Casadio Sara-Homework 3

2022-10-19

Consider the following dataset with n = 4 observations and p = 5 variables, the first of which is categorical (for use with the simple matching distance), the second, third, and fourth are binary (Jaccard distance should be used), and the fourth is on a continuous scale. "NA" denotes missing values.

```
x1<-c("blue", 1, 1, 0, 12)
x2<-c("red", 0, 0, NA, NA)
x3<-c("red", 1, 0, NA, 17)
x4<-c("green", 1, 0, 0, 21)
dati<-t(cbind(x1,x2,x3,x4))  #observations in column
data<-as.matrix(dati)
```

What are the Gower (coefficient) dissimilarities between all pairs of observations? (a) Manually compute Gower dissimilarities based on distances for all variables separately, including variables 2-4 (Jaccard distance for a single variable, see slides).

```
d12.1<-1
d12.2<-1-0/1
d12.3<-1-0/1
d12.4<-NA
d12.5<-NA
d12.g<-(1*d12.1+1*d12.2+1*d12.3+0+0)/(1+1+1) #1

d13.1<-1
d13.2<-1-1/1
d13.3<-1-0/1
d13.4<-NA
d13.5<-(abs(12-17))/(21-12)
d13.g<-(1*d13.1+1*d13.2+1*d13.3+0+1*d13.5)/(1+1+1+1) #0.63888

d14.1<-1
d14.2<-1-1/1
d14.3<-1-0/1
d14.4<-0 #according on 0
d14.5<-abs(12-21)/(21-12)
d14.g<-(1*d14.1+1*d14.2+1*d14.3+0*d14.4+1*d14.5)/(1+1+1+1) #0.75

d23.1<-0
d23.2<-1-0/1
d23.3<-0 #according on 0
d23.4<-NA
d23.5<-NA
d23.g<-(1*d23.1+1*d23.2+0*d23.3+0+0)/(1+1) #0.5

d24.1<-1
```

```
d24.2<-1-0/1
d24.3<-0 #according on 0
d24.4<-NA
d24.5<-NA
d24.g<-(1*d24.1+1*d24.2+0*d24.3+0+0)/(1+1) #1

d34.1<-1
d34.2<-1-1/1
d34.3<-0 #according on 0
d34.4<-NA
d34.5<-(abs(17-21))/(21-12)
d34.g<-(1*d34.1+1*d34.2+0*d34.3+0+1*d34.5)/(1+1+1) #0.48148

vect1<-c(0,d12.g,d13.g,d14.g)
vect2<-c(d12.g,0,d23.g,d24.g)
vect3<-c(d13.g,d23.g,0,d34.g)
vect4<-c(d14.g,d24.g,d34.g,0)

gower.1<-cbind(vect1,vect2,vect3,vect4)
```

(b) Compute Gower dissimilarities treating variables 2-4 as a single group on which you compute a Jaccard dissmilarity (missing values mean that the Jaccard distance is only computed based on non-missing entries) that has weight 3 (because of 3 variables) in the final Gower dissimilarity. Does this give the same result as part (a)?

```
d12.234<-1-(0+0)/2
d12.new<-(1*d12.1+3*d12.234+0)/4 #1

d13.234<-1-(1+0)/2
d13.new<-(1*d13.1+3*d13.234+1*d13.5)/5 #0.6111

d14.234<-1-(1+0+0)/2
d14.new<-(1*d14.1+3*d14.234+1*d14.5)/5 #0.7

d23.234<-1-(0+0)/1
d23.new<-(1*d23.1+3*d23.234+0)/4 #0.75

d24.234<-1-(0+0)/1
d24.new<-(1*d24.1+3*d24.234+0)/4 #1

d34.234<-1-(1+0)/1
d34.new<-(1*d34.1+3*d34.234+1*d34.5)/5 #0.2888889

vect11<-c(0,d12.new,d13.new,d14.new)
vect22<-c(d12.new,0,d23.new,d24.new)
vect33<-c(d13.new,d23.new,0,d34.new)
vect44<-c(d14.new,d24.new,d34.new,0)

gower.2<-cbind(vect11,vect22,vect33,vect44)
```

(c) Compute the Gower dissimilarites using the daisy-function in R and check against the manual calculation in (a) and (b).

```r
library(cluster)
x1 <-c("blue", 1, 1, 0, 12)
x2 <-c("red", 0, 0, NA, NA)
x3 <-c("red", 1, 0, NA, 17)
x4 <-c("green", 1, 0, 0, 21)
list<-cbind(x1,x2,x3,x4)

data1<-data.frame(t(list))
data1[,1]<-as.factor(data1[,1])
data1[,2]<-as.factor(data1[,2])
data1[,3]<-as.factor(data1[,3])
data1[,4]<-as.factor(data1[,4])
data1[,5]<-as.numeric(data1[,5])

daisy(data1,metric="gower", type=list(asymm=c(2,3,4)))
```

```
## Warning in daisy(data1, metric = "gower", type = list(asymm = c(2, 3, 4))):
## almeno una variabile binaria non ha 2 livelli differenti.
```

```
## Dissimilarities :
##           x1        x2        x3
## x2 1.0000000
## x3 0.6388889 0.5000000
## x4 0.7500000 1.0000000 0.4814815
##
## Metric :  mixed ;  Types = N, A, A, A, I
## Number of objects : 4
```

In this last matrix I work with type=list(asymm=c(2,3,4)) since I want to show that I deal with these variables with jaccard distance (since they are binary). This matrix computed with daisy function is equal with the first matrix of the point a manually computed. The second matrix is different from the others because daisy functions can not consider the aggregation of the three variables with weight equal to 3, it works by dividing every variables.

(2) Give counterexamples to show that the correlation dissimilarity and the Gower coefficient do not fulfill the triangle inequality, i.e., in each case present three observations of which you show that they violate the triangle inequality.

```r
#correlation dissimilarity
x<-c(15,17,9)
y<-c(30,11,3)
z<-c(5,25,10)

dato<-cbind(x,y,z)
corrmat<-cor(dato)

corr.dist<-0.5-corrmat/2
corr.dist<-as.dist(corr.dist)

d.xy<-corr.dist[1]
d.xz<-corr.dist[2]
d.yz<-corr.dist[3]
```

```
d.xy+d.yz>=d.xz #TRUE
```

```
## [1] TRUE
```

```
d.yz+d.xz>=d.xy #TRUE
```

```
## [1] TRUE
```

```
d.xy+d.xz>=d.yz #FALSE
```

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

The triangle inequality doesn't hold since it has to be verified for both three cases, but here d.xy+d.xz>=d.yz it doesn't work.

```
#gower coefficient
v1<-c(NA,0,0)
v2<-c(0,1,NA)
v3<-c(1,1,1)
v4<-c(1,0,0)

dati<-cbind(v1,v2,v3,v4)
data<-as.data.frame(dati)

gower<-daisy(data, metric=c("gower"))

d.xy<-gower[1]
d.xz<-gower[2]
d.yz<-gower[3]

d.xy+d.yz>=d.xz #TRUE
```

```
## [1] TRUE
```

```
d.xy+d.xz>=d.yz #TRUE
```

```
## [1] TRUE
```

```
d.yz+d.xz>=d.xy #FALSE
```

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

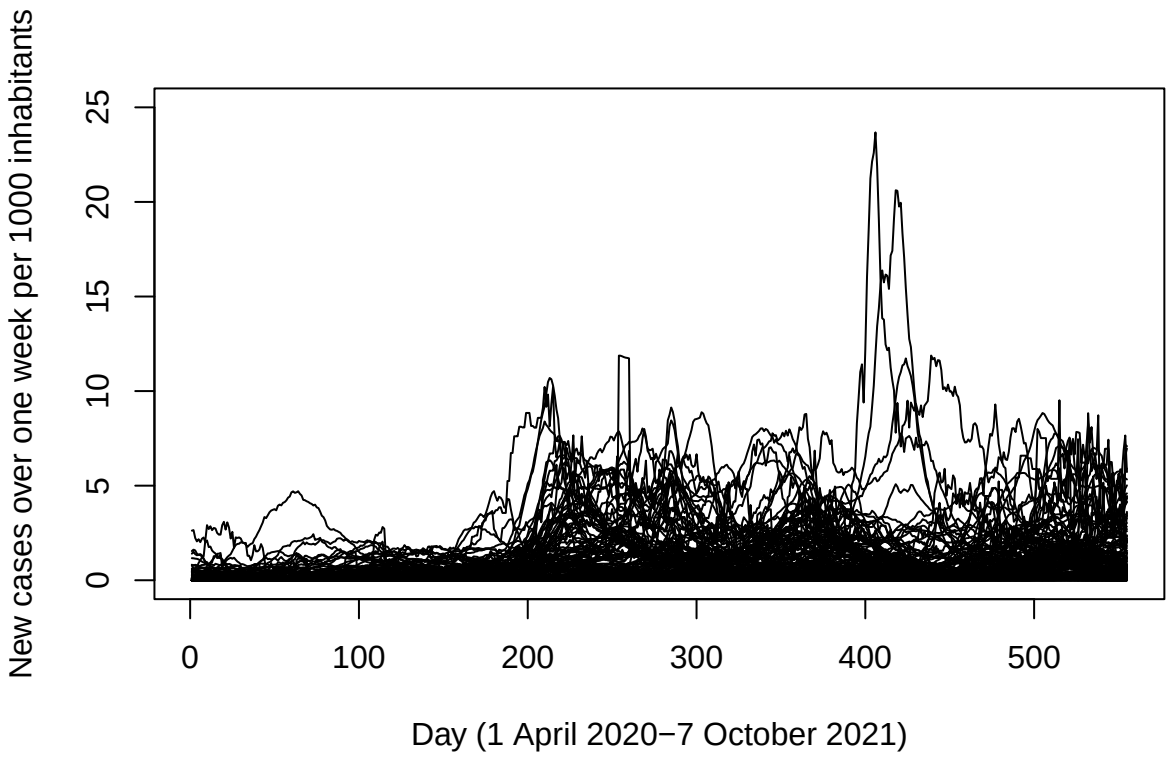
The triangle inequality doesn't hold since it has to be verified for both three cases, but here d.yz+d.xz>=d.x it doesn't work.

(4) (a)The task here is to cluster the countries in order to find groups of countries with similar developments. Try out one or more dissimilarity-based hierarchical clustering methods together with Euclidean and correlation dissimilarity. You may try to come up with further ideas for defining a dissimilarity for these data. Choose a number of clusters, try to understand and interpret the clusters as good as you can, using the information in the data, and built yourself an opinion which of the tried out clusterings is most appropriate, and how appropriate they are in general.

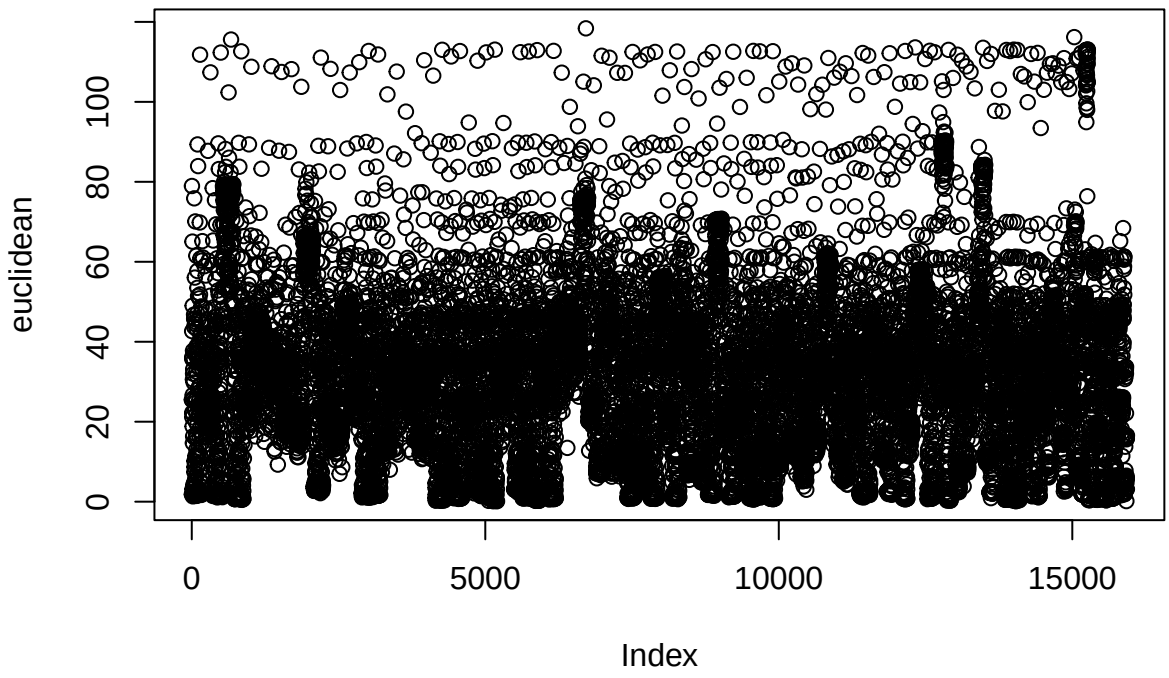
```
library(fpc)
library(pdfCluster)
```

```
## pdfCluster 1.0-3
```

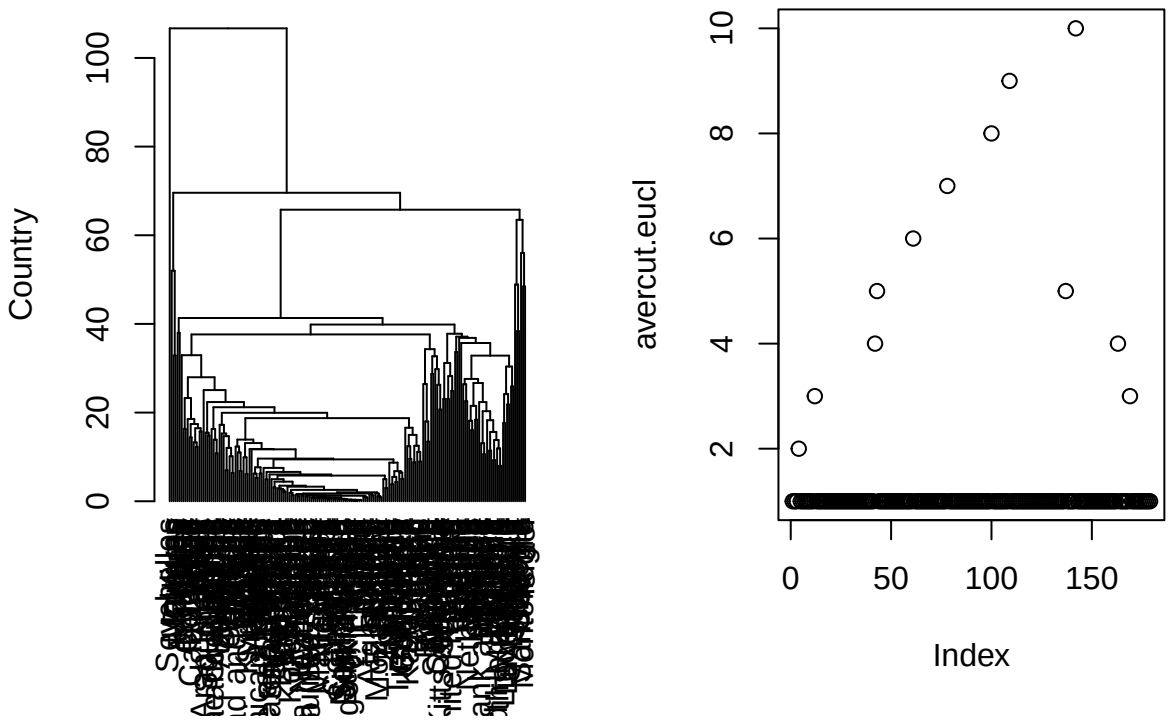
```
covid2021 <- read.table("C:/Users/Utente/OneDrive/Desktop/bigData/covid2021.dat", quote="\"", comment.ch
covid<-covid2021[,5:559]

plot(1:555,covid[1,],type="l", ylim=c(0,25),
     ylab="New cases over one week per 1000 inhabitants",
     xlab="Day (1 April 2020-7 October 2021)")
for(i in 2:179){
  points(1:555,covid[i,],type="l")}
```




```
#euclidean distance
euclidean<-dist(covid, method = "euclidean")
plot(euclidean)
```




```
par(mfrow=c(1,2))
aver.eucl<-hclust(euclidean, method = "average")
plot(as.dendrogram(aver.eucl), main="Average Linkage-Euclidean distance", xlab = "n.clusters", ylab="Co
avercut.eucl<- cutree(aver.eucl,k=10)
plot(avercut.eucl)
```




```
table(avercut.eucl)
```

```
## avercut.eucl
##   1   2   3   4   5   6   7   8   9  10
## 167   1   2   2   2   1   1   1   1   1
```

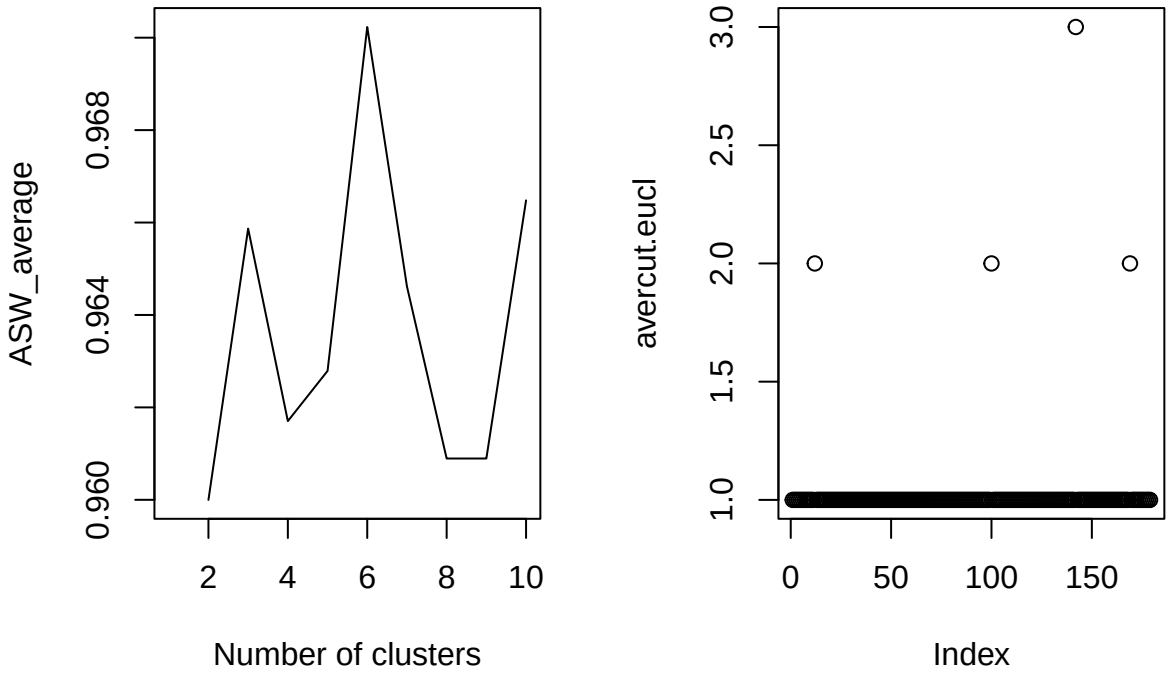
```
library(clusterSim)
```

```
## Caricamento del pacchetto richiesto: MASS
```

```
pasw.aver <- NA
pclusk.aver <- list()
psil.aver <- list()

for (k in 2:10){
  pclusk.aver[[k]] <- pam(avercut.eucl,k)
  psil.aver[[k]] <- silhouette(pclusk.aver[[k]],dist=avercut.eucl)
  pasw.aver[k] <- summary(psil.aver[[k]])$avg.width
}
plot(1:10,pasw.aver,type="l",xlab="Number of clusters",ylab="ASW_average")
pasw.aver
```

```
##  [1]        NA 0.9599987 0.9658680 0.9617035 0.9627898 0.9702314 0.9646182
##  [8] 0.9608939 0.9608939 0.9664804
```

```
avercut.eucl<- cutree(aver.eucl,k=3)
plot(avercut.eucl)
```




```
table(avercut.eucl)
```

```
## avercut.eucl
##   1   2   3
## 175   3   1
```

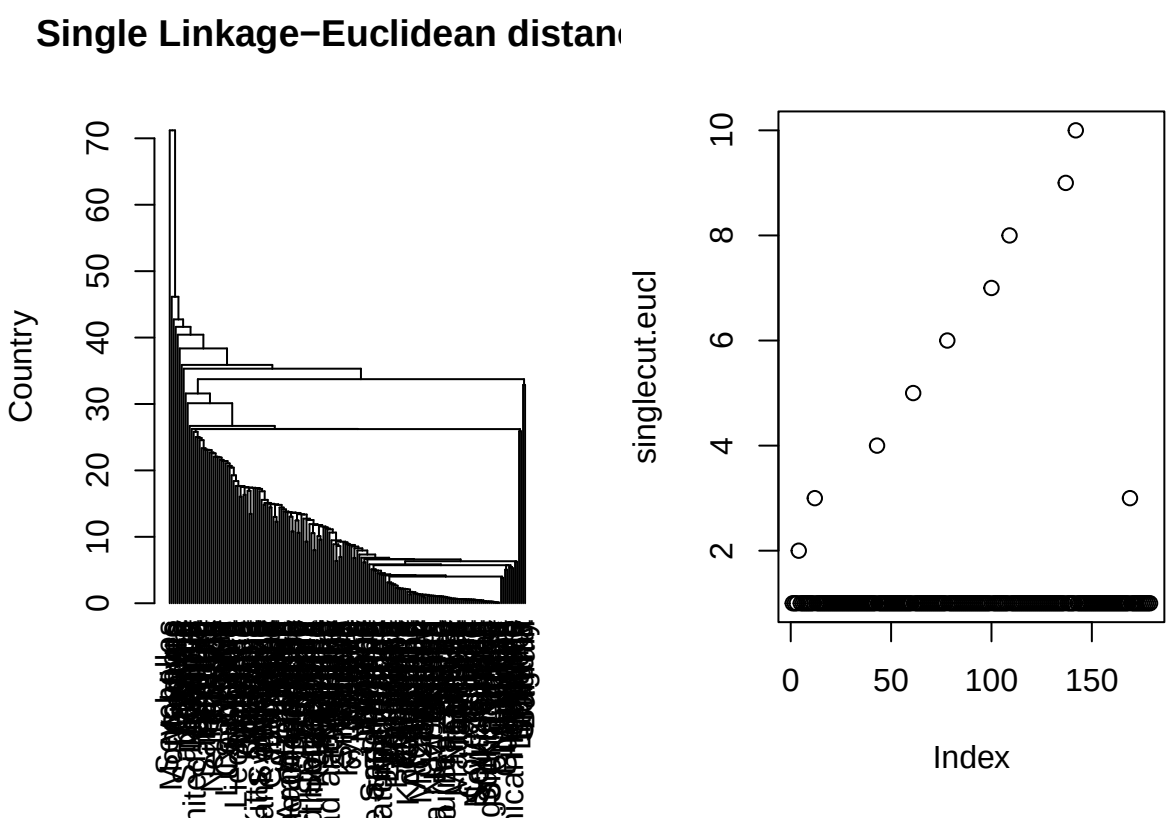
This is a compromise between the the single and the complete approaches, obtained as the average of the corresponding distances.

We obtain 167 elements in the first cluster, 2 in the clusters 3,4,5 and 1 in the others if we take k=10.

The best k is 3 following the silhouette index. [1] NA 0.9599987 0.9658680 0.9617035

If we compute again the clustering with k=3 we obtain the first cluster with 175 element, the second with 3 elements and the third with only one.

```
par(mfrow=c(1,2))
single.eucl<-hclust(euclidean, method = "single")
plot(as.dendrogram(single.eucl), main="Single Linkage-Euclidean distance", xlab = "n.clusters", ylab="C
singlecut.eucl<- cutree(single.eucl,k=10)
plot(singlecut.eucl)
```



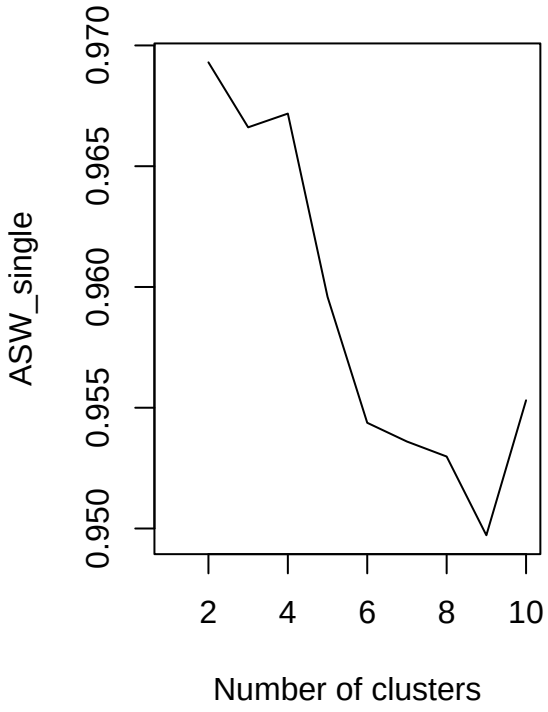

```
table(singlecut.eucl)
```

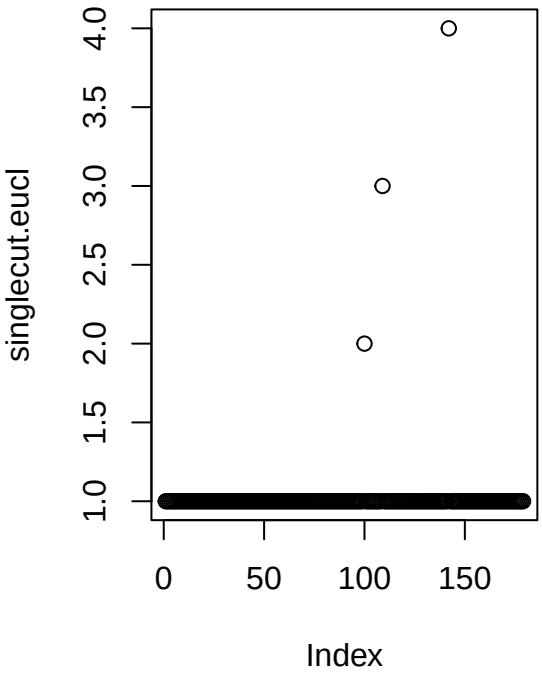
```
## singlecut.eucl
##   1   2   3   4   5   6   7   8   9  10
## 169   1   2   1   1   1   1   1   1   1
```

```
pasw.single <- NA
pclusk.single <- list()
psil.single<- list()

for (k in 2:10){
  pclusk.single[[k]] <- pam(singlecut.eucl,k)
  psil.single[[k]] <- silhouette(pclusk.single[[k]],dist=singlecut.eucl)
  pasw.single[k] <- summary(psil.single[[k]])$avg.width
}
plot(1:10,pasw.single,type="l",xlab="Number of clusters",ylab="ASW_single")
pasw.single
```

```
##  [1]        NA 0.9693001 0.9666093 0.9671788 0.9595903 0.9543762 0.9536002
##  [8] 0.9529795 0.9497207 0.9553073
```

```
singlecut.eucl<- cutree(single.eucl,k=4)
plot(singlecut.eucl)
```






```
table(singlecut.eucl)
```

```
## singlecut.eucl
##   1   2   3   4
## 176   1   1   1
```

The distance between two groups is defined as the smaller value of the distances between the items of the two groups.

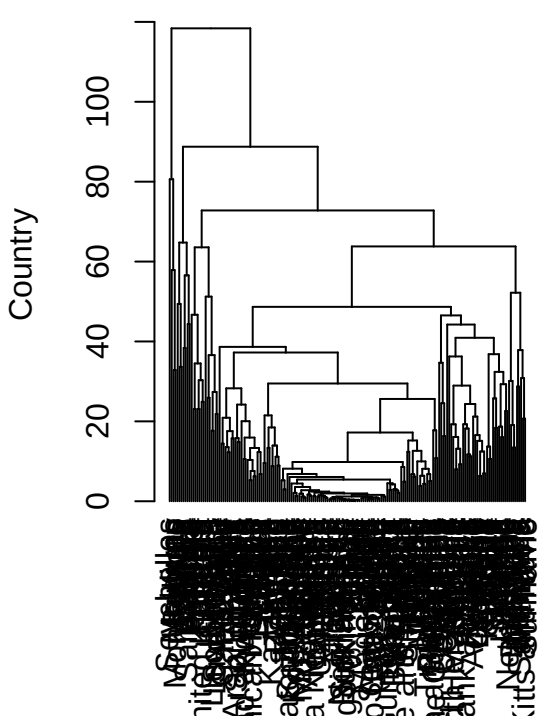
This method is affected by the chain effect: it tends to aggregate units in a big non homegeous cluster that has a spherical form and so we can see that here it aggregates all units a single cluster and if we increase the number of k it takes single-unit clusters. In fact if we consider k=10 we can see that 169 elements belong to cluster 1, while other clusters have only 1 unit.

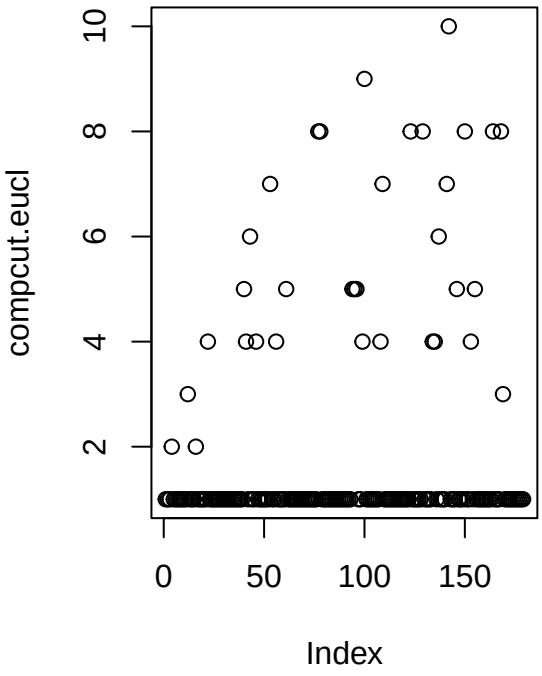
The best k is 4 following the silhouette index. [1] NA 0.9693001 0.9666093 0.9671788 0.9595903

With k=4 we obtain 176 units in the first group and 1 in the others.

```
par(mfrow=c(1,2))
comp.eucl<-hclust(euclidean, method = "complete")
plot(as.dendrogram(comp.eucl), main="Complete Linkage-Euclidean distance", xlab = "n.clusters", ylab="C
compcut.eucl<- cutree(comp.eucl,k=10)
plot(compcut.eucl)
```

Complete Linkage−Euclidean dista





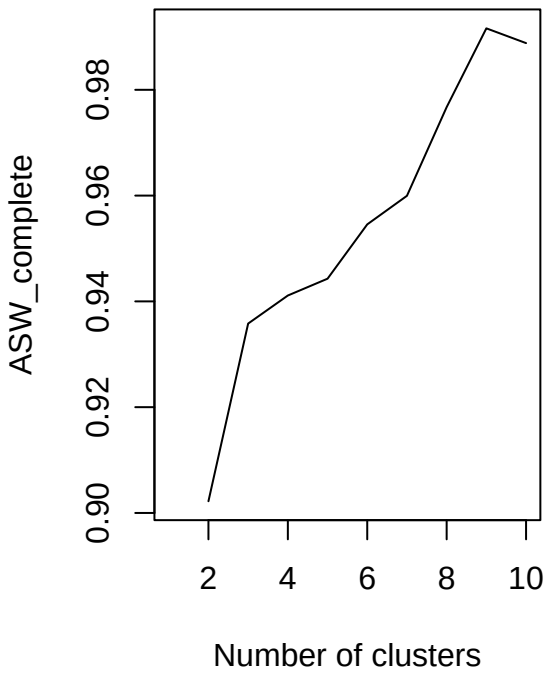

```
table(compcut.eucl)
```

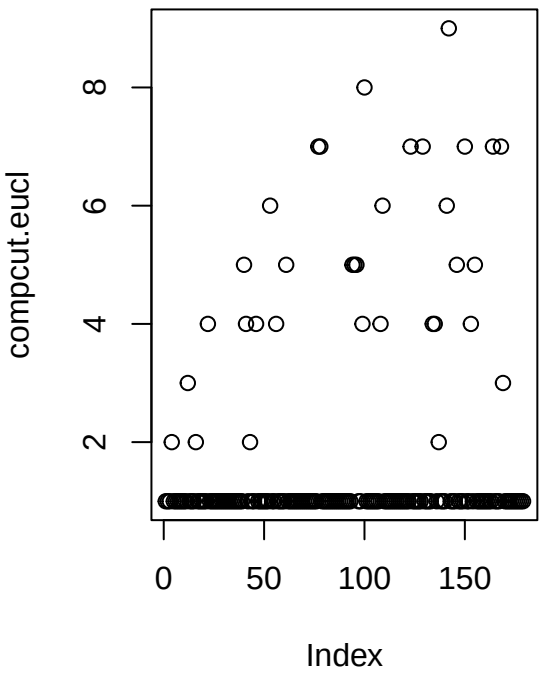
```
## compcut.eucl
##   1   2   3   4   5   6   7   8   9  10
## 145   2   2   9   7   2   3   7   1   1
```

```
pasw.comp <- NA
pclusk.comp <- list()
psil.comp<- list()

for (k in 2:10){
  pclusk.comp[[k]] <- pam(compcut.eucl,k)
  psil.comp[[k]] <- silhouette(pclusk.comp[[k]],dist=compcut.eucl)
  pasw.comp[k] <- summary(psil.comp[[k]])$avg.width
}
plot(1:10,pasw.comp,type="l",xlab="Number of clusters",ylab="ASW_complete")
pasw.comp
```

```
##  [1]        NA 0.9022156 0.9358060 0.9411120 0.9442928 0.9545542 0.9599628
##  [8] 0.9767225 0.9916201 0.9888268
```

```
compcut.eucl<- cutree(comp.eucl,k=9)
plot(compcut.eucl)
```






```
table(compcut.eucl)
```

```
## compcut.eucl
##   1   2   3   4   5   6   7   8   9
## 145   4   2   9   7   3   7   1   1
```

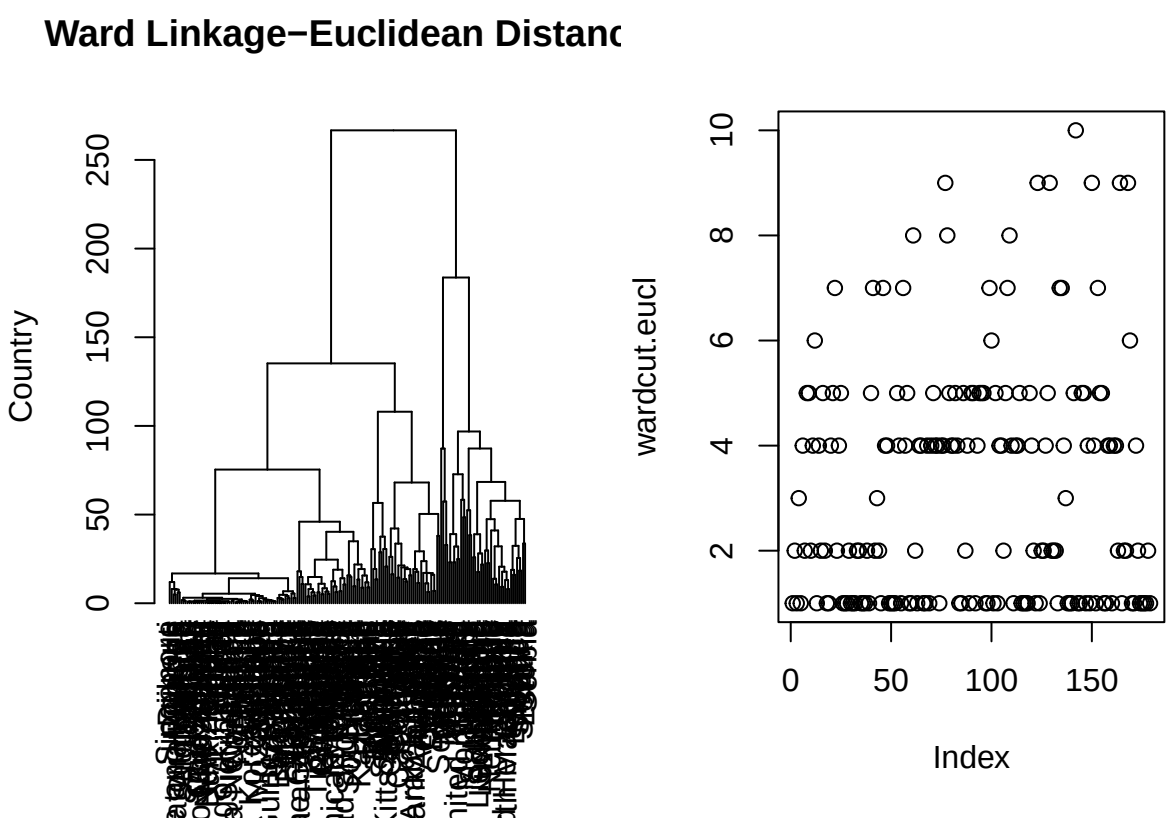
The distance between two groups is defined as the greatest distance value between the items of the two groups.

With k=10 we obtain 145 elements in the first cluster, 2 in the number 2,3 and 7, 9 in the fourth, 7 in the 5 fifth and eight, 2 in the sixth and 1 in the others.

The best k is 9 following the silhouette index [1] NA 0.9022156 0.9358060 0.9411120 0.9442928 0.9545542 0.9599628 0.9767225 0.9916201 0.9888268

Considering k=9 e have 145 elements in the first cluster, 4 units in the second, 2 in the third, 9 in the fourth, 7 in the fifth and seventh, 3 in the sixth and 1 in the other two.

```
par(mfrow=c(1,2))
ward.eucl<-hclust(euclidean, method = "ward.D2")
plot(as.dendrogram(ward.eucl), main="Ward Linkage-Euclidean Distance", xlab = "n.clusters", ylab="Counti
wardcut.eucl<- cutree(ward.eucl,k=10)
plot(wardcut.eucl)
```

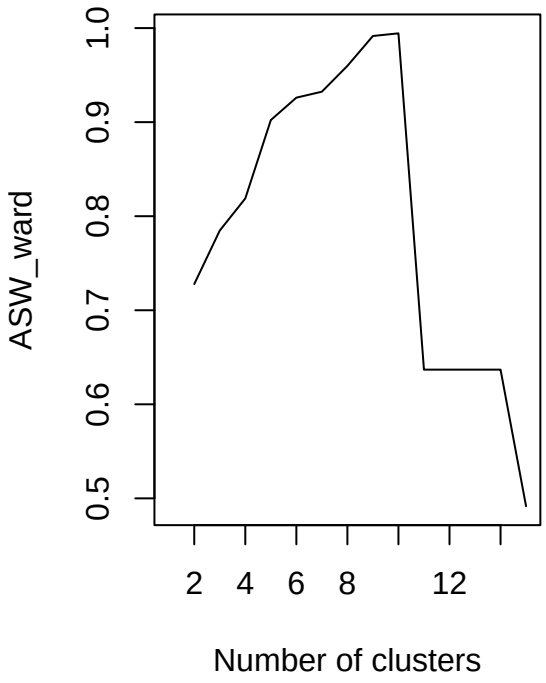



```
table(wardcut.eucl)
```

```
## wardcut.eucl
##  1  2  3  4  5  6  7  8  9 10
## 64 26  3 37 27  3  9  3  6  1
```

```
pasw.ward <- NA
pclusk.ward <- list()
psil.ward<- list()

for (k in 2:15){
  pclusk.ward[[k]] <- pam(wardcut.eucl,k)
  psil.ward[[k]] <- silhouette(pclusk.ward[[k]],dist=wardcut.eucl)
  pasw.ward[k] <- summary(psil.ward[[k]])$avg.width
}
plot(1:15,pasw.ward,type="l",xlab="Number of clusters",ylab="ASW_ward")
pasw.ward
```

```
##  [1]        NA 0.7278179 0.7847599 0.8188592 0.9022208 0.9260376 0.9323944
##  [8] 0.9598195 0.9916201 0.9944134 0.6368715 0.6368715 0.6368715 0.6368715
## [15] 0.4916201
```

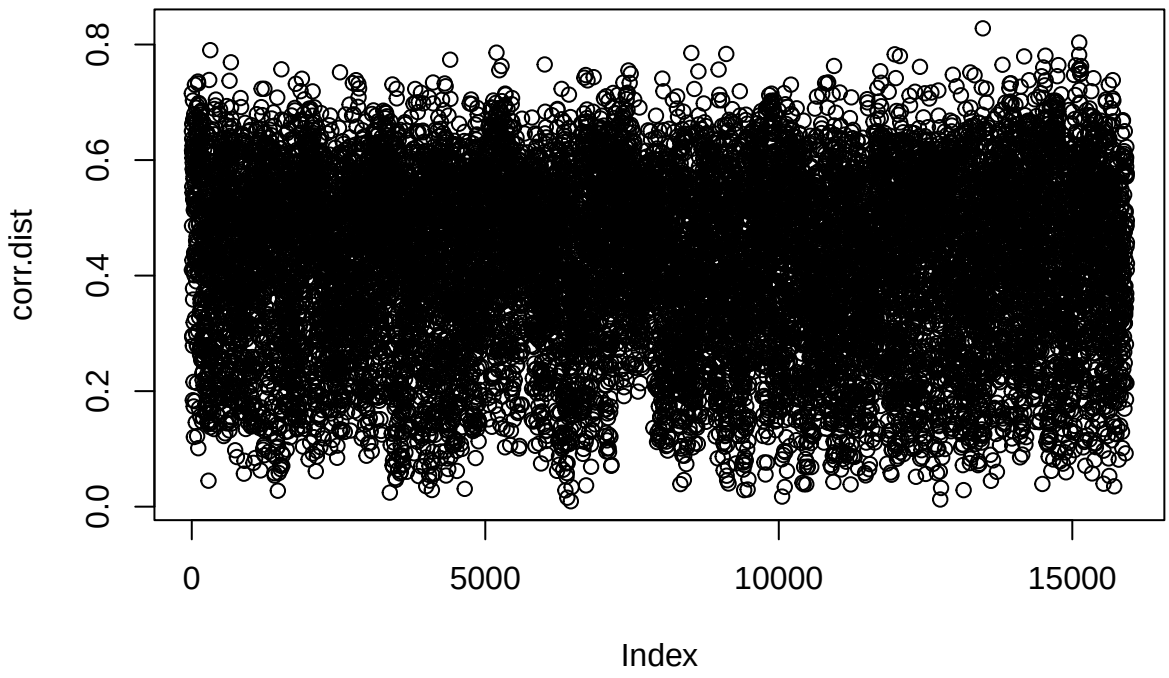


It joins groups that achieve the minimum growth of a given measure of heterogeneity within groups. Considering k=10 we can see that there are 64 elements in the first group, 26 in the second 3 in the third 37 in the fourth 27 in the fifth 3 in the sixth and in the eighth, 9 in the seventh, 6 in the ninth, 1 in the last.

The ward method may return a best partition with respect to the other method by dividing observation in more homogeneous cluster between and more heterogeneous clusters within.

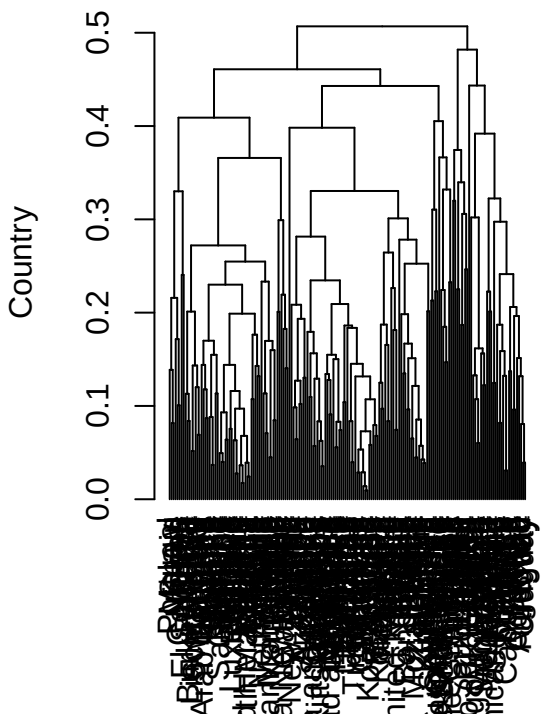
The best k is 10 following the silhouette index [1] NA 0.7278179 0.7847599 0.8188592 0.9022208 0.9260376 0.9323944 0.9598195 0.9916201 0.9944134 0.6368715

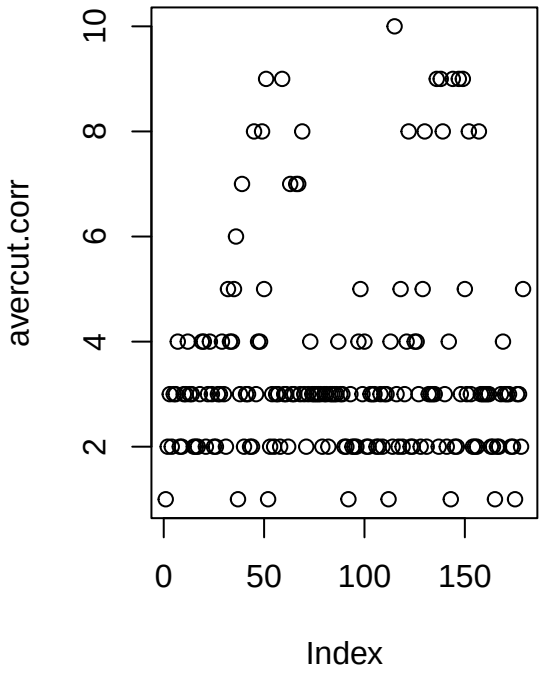
```
#correlation distance
corr.covid<-cor(t(covid))

corr.dist<-0.5-corr.covid/2
corr.dist<-as.dist(corr.dist)
plot(corr.dist)
```




```
par(mfrow=c(1,2))
aver.corr<-hclust(corr.dist, method = "average")
plot(as.dendrogram(aver.corr), main="Average Linkage-Correlation distance", xlab = "n.clusters", ylab="
avercut.corr<- cutree(aver.corr,k=10)
plot(avercut.corr)
```

Average Linkage−Correlation dista





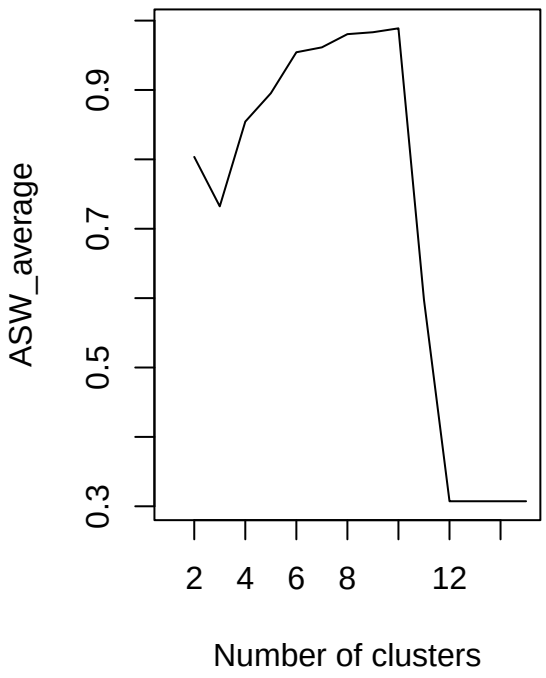

```
table(avercut.corr)
```

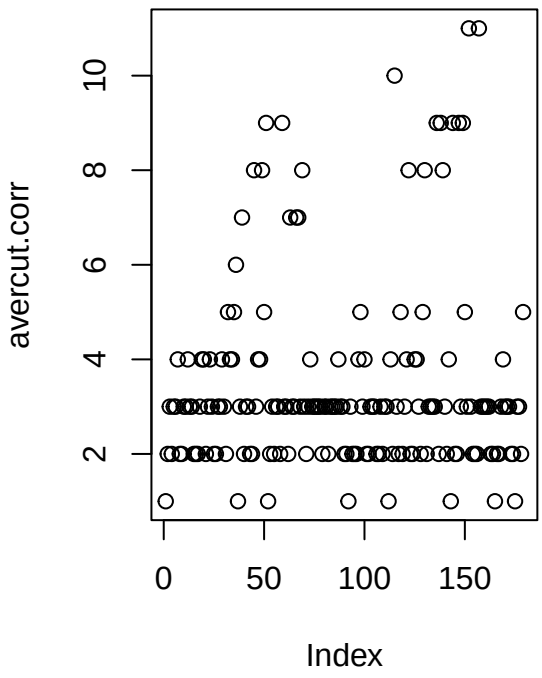
```
## avercut.corr
##  1  2  3  4  5  6  7  8  9 10
##  8 52 70 20  8  1  4  8  7  1
```

```
pasw.avercorr <- NA
pclusk.avercorr <- list()
psil.avercorr<- list()

for (k in 2:15){
  pclusk.avercorr[[k]] <- pam(avercut.corr,k)
  psil.avercorr[[k]] <- silhouette(pclusk.avercorr[[k]],dist=avercut.corr)
  pasw.avercorr[k] <- summary(psil.avercorr[[k]])$avg.width
}
plot(1:15,pasw.avercorr,type="l",xlab="Number of clusters",ylab="ASW_average")
pasw.avercorr
```

```
##  [1]        NA 0.8035332 0.7322565 0.8544105 0.8952810 0.9543663 0.9614272
##  [8] 0.9804469 0.9832402 0.9888268 0.5977654 0.3072626 0.3072626 0.3072626
## [15] 0.3072626
```

```
avercut.corr<- cutree(aver.corr,k=11)
plot(avercut.corr)
```






```
table(avercut.corr)
```

```
## avercut.corr
##  1  2  3  4  5  6  7  8  9 10 11
##  8 52 70 20  8  1  4  6  7  1  2
```

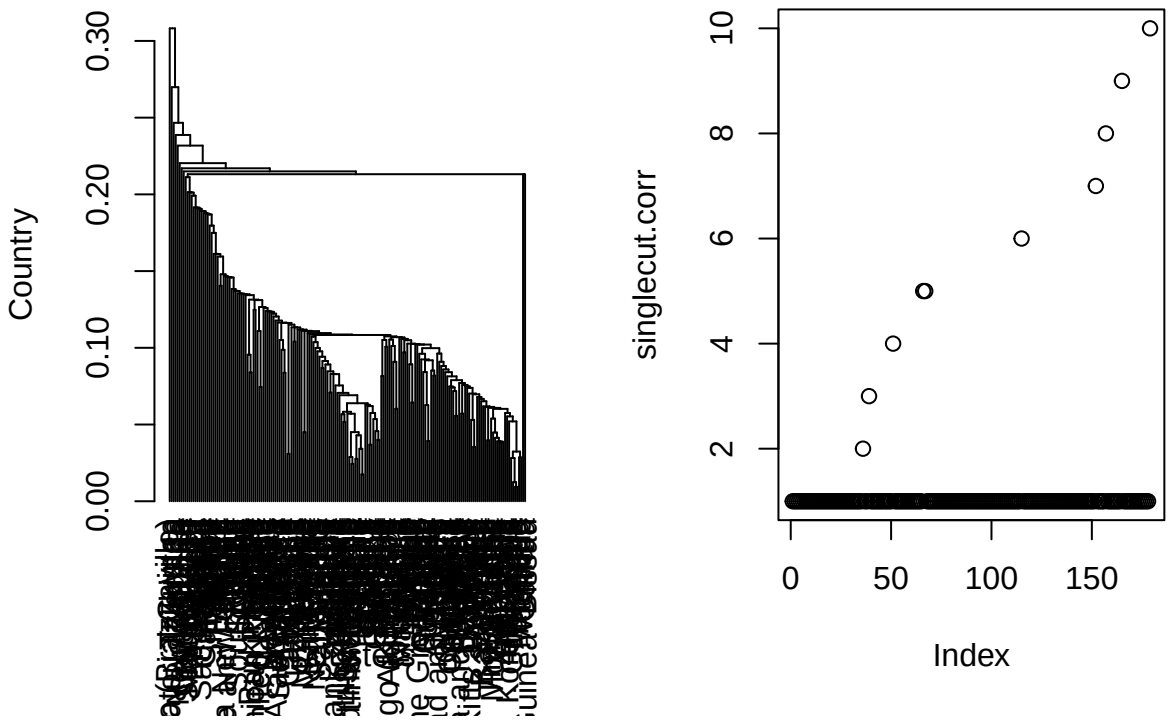
By considering k=10 we have: 8 elements in the first group, 52 in the second 70 in the third 20 in the fourth 8 in the fifth and in the eighth, 1 in the sixth and in the last, 4 in the seventh, 7 in the ninth.

The best k according to the silhouette index is 11. [1] NA 0.8035332 0.7322565 0.8544105 0.8952810 0.9543663 0.9614272 0.9804469 0.9832402 0.9888268 0.5977654

with k=11 8 elements in the first group, 52 in the second 70 in the third 20 in the fourth 8 in the fifth, 1 in the sixth and in the tenth, 4 in the seventh, 6 in the eighth, 7 in the ninth, 2 in the last.

```
par(mfrow=c(1,2))
single.corr<-hclust(corr.dist, method = "single")
plot(as.dendrogram(single.corr), main="Single Linkage-Correlation distance", xlab = "n.clusters", ylab=
singlecut.corr<- cutree(single.corr,k=10)
plot(singlecut.corr)
```

Single Linkage−Correlation distar



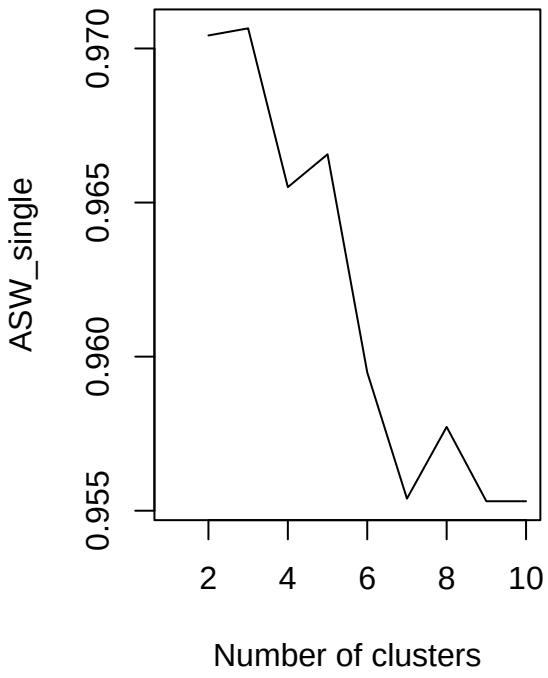

```
table(singlecut.corr)
```

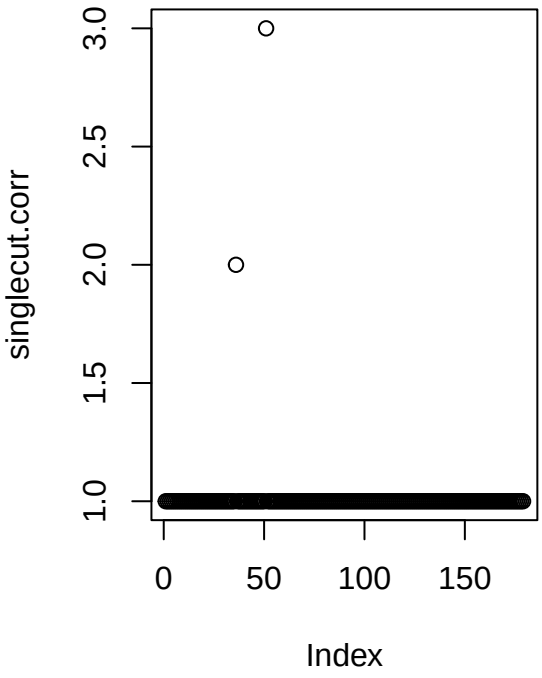
```
## singlecut.corr
##   1   2   3   4   5   6   7   8   9  10
## 169   1   1   1   2   1   1   1   1   1
```

```
pasw.singlecorr <- NA
pclusk.singlecorr <- list()
psil.singlecorr<- list()

for (k in 2:10){
  pclusk.singlecorr[[k]] <- pam(singlecut.corr,k)
  psil.singlecorr[[k]] <- silhouette(pclusk.singlecorr[[k]],dist=singlecut.corr)
  pasw.singlecorr[k] <- summary(psil.singlecorr[[k]])$avg.width
}
plot(1:10,pasw.singlecorr,type="l",xlab="Number of clusters",ylab="ASW_single")
pasw.singlecorr
```

```
##  [1]        NA 0.9704224 0.9706529 0.9654990 0.9665653 0.9594890 0.9553921
##  [8] 0.9577199 0.9553073 0.9553073
```

```
singlecut.corr<- cutree(single.corr,k=3)
plot(singlecut.corr)
```






```
table(singlecut.corr)
```

```
## singlecut.corr
##   1   2   3
## 177   1   1
```

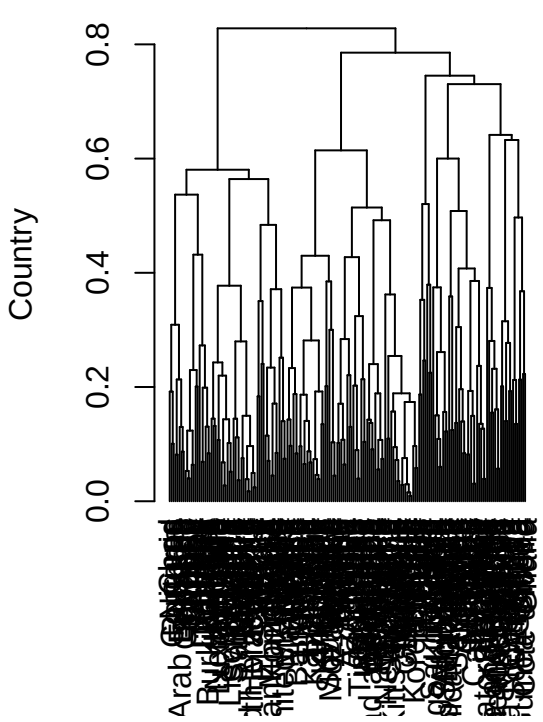
In this case like previous we have the first cluster with 169 units and all the other with only 1, cause it eists the chain effect.

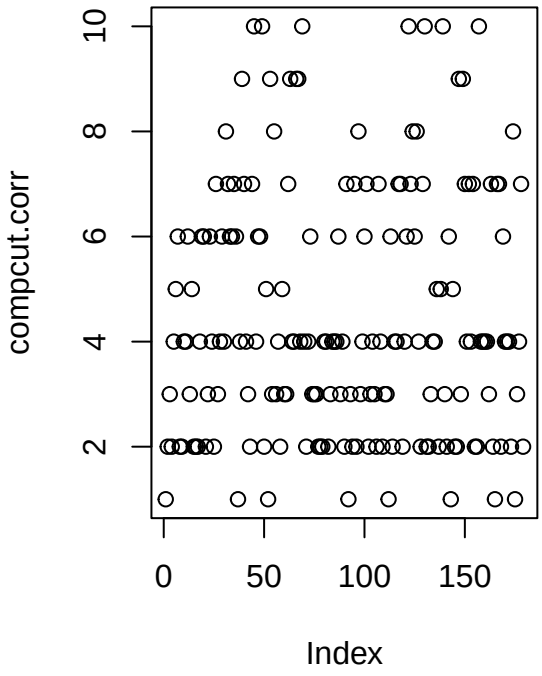
The best k according to the silhouette index is 3. [1] NA 0.9704224 0.9706529 0.9654990

With k=3 we obtain the first cluster with 177 units, and others always with 1 element.

```
par(mfrow=c(1,2))
comp.corr<-hclust(corr.dist, method = "complete")
plot(as.dendrogram(comp.corr), main="Complete Linkage-Correlation distance", xlab = "n.clusters", ylab=
compcut.corr<- cutree(comp.corr,k=10)
plot(compcut.corr)
```

Complete Linkage−Correlation dista



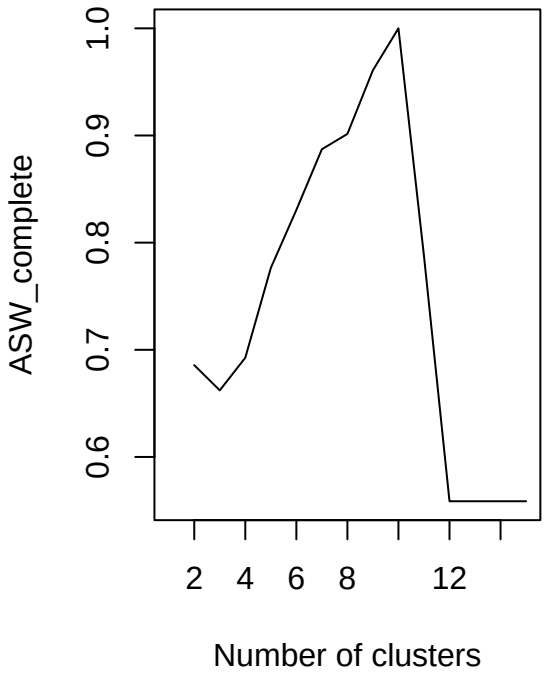



```
table(compcut.corr)
```

```
## compcut.corr
##  1  2  3  4  5  6  7  8  9 10
##  8 38 25 41  7 19 21  6  7  7
```

```
pasw.compcorr <- NA
pclusk.compcorr <- list()
psil.compcorr<- list()

for (k in 2:15){
  pclusk.compcorr[[k]] <- pam(compcut.corr,k)
  psil.compcorr[[k]] <- silhouette(pclusk.compcorr[[k]],dist=compcut.corr)
  pasw.compcorr[k] <- summary(psil.compcorr[[k]])$avg.width
}
plot(1:15,pasw.compcorr,type="l",xlab="Number of clusters",ylab="ASW_complete")
pasw.compcorr
```

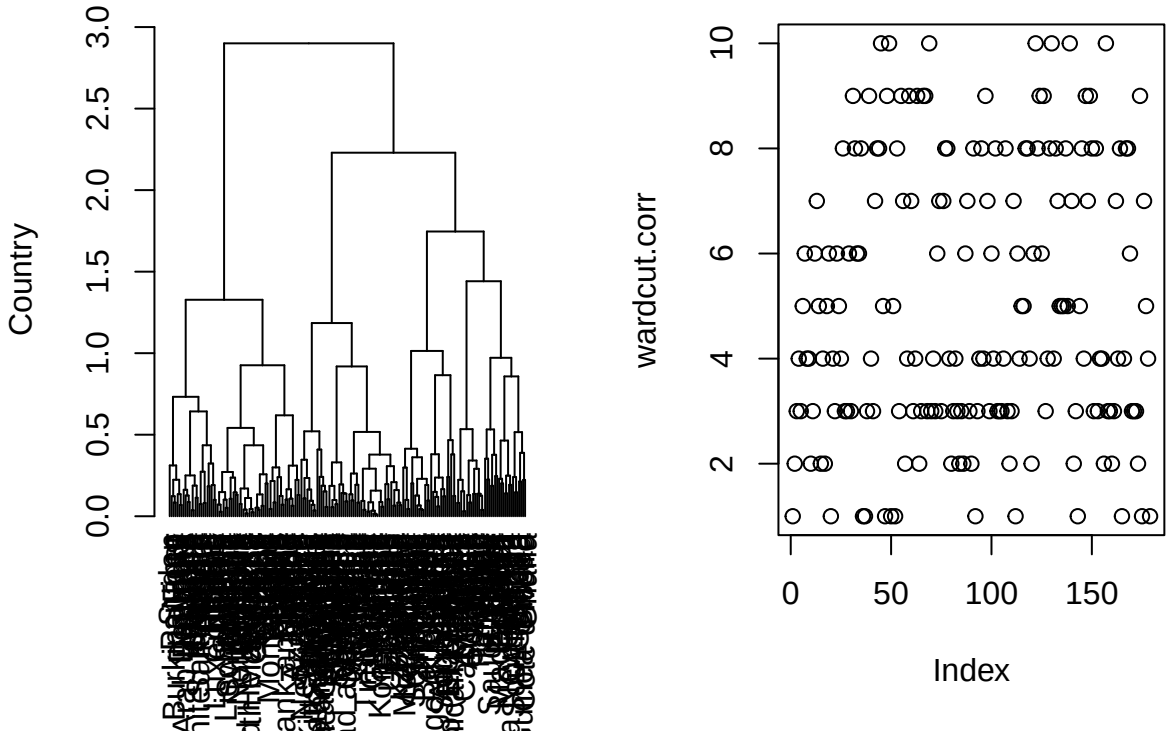
```
##  [1]        NA 0.6856753 0.6621425 0.6926216 0.7764214 0.8305468 0.8871576
##  [8] 0.9014525 0.9608939 1.0000000 0.7877095 0.5586592 0.5586592 0.5586592
## [15] 0.5586592
```



With k=10 we obtain 8 elements in the first cluster, 38 in the second, 25 in the third, 41 in the fourth, 7 in the 5 fifth, ninth and tenth, 19 in the sixth, 21 in the seventh, 6 in the eighth.

The best k according to the silhouette index is 10. [1] NA 0.6856753 0.6621425 0.6926216 0.7764214 0.8305468 0.8871576 0.9014525 0.9608939 1.0000000 0.7877095

```
par(mfrow=c(1,2))
ward.corr<-hclust(corr.dist, method = "ward.D2")
plot(as.dendrogram(ward.corr), main="Ward Linkage-Correlation Distance", xlab = "n.clusters", ylab="Cou
wardcut.corr<- cutree(ward.corr,k=10)
plot(wardcut.corr)
```

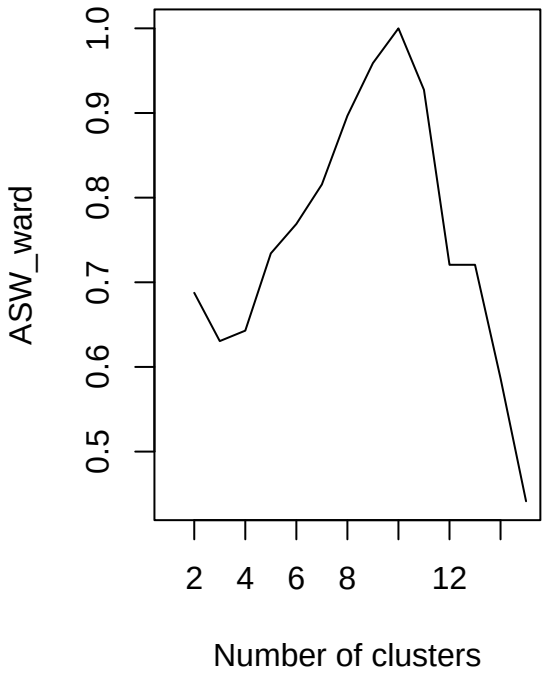



```
table(wardcut.corr)
```

```
## wardcut.corr
##  1  2  3  4  5  6  7  8  9 10
## 13 16 37 26 14 14 14 24 14  7
```

```
pasw.wardcorr <- NA
pclusk.wardcorr <- list()
psil.wardcorr<- list()

for (k in 2:15){
  pclusk.wardcorr[[k]] <- pam(wardcut.corr,k)
  psil.wardcorr[[k]] <- silhouette(pclusk.wardcorr[[k]],dist=wardcut.corr)
  pasw.wardcorr[k] <- summary(psil.wardcorr[[k]])$avg.width
}
plot(1:15,pasw.wardcorr,type="l",xlab="Number of clusters",ylab="ASW_ward")
pasw.wardcorr
```

```
##  [1]        NA 0.6875494 0.6305342 0.6428968 0.7341406 0.7689058 0.8155789
##  [8] 0.8966879 0.9589385 1.0000000 0.9273743 0.7206704 0.7206704 0.5865922
## [15] 0.4413408
```



Considering k=10 we can see that there are 13 elements in the first group, 16 in the second 37 in the third 26 in the fourth 14 in the fifth, sixth, seventh, ninth, 24 in the eighth, 7 in the last.

The best k according to the silhouette index is 10. [1] NA 0.6875494 0.6305342 0.6428968 0.7341406 0.7689058 0.8155789 0.8966879 0.9589385 1.0000000 0.9273743

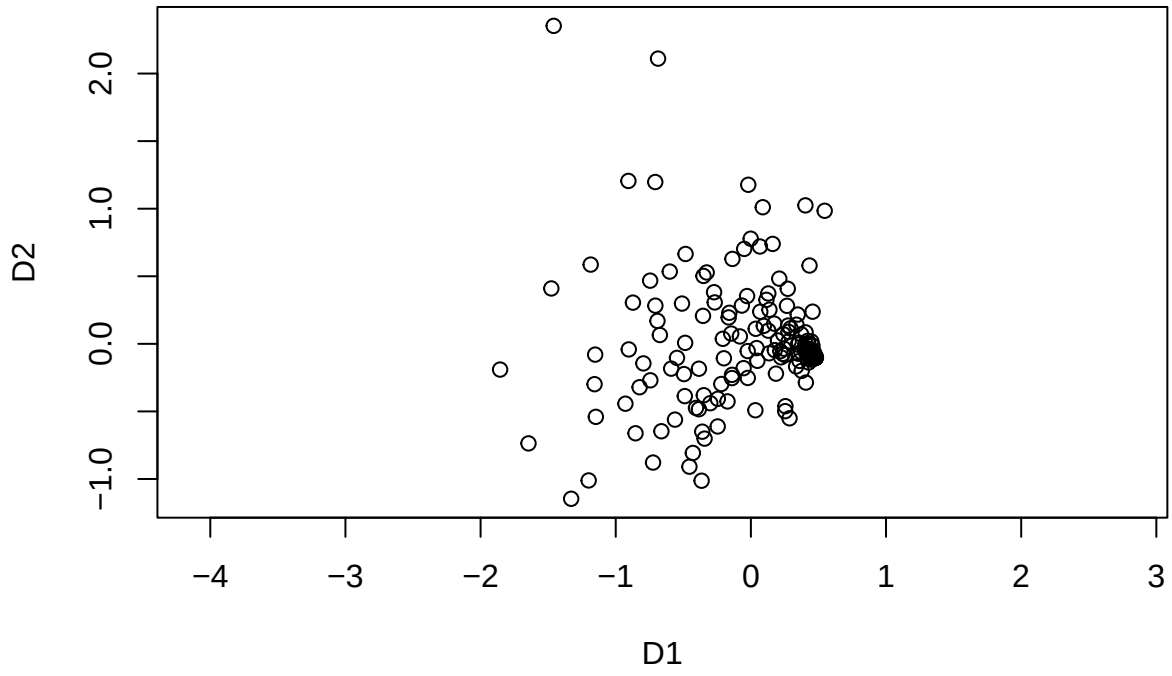
Generally we can say that the method applied on the correlation distances matrix returns best numbers of cluster higher that the results of the euclidean distance matrix, if we compute it with the silhouette index. The only one case in which results are similar is the Single Linkage one.

(b)Also visualise the data set using multidimensional scaling, coloring the observations according to the clusters.

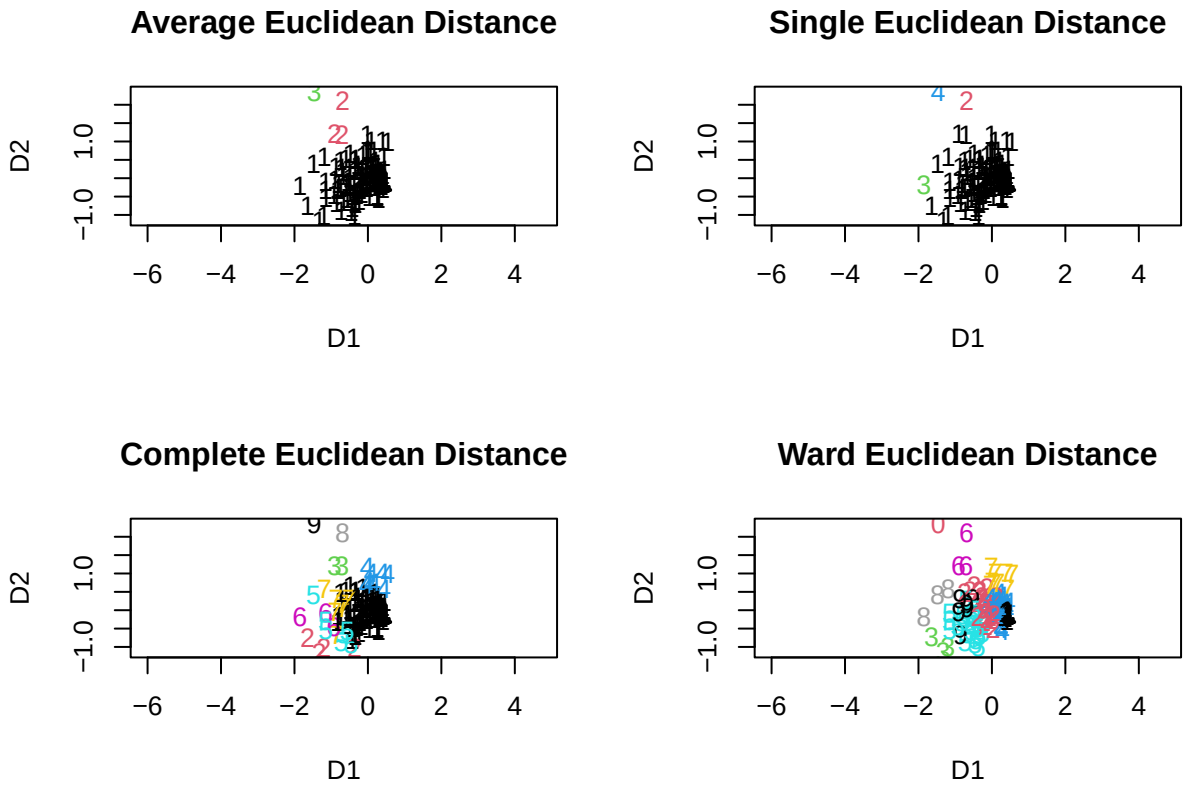
```
library(smacof)
```

```
## Caricamento del pacchetto richiesto: plotrix

## Caricamento del pacchetto richiesto: colorspace

## Caricamento del pacchetto richiesto: e1071

##
## Caricamento pacchetto: 'smacof'

## Il seguente oggetto è mascherato da 'package:base':
##
##     transform
```

```
multidim.eucl<-mds(euclidean,ndim=2)
plot(multidim.eucl$conf,type="p", asp=1)
```

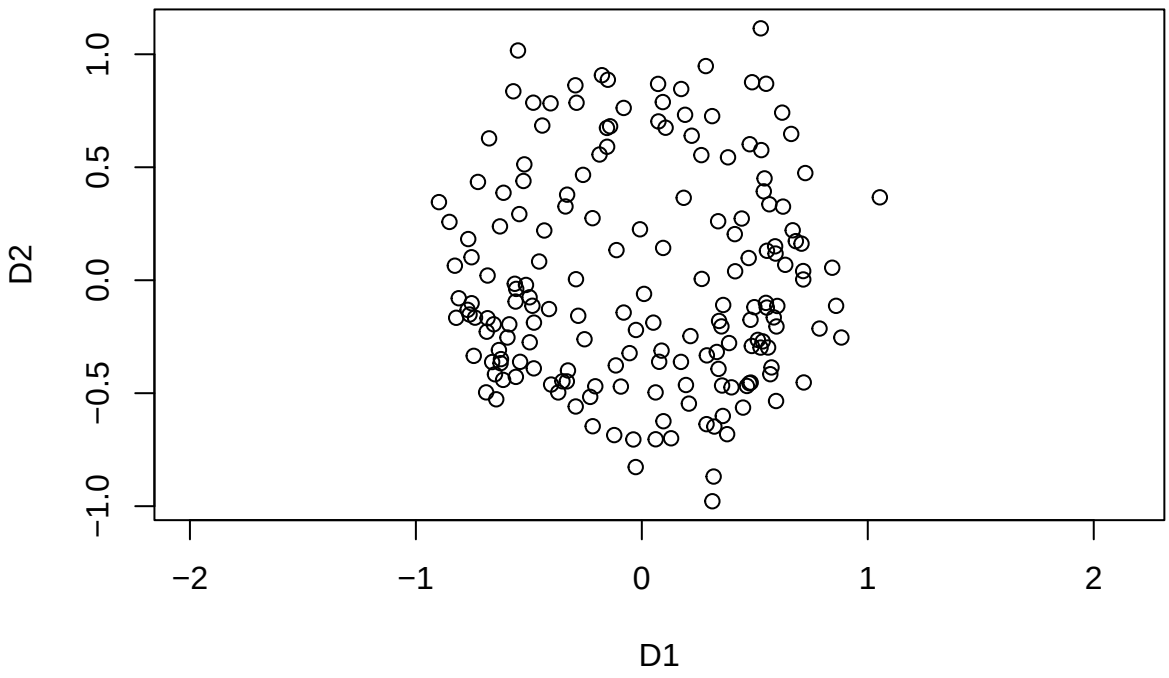



```
avercut.eucl<- cutree(aver.eucl,k=3)
singlecut.eucl<- cutree(single.eucl,k=4)
compcut.eucl<- cutree(comp.eucl,k=9)
wardcut.eucl<- cutree(ward.eucl,k=10)

par(mfrow=c(2,2))

plot(multidim.eucl$conf,col=avercut.eucl,
     pch=clusym[avercut.eucl],
     asp=1, main= "Average Euclidean Distance")

plot(multidim.eucl$conf,col=singlecut.eucl,
     pch=clusym[singlecut.eucl],
     asp=1, main= "Single Euclidean Distance")

plot(multidim.eucl$conf,col=compcut.eucl,
     pch=clusym[compcut.eucl],
     asp=1, main= "Complete Euclidean Distance")

plot(multidim.eucl$conf,col=wardcut.eucl,
     pch=clusym[wardcut.eucl],
     asp=1, main= "Ward Euclidean Distance")
```

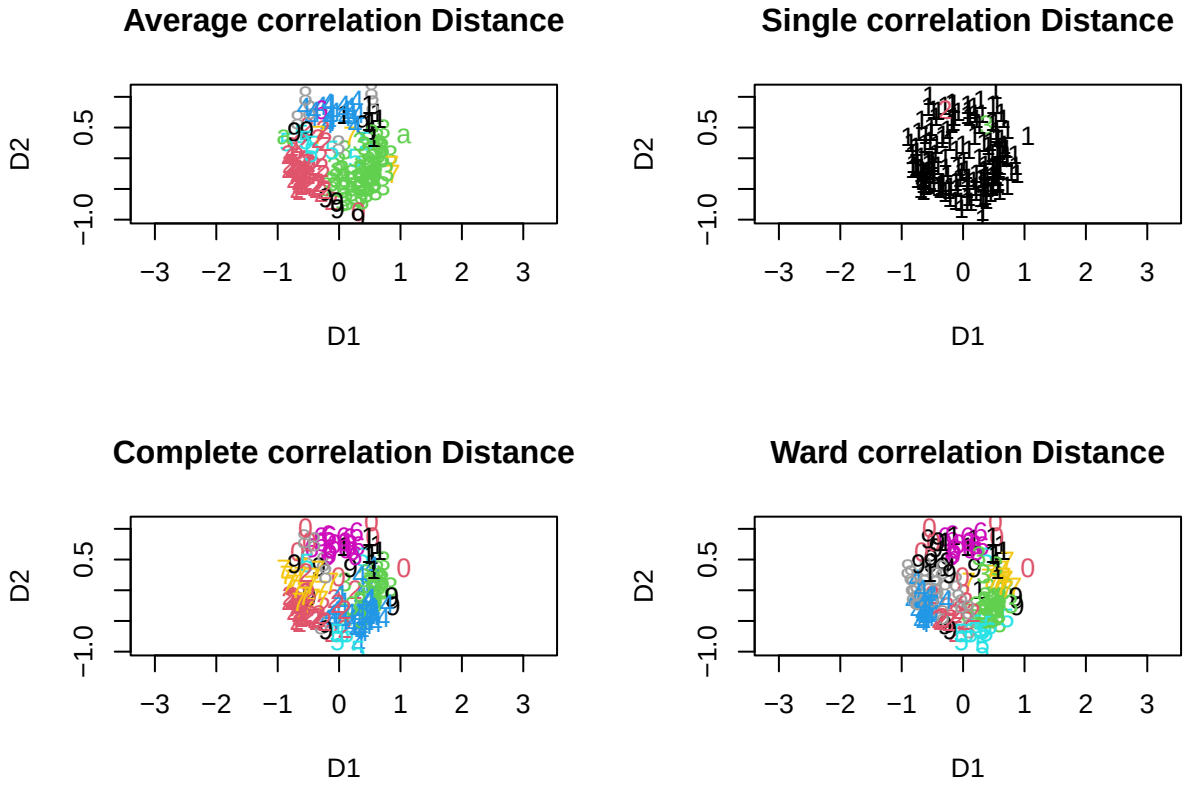


The best partition is the one obtained with the Ward Linkage methods since it provides more homogeneous clusters within and heterogeneous between. The worse is provided by the Single Linkage Method but it is not so good also the Average method because they aggregate the biggest part of units in the same cluster. The plot about Complete Method Distance is a little bit confused and difficult to interpret.

```
multidim.corr<-mds(corr.dist ,ndim=2)
plot(multidim.corr$conf,type="p", asp=1)
```




```
avercut.corr<- cutree(aver.corr,k=11)
singlecut.corr<- cutree(single.corr,k=3)
compcut.corr<- cutree(comp.corr,k=10)
wardcut.corr<- cutree(ward.corr,k=10)
par(mfrow=c(2,2))

plot(multidim.corr$conf,col=avercut.corr,
     pch=clusym[avercut.corr],
     asp=1, main= "Average correlation Distance")

plot(multidim.corr$conf,col=singlecut.corr,
     pch=clusym[singlecut.corr],
     asp=1, main= "Single correlation Distance")

plot(multidim.corr$conf,col=compcut.corr,
     pch=clusym[compcut.corr],
     asp=1, main= "Complete correlation Distance")

plot(multidim.corr$conf,col=wardcut.corr,
     pch=clusym[wardcut.corr],
     asp=1, main= "Ward correlation Distance")
```



The plots about correlation distances matrix are so confused and full of partitions since, as I have said before, with these data a bigger number of clusters is required in order to obtain the best partition. In the Single Linkage method plot it is almost impossible to recognize the different division in groups. The most clear is the graph related to the Average Linkage method, it provides a more clear subdivision of units though it needs the highest value for k. Complete Linkage plot and Ward Linkage plot seems to be quite similar and they require the same value for k.