Practica 3

Juan Cantero Jimenez

1/30/2022

Tarea 4

Para la realización de esta tarea se seguirá el guión de la Tarea 3 que se encuentra en la Practica 3

1. Carga los datos y explóralos. ¿Crees que en este caso debes estandarizar las variables antes de realizar un análisis de agrupamiento o no? Justifica brevemente tu respuesta.

```
describe_custom <- function(data){
  require(e1071)
  result <- apply(data, 2, function(x){


    c(media=mean(x),
      mediana=median(x),
      varianza = var(x),
      des_tipic = sd(x),
      skew = e1071::skewness(x),
      kurto = e1071::kurtosis(x),
      maximo = max(x),
      minimo = min(x),
      rango = max(x)- min(x),
      quantile(x , 0.25 ),
      quantile(x, 0.50),
      quantile(x, 0.75),
      shapiro_pvalor = shapiro.test(x)$p.value)

  })
  return(result)
}

load("datoscluster.RData")
head(datosfinal)
```

```
##    country smoking_men alcohol2008 blood_pres_men2008 bmi_men fat_blood_men
## 96  Russia        70.1        16.2                126    22.9          4.70
## 11 Belarus        63.7        18.9                137    26.2          5.02
## 47 Hungary        45.7        16.1                128    25.1          4.31
## 5  Armenia        55.1        13.7                135    25.4          4.71
## 36 Estonia        49.9        17.2                129    20.9          4.11
## 19  Canada        24.3        10.2                124    27.4          5.09
##    mort_c_men TM_Lung_men
## 96       43.1        62.1
## 11       45.2        65.3
```

```
## 47       33.0        94.3
## 5        35.1        70.1
## 36       28.6        61.6
## 19       13.2        48.3
```

```
paises <- datosfinal$country
datosfinal_numeric <- datosfinal[,-1]
describe_custom(datosfinal_numeric)
```

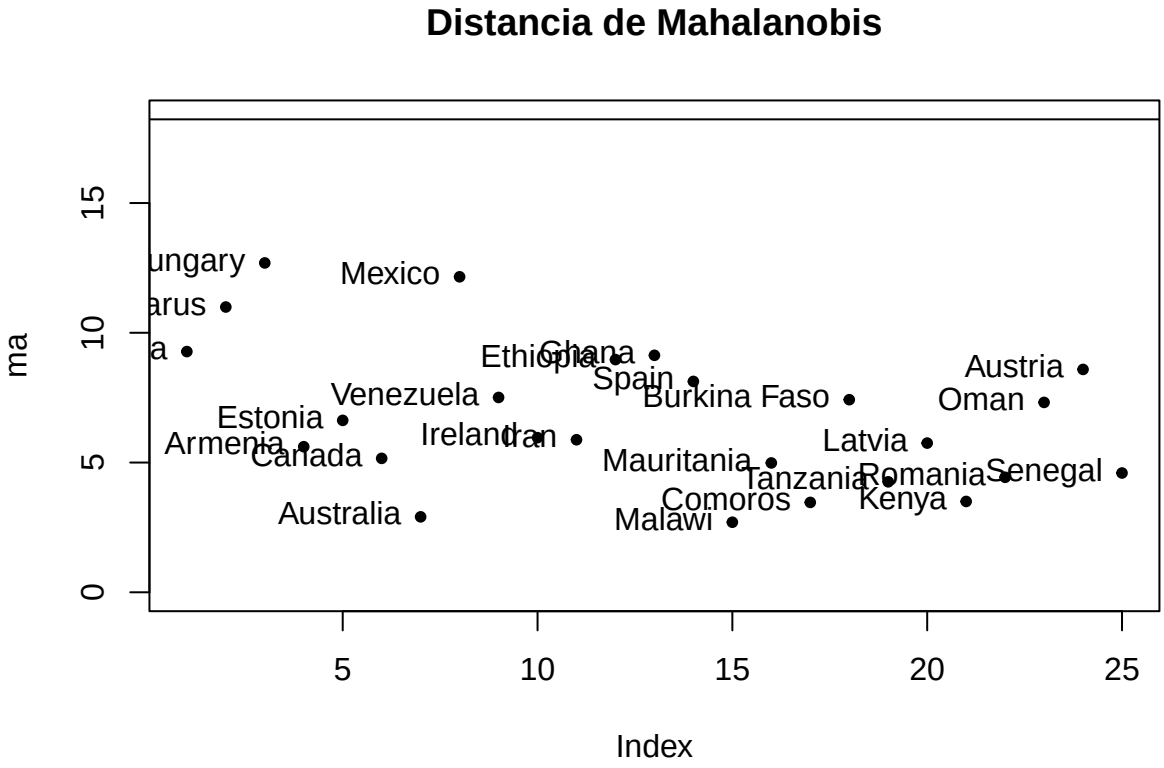
```
## Loading required package: e1071
```

```
##                 smoking_men alcohol2008 blood_pres_men2008     bmi_men
## media            33.9880000  8.71240000        130.8400000 24.74400000
## mediana          27.7000000  8.55000000        131.0000000 25.40000000
## varianza        248.7477667 38.27894400         20.8900000  7.22006667
## des_tipic        15.7717395  6.18699798          4.5705580  2.68701817
## skew              0.5729497 -0.00770998          0.1394046 -0.10219206
## kurto            -0.4857348 -1.50048886         -1.1662420 -1.51195197
## maximo           70.1000000 18.90000000        139.0000000 29.40000000
## minimo            7.6000000  0.11000000        123.0000000 20.90000000
## rango            62.5000000 18.79000000         16.0000000  8.50000000
## 25%              24.3000000  3.11000000        127.0000000 21.90000000
## 50%              27.7000000  8.55000000        131.0000000 25.40000000
## 75%              45.7000000 13.70000000        134.0000000 26.70000000
## shapiro_pvalor    0.1684845  0.05886849          0.3733001  0.02814777
##                 fat_blood_men   mort_c_men   TM_Lung_men
## media               4.6732000 23.184000000  34.795600000
## mediana             4.7100000 19.900000000  25.800000000
## varianza            0.2252477 88.285566667 681.603659000
## des_tipic           0.4746026  9.396039946  26.107540271
## skew               -0.1713004  0.908977567   0.485648110
## kurto              -1.0942494 -0.330223279  -1.130678141
## maximo              5.5600000 45.200000000  94.300000000
## minimo              3.8000000 12.100000000   5.350000000
## rango               1.7600000 33.100000000  88.950000000
## 25%                 4.3100000 17.600000000  10.600000000
## 50%                 4.7100000 19.900000000  25.800000000
## 75%                 5.0700000 28.600000000  60.200000000
## shapiro_pvalor      0.6475618  0.009242466   0.005654532
```

Puesto que existe una diferencia notable en las escalas de las distintas variables, además de no poseer las mismas unidades, se escalarán en los subsiguientes análisis.

2. Realiza un análisis de outliers mediante la distancia de Mahalanobis para conocer el comportamiento de tu banco de datos.

```
x <- scale(datosfinal_numeric)
rownames(x) <- paises
ma <- mahalanobis(x, apply(x, 2, mean), cov(x))
k <- dim(x)[2]
Lim <- k + 3 * sqrt(k * 2)
plot(ma, pch = 20, ylim = c(0, max(ma, Lim, na.rm = TRUE)))
text(ma, rownames(x), pos = 2)
abline(h = Lim)
title("Distancia de Mahalanobis")
```



El análisis de outliers según la distancia de Mahalanobis no deja ver ninguna observación anómala.

3. Realiza un análisis de agrupamiento jerárquico probando los algoritmos "ward.D2","single","complete" y "average" y comprueba:

```
distancias <- dist(scale(datosfinal_numeric))
clust_ward <- hclust(distancias, method="ward.D2",)
cor(distancias, cophenetic(clust_ward))
```

```
## [1] 0.8149534
```

```
clust_single <- hclust(distancias, method="single")
cor(distancias, cophenetic(clust_single))
```

```
## [1] 0.7874055
```

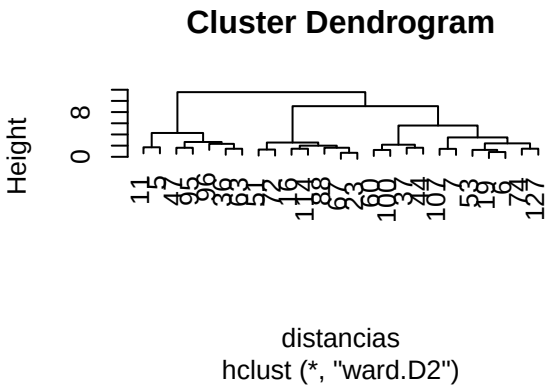
```
clust_complete <- hclust(distancias, method="complete")
cor(distancias, cophenetic(clust_complete))
```

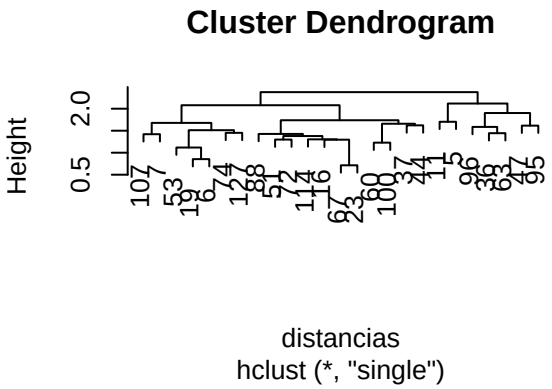
```
## [1] 0.8180485
```

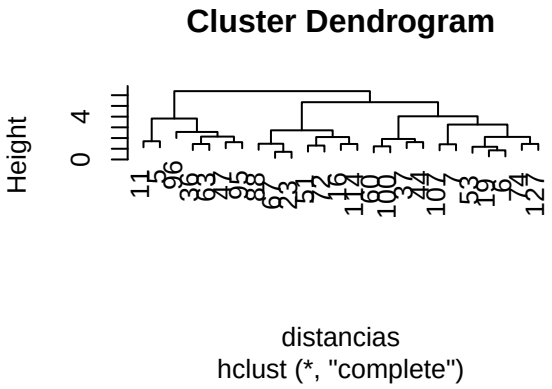
```
clust_average <- hclust(distancias, method="average")
cor(distancias, cophenetic(clust_average))
```

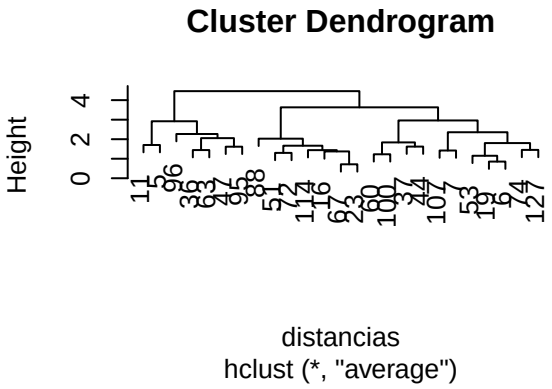
```
## [1] 0.8255034
```

```
par(mfrow=c(2,2))
plot(clust_ward)
plot(clust_single)
plot(clust_complete)
plot(clust_average)
```









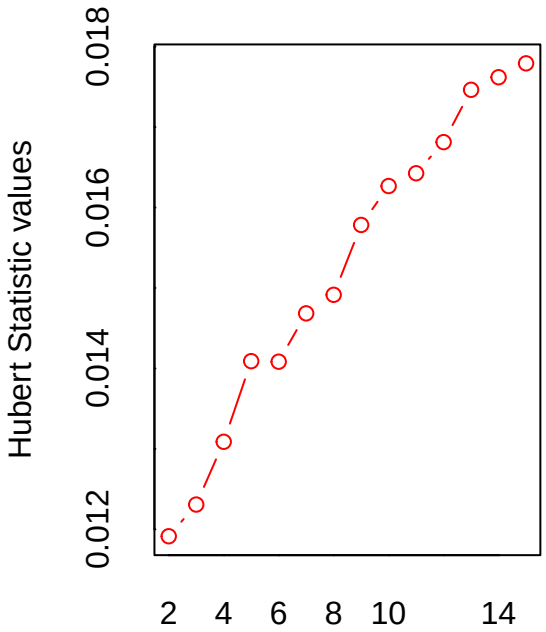
- ¿Detectas cierta estabilidad en los paises que se unen mediante los 4 dendogramas obtenidos? Los metodos "ward.D2", "complete" y "average" dan como resultado dos grandes grupos con una composición mas o menos similar. Además tanto "complete" como "average" ofrecen la agrupación más similar.

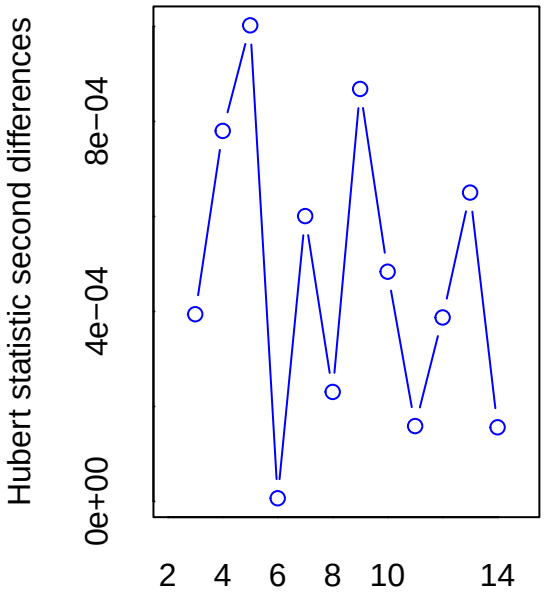
- ¿Qué algoritmo obtiene una mayor correlación cofenética? El método de "average" es el que posee una mayor correlación cofenética.

4. A partir del algoritmo con mejor comportamiento en el apartado anterior, utiliza la función Nbclust con ese método para seleccionar en base a todos los índices que calcula dicha función el mejor número de clusters para este banco de datos (seleccionando el que proporcionen como óptimo el mayor número de índices).

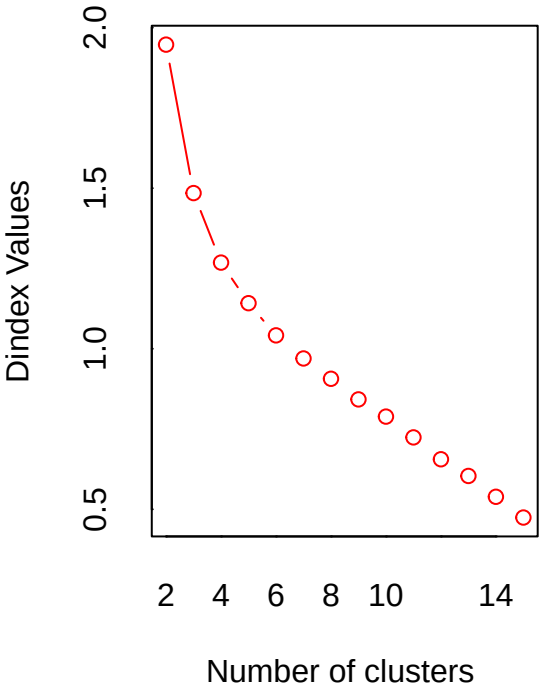
```
library(NbClust)
nbclust.complete <- NbClust(data=scale(datosfinal_numeric),
                            diss=NULL,
                            distance="euclidean",
                            method="average")
```

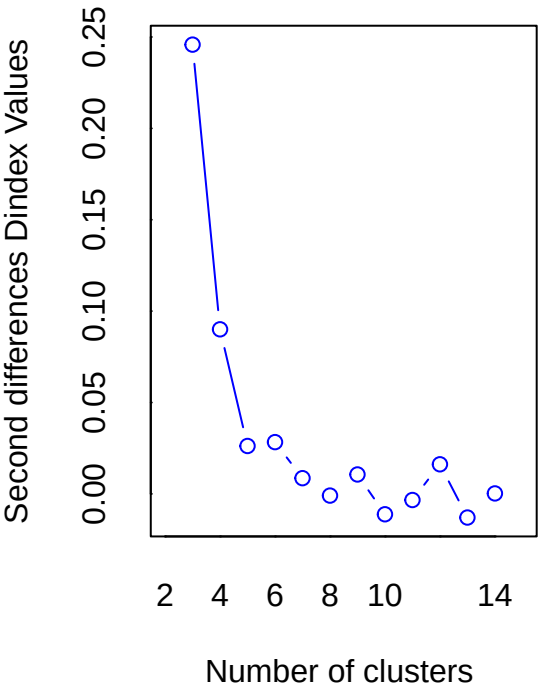
```
## Warning in pf(beale, pp, df2): NaNs produced

## Warning in pf(beale, pp, df2): NaNs produced

## Warning in pf(beale, pp, df2): NaNs produced

## Warning in pf(beale, pp, df2): NaNs produced
```






```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```






```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
```

```
##
## *****************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 11 as the best number of clusters
## * 6 proposed 15 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *****************************************************************
```

Se usarán tres clusters debido a que es el número optimo arrojado por la función NbClust.

5. En el apartado anterior has obtenido una propuesta del número óptimo de clusters y una propuesta de partición de los individuos en clusters (resultado $Best.partition de la función Nbclust)

```
partition <- nbclust.complete$Best.partition
countri_clusters <- list(clust1=as.numeric(names(partition[partition==1])),
                         clust2=as.numeric(names(partition[partition==2])),
                         clust3=as.numeric(names(partition[partition==3])))
centroides <- t(as.data.frame( lapply(countri_clusters, function(x, y){

  clust <- y[which(as.numeric(rownames(y)) %in% x),]

  apply(clust, 2, mean)
}, y = datosfinal_numeric)))
```

- Considera los clusters propuestos y calcula el centroide de cada uno de esos grupos propuestos (simplemente con la media para cada variable en cada grupo)

```
clust_kmeans <-kmeans(datosfinal_numeric,centroides,nstart = 100)
```

- Realiza un análisis de agrupamiento mediante el algoritmo de k-medias considerando como centroides iniciales los que has obtenido en el apartado anterior

```
nbclust.complete$Best.partition
```

- ¿Obtienes la misma composición de paises en los grupos o se produce alguna variación?

```
##  96  11  47   5  36  19   6  74 127  53  51  37  44 107  67  72  23  16 114  63
##   1   1   1   1   1   2   2   2   2   2   3   2   2   2   3   3   3   3   3   1
##  60  95  88   7 100
##   2   1   3   2   2
```

```
clust_kmeans$cluster
```

```
##  96  11  47   5  36  19   6  74 127  53  51  37  44 107  67  72  23  16 114  63
##   1   1   1   1   1   2   2   3   3   2   3   3   3   2   3   3   3   3   3   1
##  60  95  88   7 100
##   3   1   3   2   3
```

```
nbclust.complete$Best.partition==clust_kmeans$cluster
```

```
##    96    11    47     5    36    19     6    74   127    53    51    37    44
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
##   107    67    72    23    16   114    63    60    95    88     7   100
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
```

Como se puede observar se obtiene practicamente la misma composición de clusters.

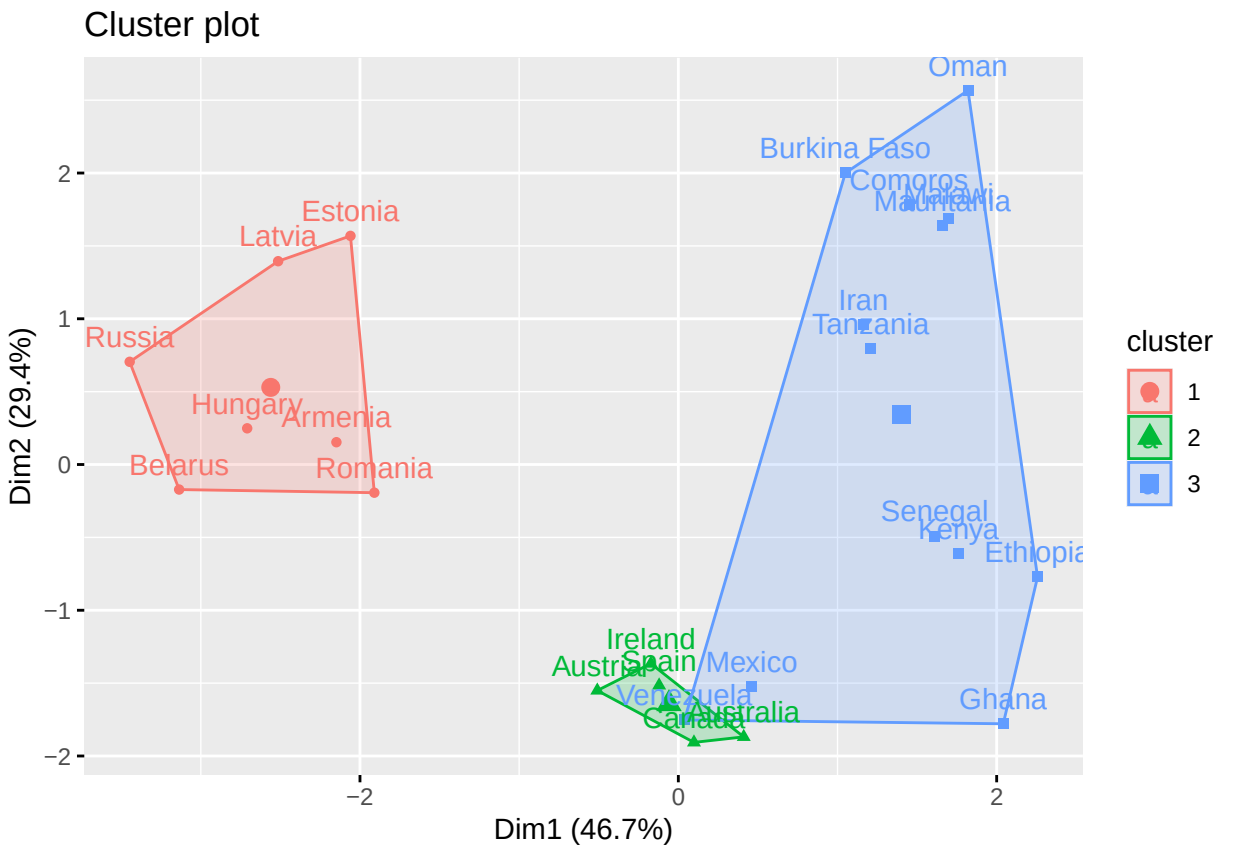
6. Partiendo del resultado de la composición de los clusters obtenidos en el apartado anterior, realiza un análisis exploratorio que te permita explicar la composición de cada grupo en función de las variables disponibles. Puedes apoyarte de descriptivos de las varibles en cada grupo o bien de un análisis de componentes principales que te ayude a explicar, en dos dimensiones, la composición de los grupos.

```
library(factoextra)
```

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

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

```
fviz_cluster(clust_kmeans, data=x)
```



En la representación del resultado del análisis de componentes principales del dataset, coloreado en función de los cluster, podemos observar como países pertenecientes al extinto Pacto de Varsovia o a la URSS se agrupan en un cluster, algo lógico puesto que la independencia de estos se produjo hace relativamente poco. También podemos observar como el cluster 2 solo posee países de la OCDE, aunque debe destacarse que tanto Hungría como México pertenecen a esta asociación de países. Por último encontramos el cluster 3 que aparte de ser el más disperso, presenta el conjunto de países más heterogéneo, pudiendo observarse regiones tan distantes como Centroamérica, México; Sudamérica, Venezuela; Sahel, Burkina Faso o Oriente Próximo, Omán y Irán. De todos los cluster este es el que presenta una mayor dificultad en la interpretación.