4 4 4 4 4 3 3 3 3 4 3 3 3 4 3 4 3 3 3 3 2 3 3 3 4 3 3 3 3 3 3
##  [186] 3 3 3 3 3 3 3 3 3 3 3 3 4 4 1 2 4 3 3 3 3 2 2 3 3 4 1 4 4 4 3 4 2 3 3 3 3
##  [223] 3 4 4 4 4 3 3 4 2 1 4 3 2 3 3 3 3 3 3 3 2 3 3 3 4 3 3 3 4 4 4 4 3 4 3 3 3
##  [260] 4 3 3 4 3 2 4 3 1 3 4 4 4 4 3 3 4 3 4 1 4 3 4 3 4 3 3 4 4 4 3 4 4 4 4 3 3
##  [297] 4 3 3 3 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 2 2 4 4 2 4 2 3 3 3 3 3 2 2 3 3 1 1 3
##  [334] 2 2 1 1 4 2 3 3 2 3 3 4 4 1 1 4 3 4 3 3 1 1 1 2 2 4 3 3 4 4 4 4 2 4 3 3 1
##  [371] 4 3 4 3 2 2 2 1 1 2 3 4 3 3 3 3 3 3 3 4 1 4 3 3 4 1 4 2 2 3 4 2 4 4 3 4 2
##  [408] 2 2 3 3 3 3 2 3 3 2 4 4 4 2 1 2 2 2 1 2 4 3 4 2 3 1 3 4 3 4 2 4 3 3 2 2 2
##  [445] 1 4 4 2 4 2 2 3 4 2 2 1 4 3 2 3 2 3 2 3 3 3 2 1 3 3 4 2 4 4 3 4 3 2 4 4 3
##  [482] 2 3 3 1 4 4 4 2 4 4 1 1 2 1 2 3 2 2 2 3 1 1 2 2 1 2 3 4 2 3 4 3 2 2 3 1 3
##  [519] 2 2 4 3 3 3 3 4 2 2 3 3 4 4 4 1 4 4 3 4 1 2 3 2 4 2 3 3 3 4 4 3 3 4 4 2 2
##  [556] 2 2 2 2 1 2 3 3 4 1 2 3 3 4 2 2 2 4 3 4 4 4 3 3 3 3 3 3 4 4 3 4 3 1 4 3 2
##  [593] 3 3 3 3 3 4 3 3 4 3 3 3 3 3 1 4 4 1 3 3 4 3 4 3 3 3 4 3 3 3 3 1 4 4 3 3 3
```

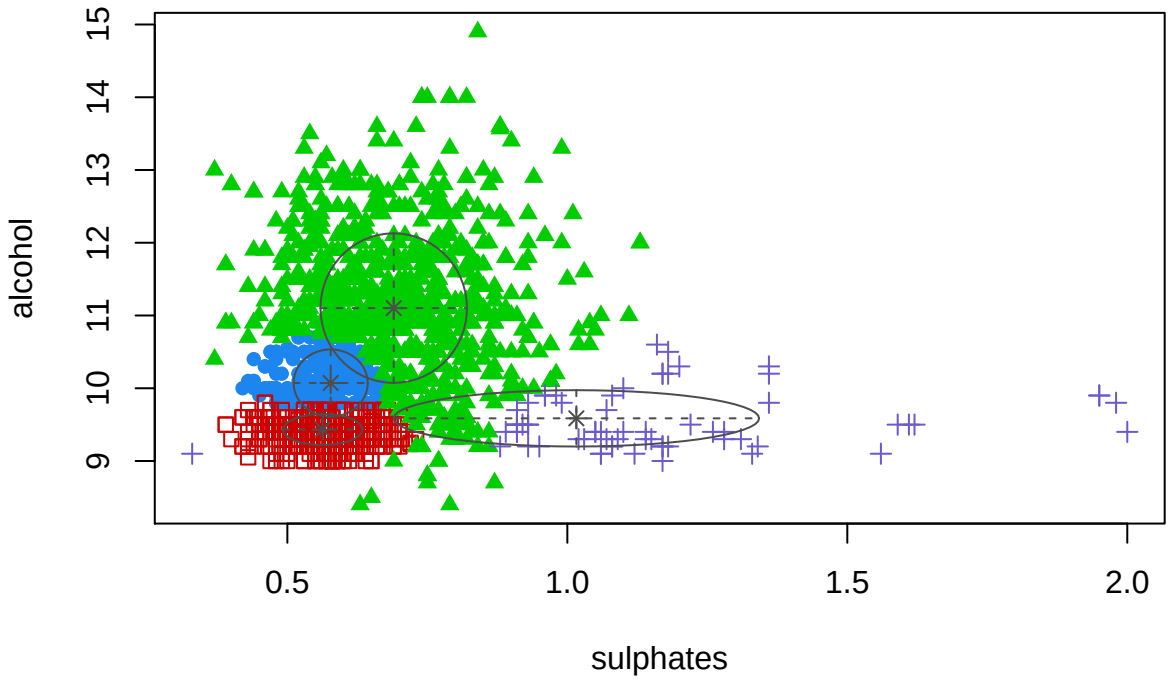
```
##  [630] 3 3 3 4 2 3 3 3 3 4 4 3 3 3 3 3 4 4 2 2 2 3 2 1 1 3 3 3 4 3 2 3 3 3 4 4 3
##  [667] 3 3 2 3 4 3 3 3 3 4 3 4 4 3 3 4 3 2 3 2 3 3 3 3 4 3 3 3 3 1 3 3 3 2 4 3 3
##  [704] 3 3 4 4 4 2 4 4 3 3 3 3 3 3 4 3 3 3 3 4 3 2 4 2 3 3 1 3 2 3 3 3 3 3 3 3 3
##  [741] 2 3 3 4 3 3 3 3 3 3 3 3 3 3 3 4 2 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 4 4 4
##  [778] 4 4 3 3 3 4 3 3 3 3 4 4 3 3 3 3 2 1 4 3 2 4 4 3 4 1 3 4 1 1 1 3 4 4 2 2 2
##  [815] 2 2 4 1 3 3 3 1 3 3 4 4 2 4 1 2 2 2 4 4 3 3 2 2 2 4 1 4 2 3 4 3 3 4 3 3 3
##  [852] 3 3 2 2 2 2 2 4 2 3 2 4 3 3 3 2 2 2 2 2 4 4 2 2 2 2 2 4 3 4 2 1 3 4 4 4 1
##  [889] 2 3 4 3 3 3 3 4 1 4 1 2 2 2 2 2 2 3 2 2 2 2 1 2 1 1 2 1 2 4 2 2 4 2 2 4 2
##  [926] 2 4 3 2 1 4 3 3 3 4 2 2 4 1 2 1 1 4 3 1 2 1 1 1 1 1 1 2 1 1 4 2 2 2 4 2 3
##  [963] 3 2 2 2 1 3 1 4 2 2 2 4 1 3 3 3 2 4 4 4 1 4 4 2 2 3 3 2 3 3 3 3 3 4 1 1 3
## [1000] 1 2 4 2 1 4 1 2 2 1 2 1 4 4 4 2 2 1 1 1 4 1 1 4 1 4 4 1 2 4 4 2 2 3 4 3 4
## [1037] 1 3 1 2 4 3 2 2 1 2 4 4 2 2 4 3 1 1 3 3 1 3 2 1 4 1 2 2 2 4 1 2 2 2 2 3 3
## [1074] 4 3 4 2 2 2 1 2 1 3 1 3 3 2 2 4 4 2 2 2 1 2 4 2 3 1 3 1 1 2 1 1 1 1 2 3 2
## [1111] 4 1 2 3 1 2 2 2 1 1 1 1 1 2 2 2 1 2 4 4 4 3 1 2 2 1 2 2 3 3 3 2 2 4 4 4 3
## [1148] 2 2 1 1 1 3 2 2 3 2 1 4 2 2 4 2 4 4 3 4 1 1 2 4 2 1 3 3 2 2 1 2 2 2 2 4 3
## [1185] 2 1 1 1 2 3 1 2 1 2 3 3 3 3 2 3 3 2 2 3 2 2 2 4 2 2 4 3 4 4 4 2 3 1 4 2 2
## [1222] 2 3 1 4 3 3 3 1 3 2 2 3 3 1 2 3 1 3 2 3 3 1 3 4 4 4 4 2 2 2 3 3 3 2 2 3 2
## [1259] 2 2 3 2 3 3 2 4 4 2 3 1 1 2 2 3 2 3 1 3 3 2 4 4 2 4 2 4 1 1 4 4 2 2 1 3 2
## [1296] 3 3 1 1 2 1 2 2 2 3 3 4 2 4 3 3 1 4 4 4 3 1 1 3 3 4 1 2 2 4 4 4 4 3 3 3 3
## [1333] 3 3 4 1 3 3 3 4 4 4 4 4 2 4 2 3 3 2 4 2 4 4 4 4 4 2 3 4 2 4 4 3 2 3 3 4 3
## [1370] 4 3 2 3 3 3 4 4 1 4 2 2 3 4 4 3 3 3 3 4 3 1 2 3 4 3 3 3 4 4 2 4 4 1 4 4 2
## [1407] 2 4 1 4 1 4 2 4 3 3 3 1 3 4 3 3 2 2 3 3 1 2 2 2 4 4 2 4 3 3 3 2 4 2 2 3 3
## [1444] 2 2 3 3 3 3 2 2 2 2 3 2 2 4 3 4 1 4 4 4 2 3 3 4 2 4 3 3 1 2 2 3 1 3 1 4 4
## [1481] 4 4 4 4 2 3 4 4 2 2 1 2 1 3 4 4 3 2 2 2 3 3 3 2 2 4 2 2 4 2 3 3 4 4 3 3 4
## [1518] 4 4 4 4 3 4 4 4 3 3 2 4 4 2 4 4 3 2 3 4 4 1 4 2 2 2 4 2 4 4 1 4 2 3 3 2 3
## [1555] 2 4 4 2 3 4 4 4 4 4 4 2 2 4 3 2 1 2 3 1 4 2 2 1 2 2 1 2 1 3 2 2 2 2 1 3 2
## [1592] 2 2 3 4 2 2 4 2
##
## Within cluster sum of squares by cluster:
## [1] 58.82176 42.80950 55.91634 40.59716
##  (between_SS / total_SS =  89.4 %)
##
## Available components:
##
##  [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
##  [6] "betweenss"    "size"         "iter"         "ifault"       "clust_plot"
## [11] "silinfo"      "nbclust"      "data"
```

```
# GMM
gmm_wine = Mclust(wine_s_a, G = 4)
summary(gmm_wine, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVI (diagonal, varying volume and shape) model with 4 components:
##
##  log-likelihood    n df       BIC      ICL
##       -1128.705 1599 19 -2397.576 -3185.93
##
## Clustering table:
##   1   2   3   4
## 303 448 782  66
##
```

```
## Mixing probabilities:
##         1         2         3         4
## 0.1930452 0.2331972 0.5139746 0.0597830
##
## Means:
##                 [,1]      [,2]       [,3]     [,4]
## sulphates  0.5773263 0.5631875  0.6899573 1.016083
## alcohol   10.0718650 9.4327743 11.1014611 9.586210
##
## Variances:
## [,,1]
##             sulphates   alcohol
## sulphates 0.004376627 0.0000000
## alcohol   0.000000000 0.2131437
## [,,2]
##             sulphates   alcohol
## sulphates 0.004993995 0.0000000
## alcohol   0.000000000 0.0450906
## [,,3]
##            sulphates  alcohol
## sulphates 0.01710889 0.000000
## alcohol   0.00000000 1.057248
## [,,4]
##           sulphates   alcohol
## sulphates 0.1065265 0.0000000
## alcohol   0.0000000 0.1499837
```

```
plot(gmm_wine, what = "classification")
```



CONCLUSION:

1. El metodo GMM es el que mejor realiza el agrupamiento de las muestras, puesto que el metodo de K-means establece una fontera estricta para clasificar los datos en los diferentes cluster, mientras que GMM evalua la probabilidad de cada muestra para pertenecer a un cluster.

2. Al agrupar en 4 cluster se podria clasificar al vino en calidad alta, moderada, baja y muy baja.

Bibliografía

[1] Dataset = https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

[2]Clustering = https://conceptosclaros.com/que-es-clustering/

[3] GMM = https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

[4] Rattle = https://rstudio-pubs-static.s3.amazonaws.com/317830_dde0f80173c047a2badbbe8918e95048.html