Clustering

Ana Laura Diedrichs

May 22, 2019

```
library(readxl)
library(DataExplorer)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
#dataset <- read_excel("data/dataset para análisis exploratorio.xlsx")
dataset <- read_excel("data/Dataset para enfoque 2 y 3.xlsx")
# preparacion de datos
#data <- dataset[-1]
#data <- scale(data)
```

```
# normalizamos los datos entre 0 y 1
range01 <- function(x){(x-min(x))/(max(x)-min(x))}

dataset <- data.frame(dataset[,1], apply(dataset[-1],2,range01))

data <- dataset[-1]
```

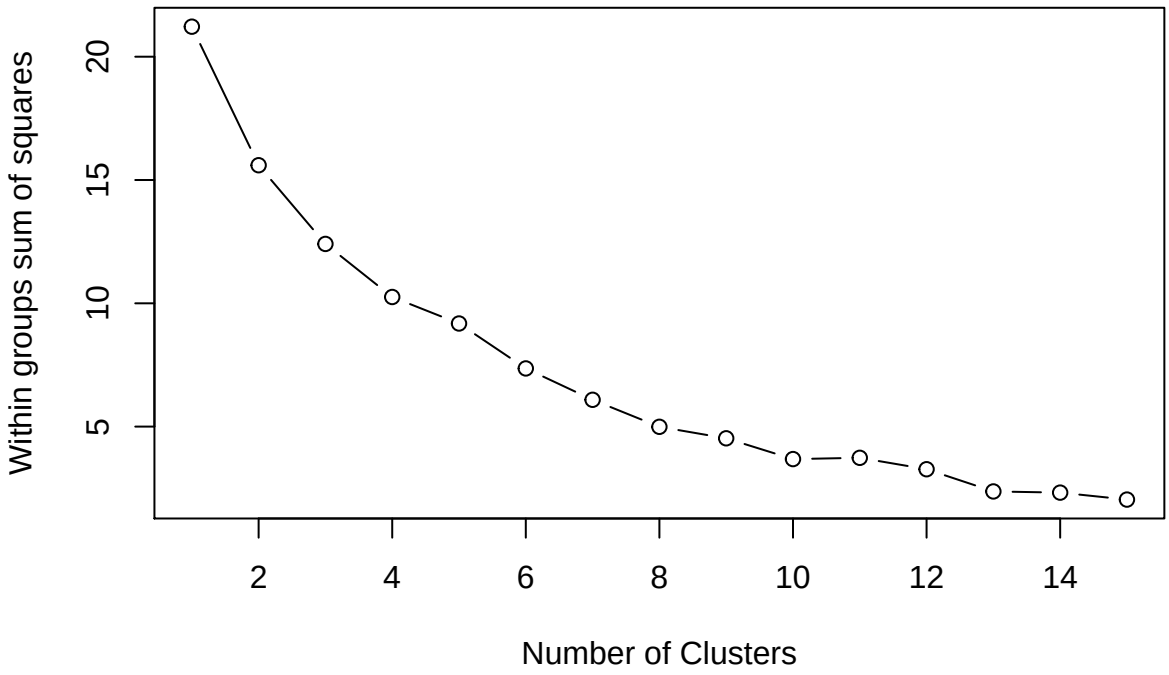
Clustering intro

Vamos a aplicar un enfoque no supervisado sobre los datos mediante agrupamiento, sin considerar la etiqueta Origen o variable de clase. Sí usaremos la misma para relacionar y analizar el agrupamiento obtenido.

Primero usaremos k-means. Previo al uso de este algoritmo, los datos no deben tener valores NULOS o perdidos y son escalados entre 0 y 1.

En el siguiente gráfico muestra como la suma del cuadrado de las distancias intra-cluster disminuye a medida que se agrega un cluster (aumenta k) en kmeans. Computamos el WSS para distintos números de clusteres para k-means para entender como disminuye el total within-cluster sum of square (WSS) a medida que se incrementan los clústeres.

```
# Determine number of clusters
wss <- (nrow(data)-1)*sum(apply(data,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(data,
   centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```



Observamos que de un k=1 a un k=2 disminuye en un tercio el WSS.

K-means

Aplicamos k-means clustering considerando $k = 2$.

```
set.seed(11235)
# K-Means Cluster Analysis
fit <- kmeans(data, centers=2,nstart = 50) # 2 cluster solution
# append cluster assignment
mydata <- data.frame(data, fit$cluster)
fit
```

```
## K-means clustering with 2 clusters of sizes 16, 15
##
## Cluster means:
##    As...ppb. Cr...ppb. Cu...ppb.  Fe...ppb. Mn...ppb.   Mo..ppb. Ni...ppb.
## 1 0.02177439 0.3260808 0.2099693 0.09094435 0.2404567 0.04964936 0.2702516
## 2 0.14227320 0.8226218 0.4467101 0.54553723 0.3291852 0.19530516 0.2715845
##   Pd...ppb. Rb...ppb.   V...ppb.   Y...ppb.
## 1 0.1490135 0.1947900 0.08397657 0.06601686
## 2 0.2914761 0.1357101 0.26873709 0.40505617
##
## Clustering vector:
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1]  3.945621 11.653276
##  (between_SS / total_SS =  26.5 %)
##
## Available components:
##
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
#cbind(mydata$fit.cluster,dataset$Origen)
```

```
# get cluster means
aggregate(data,by=list(fit$cluster),FUN=mean)
```

```
##   Group.1  As...ppb. Cr...ppb. Cu...ppb.  Fe...ppb. Mn...ppb.   Mo..ppb.
## 1       1 0.02177439 0.3260808 0.2099693 0.09094435 0.2404567 0.04964936
## 2       2 0.14227320 0.8226218 0.4467101 0.54553723 0.3291852 0.19530516
##   Ni...ppb. Pd...ppb. Rb...ppb.   V...ppb.   Y...ppb.
## 1 0.2702516 0.1490135 0.1947900 0.08397657 0.06601686
## 2 0.2715845 0.2914761 0.1357101 0.26873709 0.40505617
```

En la siguiente tabla observamos la distribución de observaciones entre los dos clústeres, según el origen

```
table(mydata$fit.cluster,dataset$Origen)
```

```
##
##     AR BR CH
##   1  2  9  5
##   2 14  1  0
```

Observamos que para el clúster "2" han sido asignadas la mayoría de las muestras de Argentina, unas 14 en total y tan solo 1 de Brasil.

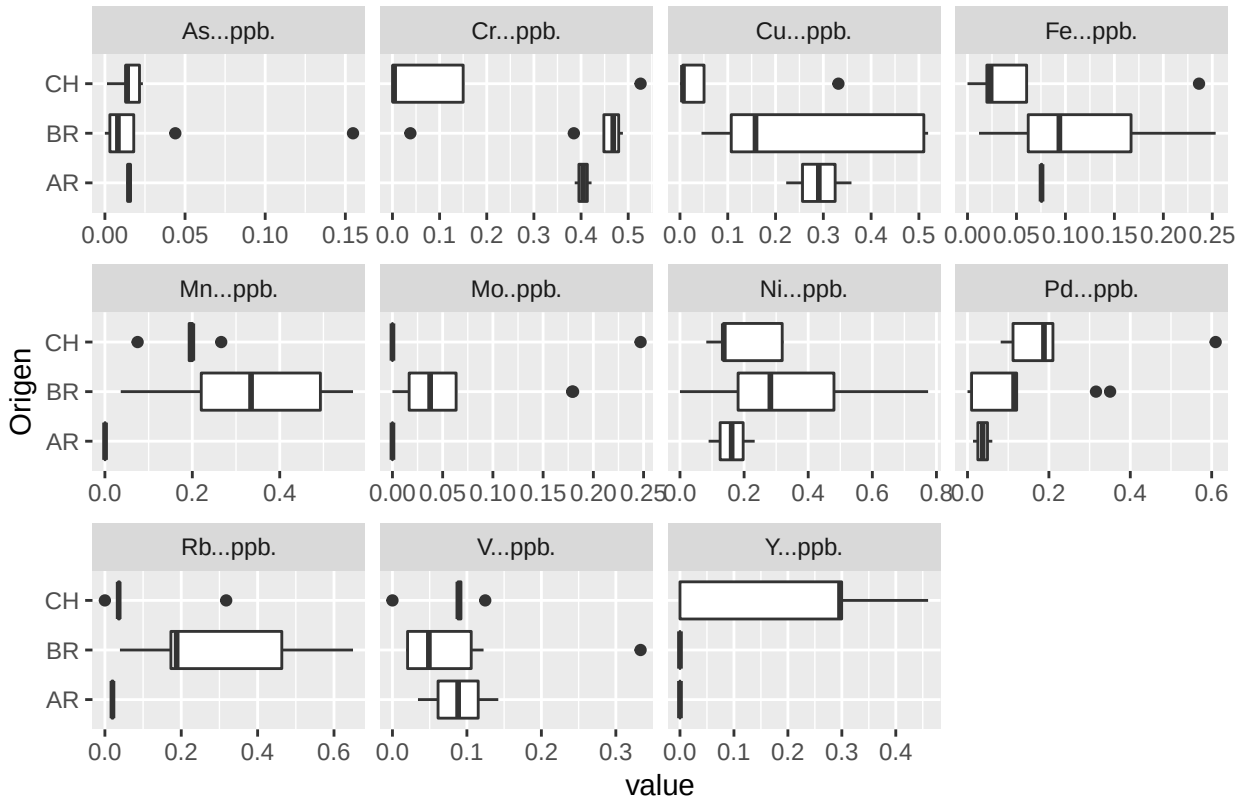
Análisis o visualización de clústers, k=2

```
k.1 <- dataset[fit$cluster==1,]
k.2 <- dataset[fit$cluster==2,]
```

A continuación mostramos la distribución (histograma) de las muestras que fueron agrupadas para el clúster 1 y para el 2.

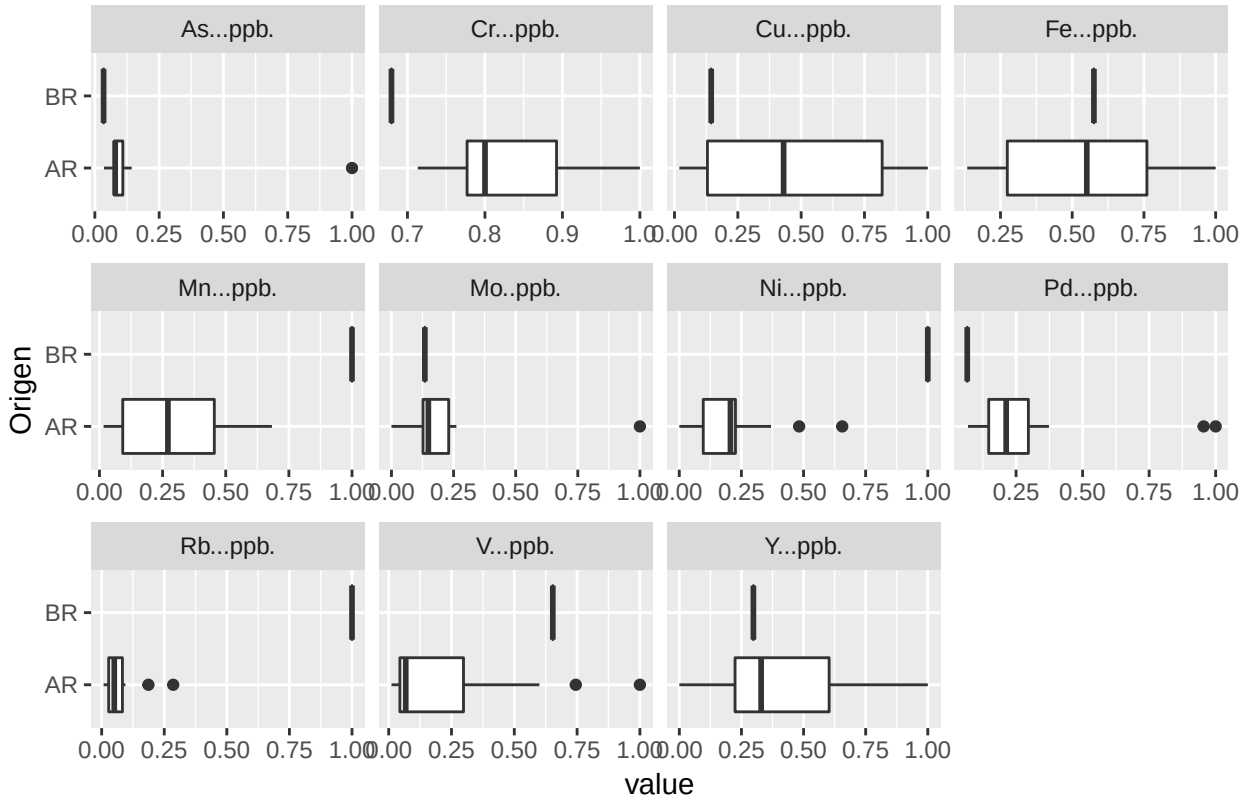
¿Qué características podemos extraer de esto?

```
plot_boxplot(k.1, by = "Origen")
```



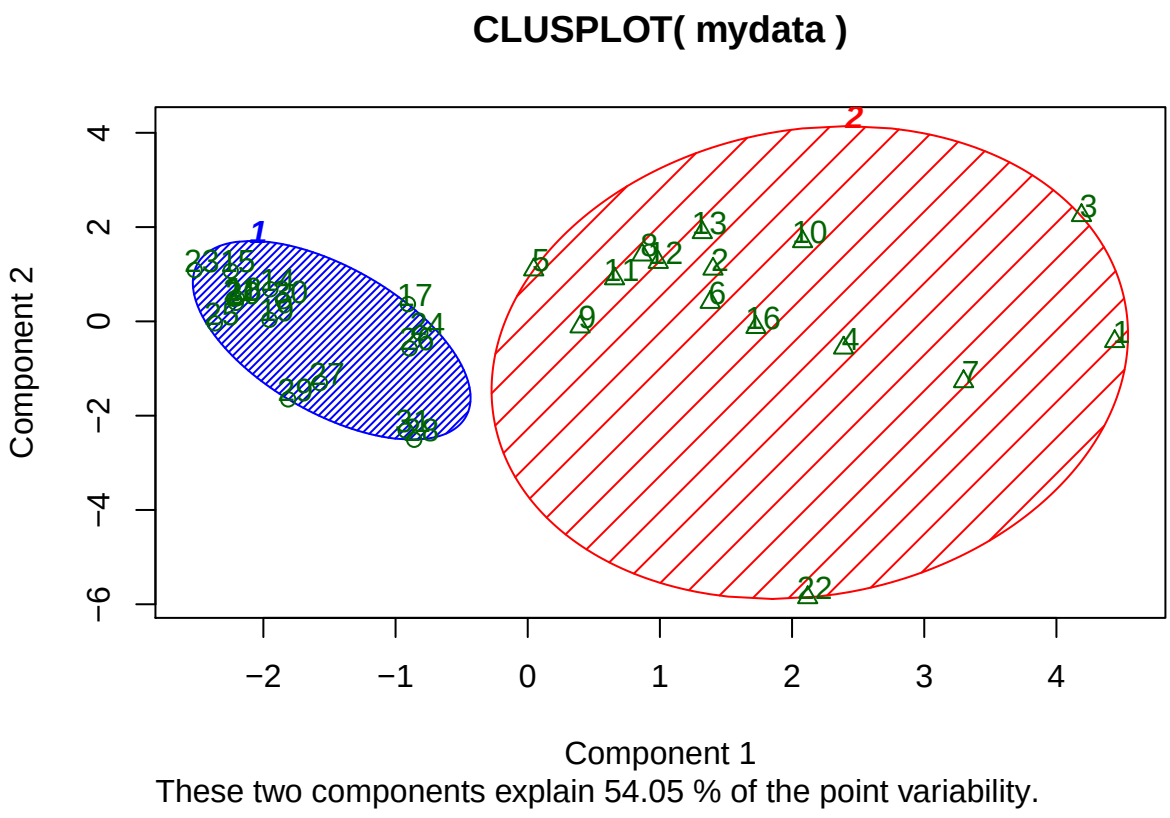
Cluster 2

```
plot_boxplot(k.2, by = "Origen")
```



Para graficar las observaciones agrupadas en los clusteres, dado que tenemos más de dos dimensiones, se nos complica graficar mas bien visualizar en dos ejes. Usamos la libreria cluster que reduce las dimensiones y permite graficar los clústeres.

```
library(cluster)
clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)
```



Ahora k= 3

Realizamos el mismo experimento pero con $k = 3$

```
set.seed(11235)
# K-Means Cluster Analysis
fit2 <- kmeans(data, centers=3)
# get cluster means
aggregate(data,by=list(fit2$cluster),FUN=mean)
```

```
##   Group.1  As...ppb. Cr...ppb. Cu...ppb.  Fe...ppb. Mn...ppb.   Mo..ppb.
## 1       1 0.02177439 0.3260808 0.2099693 0.09094435 0.2404567 0.04964936
## 2       2 0.21338449 0.8718660 0.8301140 0.31896319 0.4055213 0.23428099
## 3       3 0.08005082 0.7795332 0.1112318 0.74378951 0.2623912 0.16120131
##   Ni...ppb. Pd...ppb. Rb...ppb.   V...ppb.   Y...ppb.
## 1 0.2702516 0.1490135 0.1947900 0.08397657 0.06601686
## 2 0.1479412 0.2948596 0.1196856 0.25307135 0.28618299
## 3 0.3797724 0.2885155 0.1497316 0.28244462 0.50907021
```

```
# append cluster assignment
mydata <- data.frame(data, fit2$cluster)
```

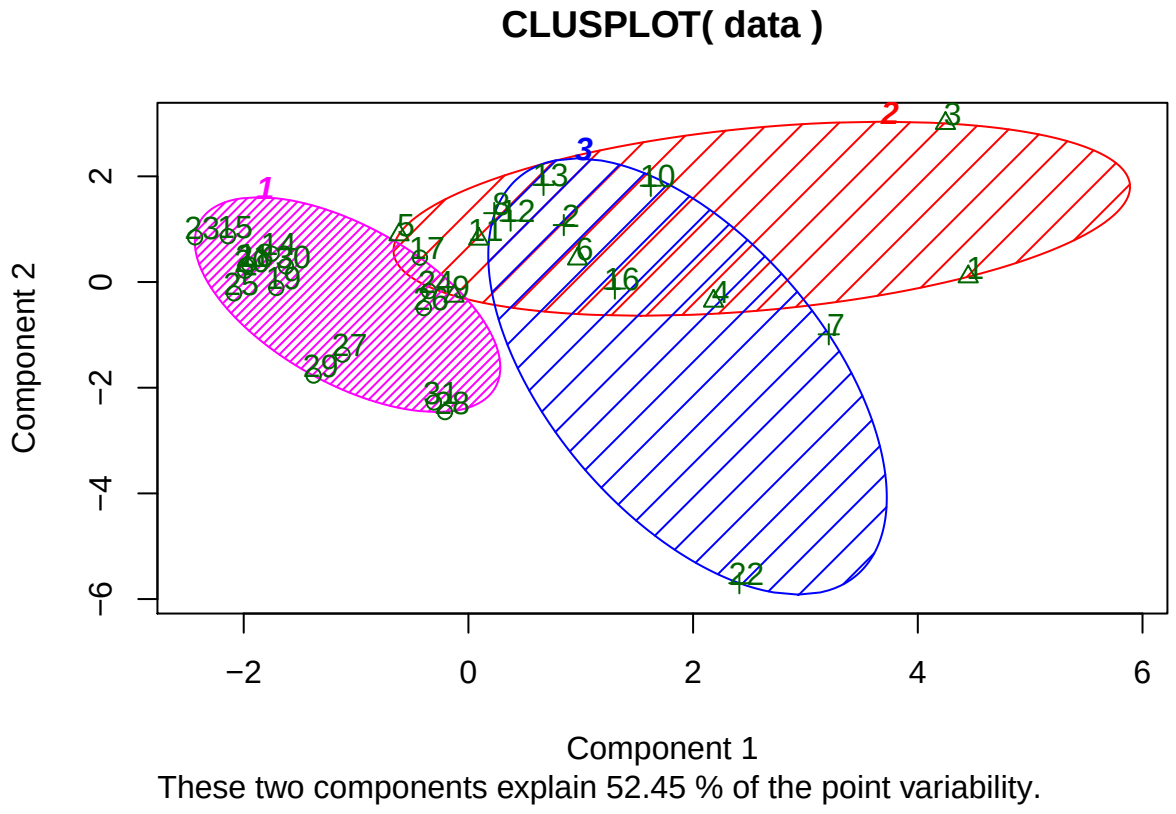
```
#cbind(mydata$fit.cluster,dataset$Origen)
```

En la siguiente tabla observamos la distribución de observaciones entre los 3 clústeres, según el origen. Notamos que el cluster 3 tiene la mitad de las muestras de Brasil, el cluster 2 la mayoría de las muestras de Argentina, un 43.75 %.

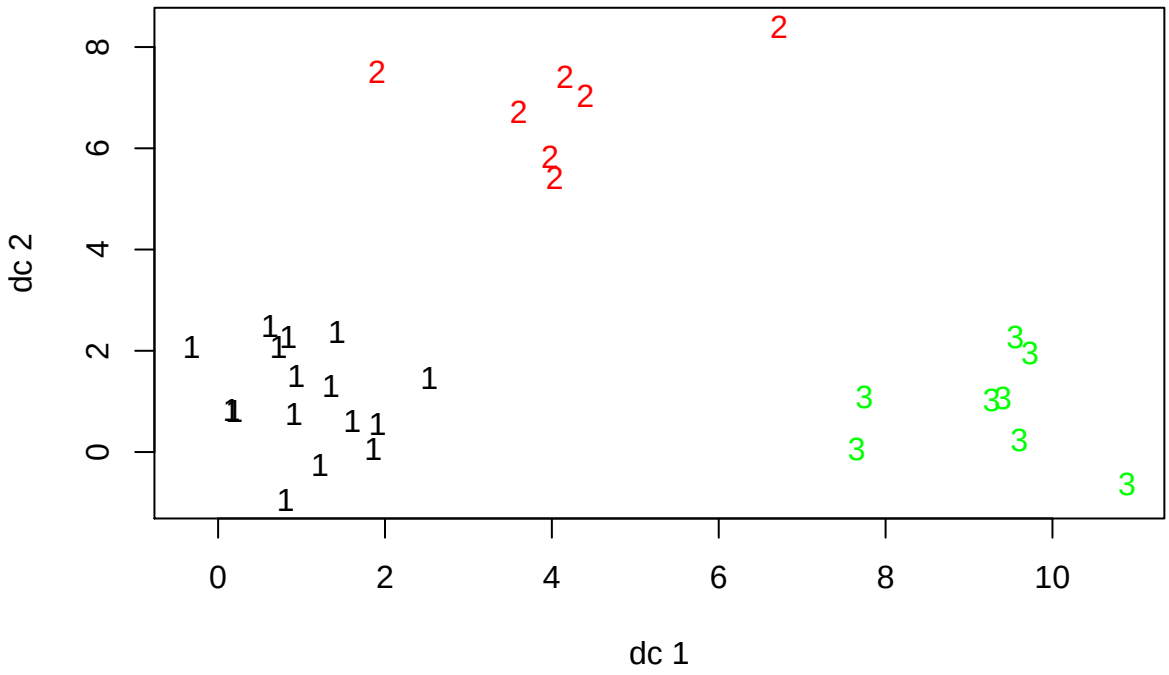
```
table(mydata$fit2.cluster,dataset$Origen)
```

```
##
##      AR BR CH
##    1  2  9  5
##    2  7  0  0
##    3  7  1  0
```

```
# vary parameters for most readable graph
clusplot(data, fit2$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)
```




```
# Centroid Plot against 1st 2 discriminant functions
library(fpc)
plotcluster(data, fit2$cluster)
```



Comparamos las dos soluciones de clustering

Estadisticas de cluster con k=2

```
library(fpc)
distancia <- dist(data)
cluster.stats(distancia, fit$cluster)
```

```
## $n
## [1] 31
##
## $cluster.number
## [1] 2
##
## $cluster.size
## [1] 16 15
##
## $min.cluster.size
## [1] 15
##
## $noisen
## [1] 0
##
## $diameter
## [1] 1.237522 2.146951
##
## $average.distance
## [1] 0.6902918 1.2279405
##
## $median.distance
## [1] 0.6847123 1.2982007
##
## $separation
## [1] 0.6401894 0.6401894
```

```
##
## $average.toother
## [1] 1.286009 1.286009
##
## $separation.matrix
##           [,1]      [,2]
## [1,] 0.0000000 0.6401894
## [2,] 0.6401894 0.0000000
##
## $ave.between.matrix
##          [,1]     [,2]
## [1,] 0.000000 1.286009
## [2,] 1.286009 0.000000
##
## $average.between
## [1] 1.286009
##
## $average.within
## [1] 0.9411945
##
## $n.between
## [1] 240
##
## $n.within
## [1] 225
##
## $max.diameter
## [1] 2.146951
##
## $min.separation
## [1] 0.6401894
##
## $within.cluster.ss
## [1] 15.5989
##
## $clus.avg.silwidths
##          1          2
## 0.46243818 0.03631062
##
## $avg.silwidth
## [1] 0.2562474
##
## $g2
## NULL
##
## $g3
## NULL
##
## $pearsongamma
## [1] 0.4281639
##
## $dunn
## [1] 0.2981853
##
```

```
## $dunn2
## [1] 1.04729
##
## $entropy
## [1] 0.6926268
##
## $wb.ratio
## [1] 0.7318722
##
## $ch
## [1] 10.44555
##
## $cwidegap
## [1] 0.5378624 1.4462664
##
## $widestgap
## [1] 1.446266
##
## $sindex
## [1] 0.6539115
##
## $corrected.rand
## NULL
##
## $vi
## NULL
```

Estadisticas de cluster k=3

```
library(fpc)
distancia <- dist(data)
cluster.stats(distancia, fit2$cluster)
```

```
## $n
## [1] 31
##
## $cluster.number
## [1] 3
##
## $cluster.size
## [1] 16  7  8
##
## $min.cluster.size
## [1] 7
##
## $noisen
## [1] 0
##
## $diameter
## [1] 1.237522 1.612465 1.891956
##
## $average.distance
## [1] 0.6902918 1.0594732 1.0536746
##
## $median.distance
```

```
## [1] 0.6847123 1.0718467 1.0242286
##
## $separation
## [1] 0.6401894 0.6401894 0.8156456
##
## $average.toother
## [1] 1.286009 1.304586 1.325193
##
## $separation.matrix
##           [,1]      [,2]      [,3]
## [1,] 0.0000000 0.6401894 0.8156456
## [2,] 0.6401894 0.0000000 0.8268388
## [3,] 0.8156456 0.8268388 0.0000000
##
## $ave.between.matrix
##          [,1]     [,2]     [,3]
## [1,] 0.000000 1.267755 1.301982
## [2,] 1.267755 0.000000 1.378249
## [3,] 1.301982 1.378249 0.000000
##
## $average.between
## [1] 1.30346
##
## $average.within
## [1] 0.7963719
##
## $n.between
## [1] 296
##
## $n.within
## [1] 169
##
## $max.diameter
## [1] 1.891956
##
## $min.separation
## [1] 0.6401894
##
## $within.cluster.ss
## [1] 12.40828
##
## $clus.avg.silwidths
##         1         2         3
## 0.4344487 0.1285541 0.1933010
##
## $avg.silwidth
## [1] 0.303144
##
## $g2
## NULL
##
## $g3
## NULL
##
```

```
## $pearsongamma
## [1] 0.6060381
##
## $dunn
## [1] 0.3383743
##
## $dunn2
## [1] 1.19659
##
## $entropy
## [1] 1.026945
##
## $wb.ratio
## [1] 0.6109676
##
## $ch
## [1] 9.939237
##
## $cwidegap
## [1] 0.5378624 1.3792350 1.4462664
##
## $widestgap
## [1] 1.446266
##
## $sindex
## [1] 0.6539115
##
## $corrected.rand
## NULL
##
## $vi
## NULL
```

ANA: de aquí en adelante no realicé análisis aún

Clustering por la mediana:

El uso de la media implica que k-means clustering sea altamente sensible a outliers o valores extremos. Esto puede afectar severamente la asignación de observaciones a los clústeres. El algoritmo PAM es más robusto.

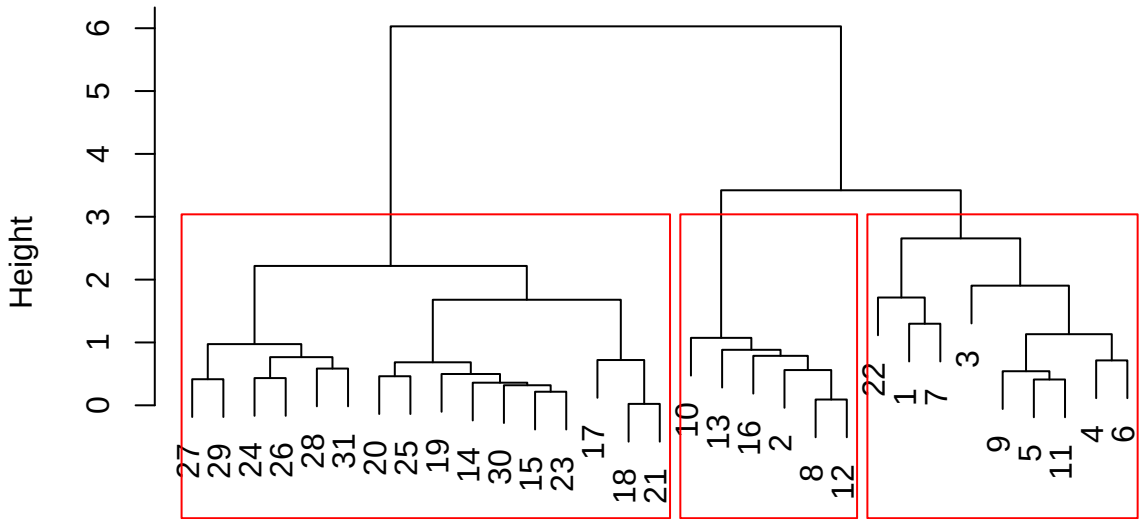
Clustering jerárquico

Ward Hierarchical Clustering

```
# Ward Hierarchical Clustering
d <- dist(data, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
plot(fit) # display dendogram
groups <- cutree(fit, k=3) # cut tree into 5 clusters
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=3, border="red")
```



####

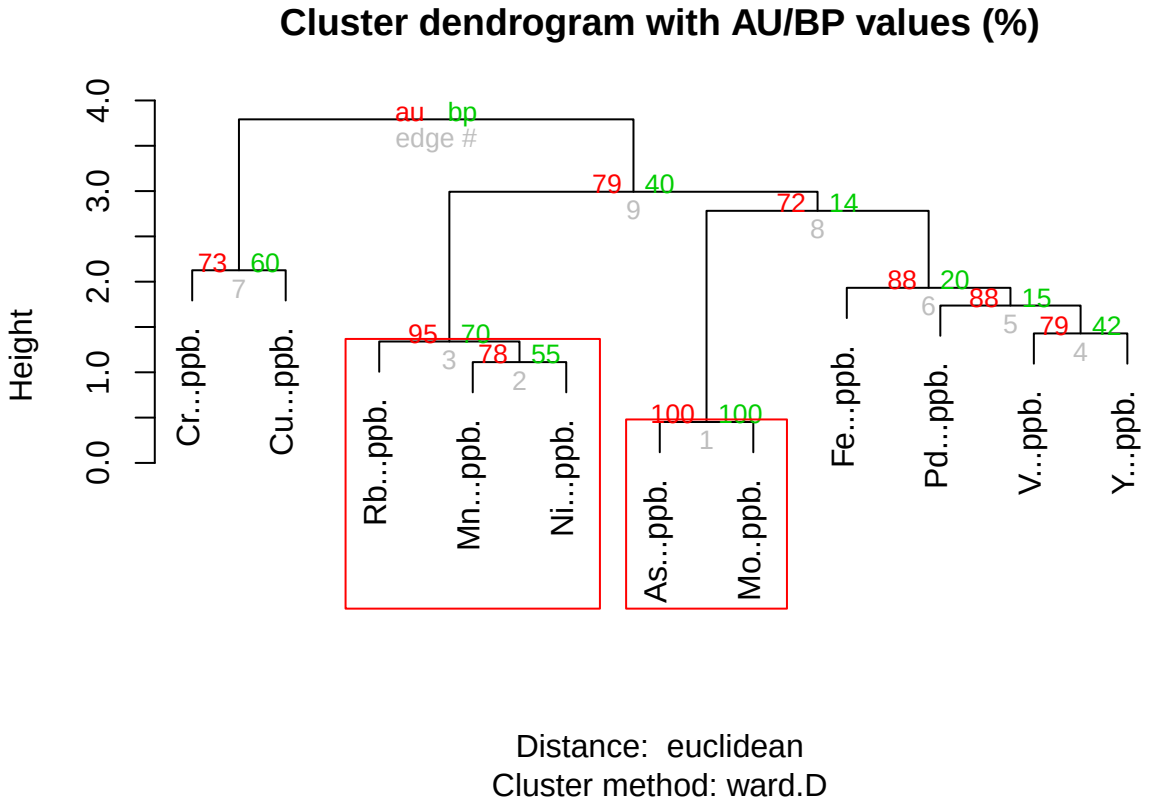
Ward Hierarchical Clustering with Bootstrapped p values

```
library(pvclust)
fit <- pvclust(data, method.hclust="ward",
               method.dist="euclidean")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
## Bootstrap (r = 0.48)... Done.
## Bootstrap (r = 0.58)... Done.
## Bootstrap (r = 0.68)... Done.
## Bootstrap (r = 0.77)... Done.
## Bootstrap (r = 0.87)... Done.
## Bootstrap (r = 1.0)... Done.
## Bootstrap (r = 1.1)... Done.
## Bootstrap (r = 1.19)... Done.
## Bootstrap (r = 1.29)... Done.
## Bootstrap (r = 1.39)... Done.
```

```
plot(fit) # dendogram with p values
# add rectangles around groups highly supported by the data
pvrect(fit, alpha=.95)
```



Mas info o info extra

https://uc-r.github.io/kmeans_clustering

https://www.semanticscholar.org/paper/Clustering-Methods-and-Their-Uses-in-Computational-Downs-Barnard/d81cea597b15deebd940873ee12a4e44019e25af

http://www.bioconductor.org/packages/release/bioc/vignettes/ChemmineR/inst/doc/ChemmineR.html

http://crdd.osdd.net/clusters.php

https://www.sciencedirect.com/topics/chemistry/molecular-cluster