Exercises 3-4

Giovanni Zurlo

19/10/2021

Exercise 1

> *Consider the following dataset with n = 4 observations and p = 5 vari- ables, 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.*

	V1	V2	V3	V4	V5
x1	blue	TRUE	TRUE	FALSE	12
x2	red	FALSE	FALSE	NA	NA
x3	red	TRUE	FALSE	NA	17
x4	green	TRUE	FALSE	FALSE	21

> **(a)** *Manually compute Gower dissimilarities based on distances for all variables separately* **(b)** *Compute Gower dissimilarities treating variables 2-4 as a single group on which you compute a Jaccard dissmilarity. Does this give the same result as part (a)?* **(c)** *Compute the Gower dissimilarites using the daisy-function in R and check against the manual calculation in (a) and (b).*

```
# Computing Gower dissimilarities for all variables separately
Gow.d12=(1+1+1)/3
Gow.d13=(1+0+1+(17-12)/9)/4
Gow.d14=(1+0+1+0+1)/4
Gow.d23=(0+1+0)/2
Gow.d24=(1+1)/2
Gow.d34=(1+0+0+(21-17)/9)/3

# Organizing the coefficients in a "daisy-like" output matrix
dist=c(Gow.d12,Gow.d13,Gow.d23,Gow.d14,Gow.d24,Gow.d34)
dist.mat <- matrix(0, nrow = 3, ncol = 3)
rownames(dist.mat)=c("x1","x2","x3")
colnames(dist.mat)=c("x2","x3","x4")
dist.mat[upper.tri(dist.mat, diag = TRUE)] <- dist
  kable(t(dist.mat),"html", row.names = T) |>
    kable_styling(full_width = F, position = "center")
```

	x1	x2	x3
x2	1.0000000	0.0	0.0000000
x3	0.6388889	0.5	0.0000000
x4	0.7500000	1.0	0.4814815

Observations which show maximum dissimilarity are *(x1,x2)* and *(x2,x4)*. Treating binary variables *V2-V4* as asymmetric has an important impact on these computations, since joint absences (*x(il) = x(jl) = 0*) are treated as missing values (not taken into account in the denominator too).

```
# Computing Jaccard dissimilarities on variables 2-4
j.d12=(1-0)
j.d13=(1-1/2)
j.d14=(1-1/2)
j.d23=(1)
j.d24=(1)
j.d34=(1-1)

# Computing final Gower diss. in which Jaccard diss. have weight 3
bGow.d12=(1+3*j.d12)/4
bGow.d13=(1+3*j.d13+(17-12)/9)/5
bGow.d14=(1+3*j.d14+1)/5
bGow.d23=(0+3*j.d23)/4
bGow.d24=(1+3*j.d24)/4
bGow.d34=(1+3*j.d34+(21-17)/9)/5

# Organizing the coefficients in a "daisy-like" output matrix
bdist=c(bGow.d12,bGow.d13,bGow.d23, bGow.d14, bGow.d24, bGow.d34)
bdist.mat <- matrix(0, nrow = 3, ncol = 3)
rownames(bdist.mat)=c("x1","x2","x3")
colnames(bdist.mat)=c("x2","x3","x4")
bdist.mat[upper.tri(bdist.mat, diag = TRUE)] <- bdist
  kable(t(bdist.mat),"html", row.names = T) |>
    kable_styling(full_width = F, position = "center")
```

	x1	x2	x3
x2	1.0000000	0.00	0.0000000
x3	0.6111111	0.75	0.0000000
x4	0.7000000	1.00	0.2888889

As you can see, results differ if we treat *V2-V4* as a single group with **weight = 3**. Dissimilarities which are somehow halfway through 0 and 1 are now slightly reduced since we take joint absences indirectly into account (by means of the constant weight 3 to Jaccard dissimilarities).

```
# Computing Gower dissimilarities using the daisy function
daisy(x,'gower')
```

```
## 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
```

These now correspond to the dissimilarities computed for all variables separately, since no grouping has occurred within the function itself. *Daisy* takes, by default, factors as nominal (uses simple matching) and binary ones as asymmetric (uses Jaccard), which is exactly how I computed my coefficients at the beginning.

Exercise 2

> *Give counterexamples to show that the correlation dissimilarity and the Gower coeffcient do not fulfill the triangle inequality (in each case present three observations).*

Correlation Dissimilarity

Considering the correlation dissimilarity

$$d(x,z) = \frac{1}{2}(1 - r(x,z))$$

we can derive an equivalent condition for the triangle inequality as:

$$d(x,z) \leq d(x,y) + d(y,z)$$
$$\frac{1}{2}(1 - r(x,z)) \leq \frac{1}{2}(1 - r(x,y)) + \frac{1}{2}(1 - r(y,z))$$
$$r(x,y) + r(y,z) \leq 1 + r(x,z)$$

By exploiting this last inequality, I derived a proper covariance matrix for a bivariate gaussian distribution whose observations violate the triangle inequality.

```
library(MASS)
set.seed(1234)
# Creating a function for the correlation dissimilarity
corr.d <- function(a,b) {0.5*(1-cor(a,b))}

# Defining the covariance matrix of X and Z
S<- matrix(c(3,0.5,1,0.5), nrow=2)
# Generating two random observations from a multivariate gaussian
rmatrix <- mvrnorm(n=4, mu=c(0,0), Sigma=S, empirical=TRUE)
X <- rmatrix[,1] # mean 0, variance 3
Z <- rmatrix[,2] # mean 0, variance 1
Y  <- X + Z

# Checking the inequalities
cor(X,Y) + cor(Y,Z) <= 1 + cor(X,Z)
```

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

```
corr.d(X,Z) <= corr.d(X,Y) + corr.d(Y,Z)
```

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

The inequalities return 'FALSE' so we found a counterexample.

Random Counterexample

X	0.6280264	-1.0574423	-1.708206	2.1376224
Z	0.6613098	0.3718598	-0.958552	-0.0746176
Y	1.2893361	-0.6855824	-2.666758	2.0630047

	(X,Z)	(X,Y)	(Y,Z)
Dissimilarity	0.2958759	0.0237103	0.1666667
Correlation	0.4082483	0.9525793	0.6666667

Gower Dissimilarity

Observations *X2*, *X3* and *X4* from Exercise 1 do not satisfy the triangle inequality for Gower dissimilarity.

$$d_G(X_2, X_4) \leq d_G(X_2, X_3) + d_G(X_3, X_4)$$

Counterexample from Exercise 1

x2	red	FALSE	FALSE	NA	NA
x3	red	TRUE	FALSE	NA	17
x4	green	TRUE	FALSE	FALSE	21

	(X2,X4)	(X2,X3)	(X3,X4)
Dissimilarity	1	0.5	0.4814815

Exercise 4

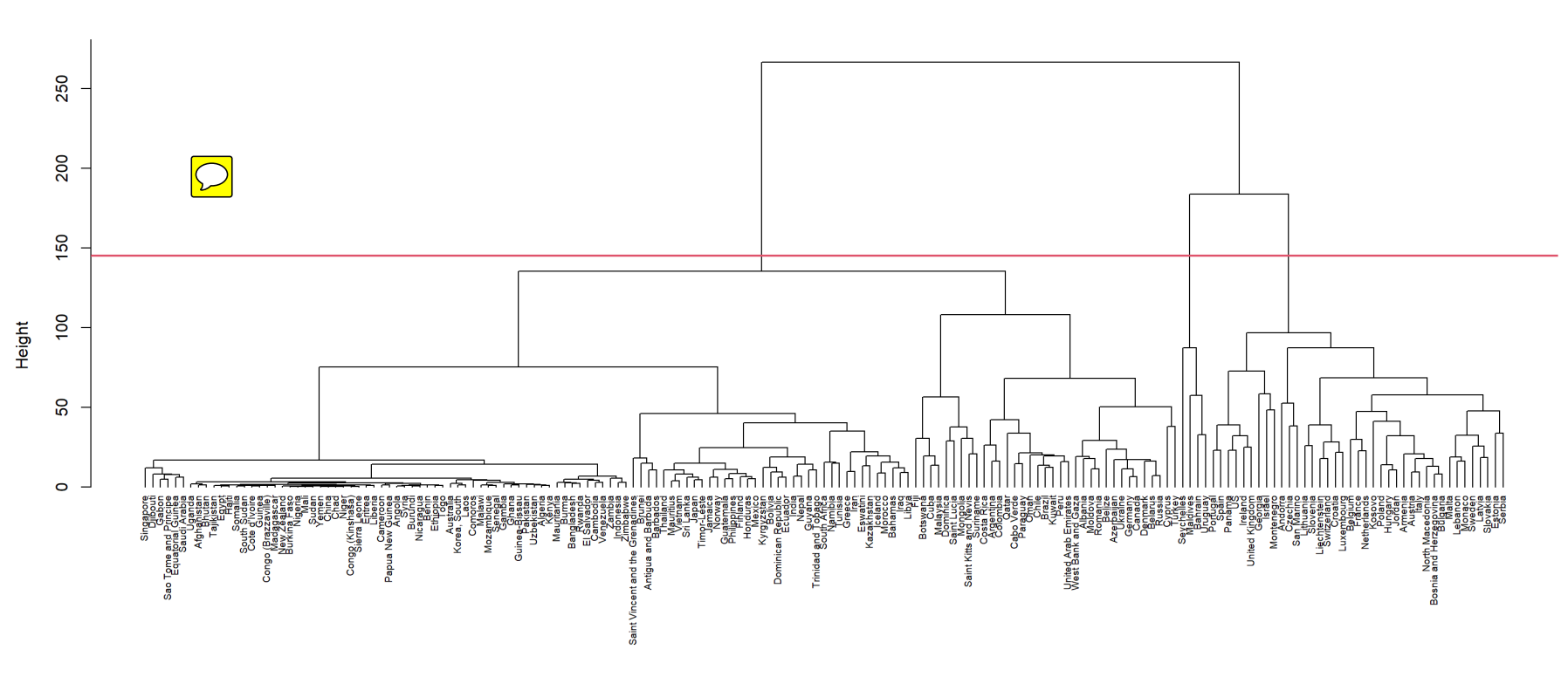
> *On Virtuale you can find the data set covid2021.dat. This data set has time series characterising the spread of Covid-19 in 179 countries.The time span is 1 April 2020 to 7 October 2021. Data give for each day the number of additional cases in the previous week divided by the country's population (in 1,000). 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.*

```
covid2021 <- read.table(file.choose())
x <- covid2021[,5:559] # This selects the variables for clustering
```

The analysis is performed on unscaled data since all variables have same measurement unit and different variability can itself be meaningful.

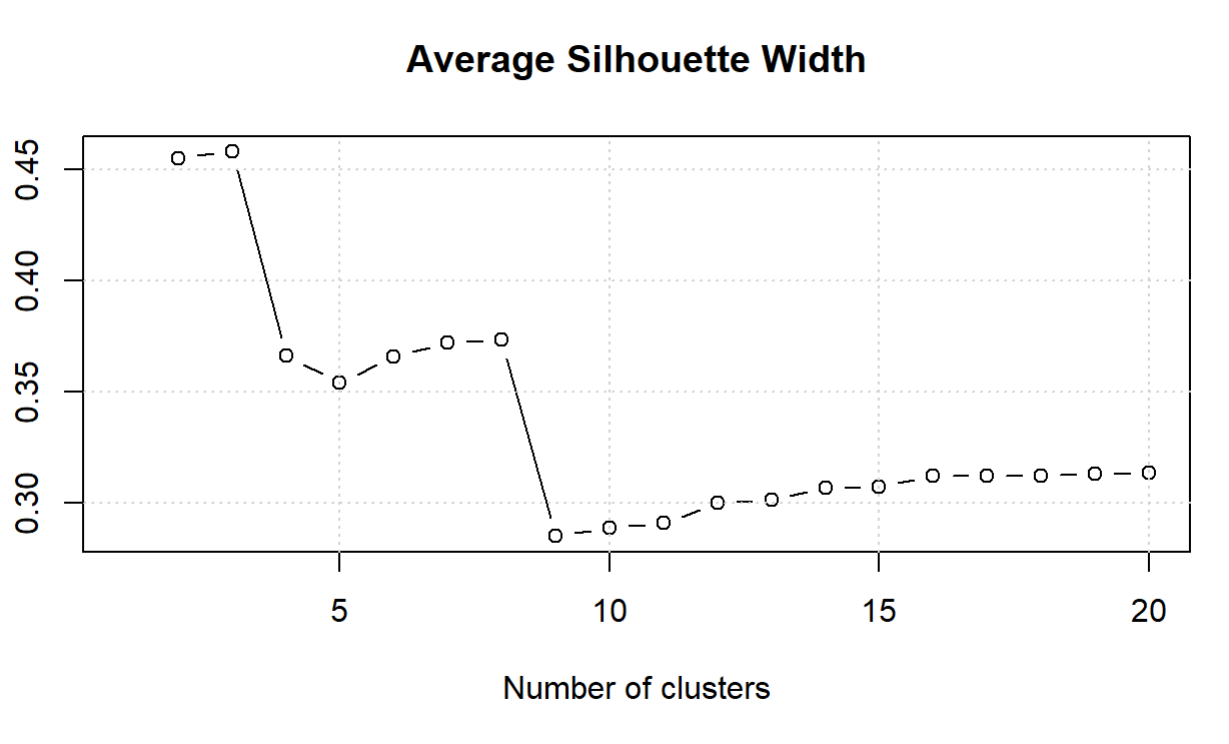
Euclidean Distance + Ward Method

```
# Computing the distance matrix
d.eucl=dist(x)
# Performing Hierarchical Cluster Analysis
out.eucl=hclust(d.eucl,method = "ward.D2")
# Plotting Dendrogram
plot(out.eucl, hang=-1, xlab="",sub="" ,cex=0.6, cex.axis=1, main="")
abline(h=145, col=2, lwd=2)
```



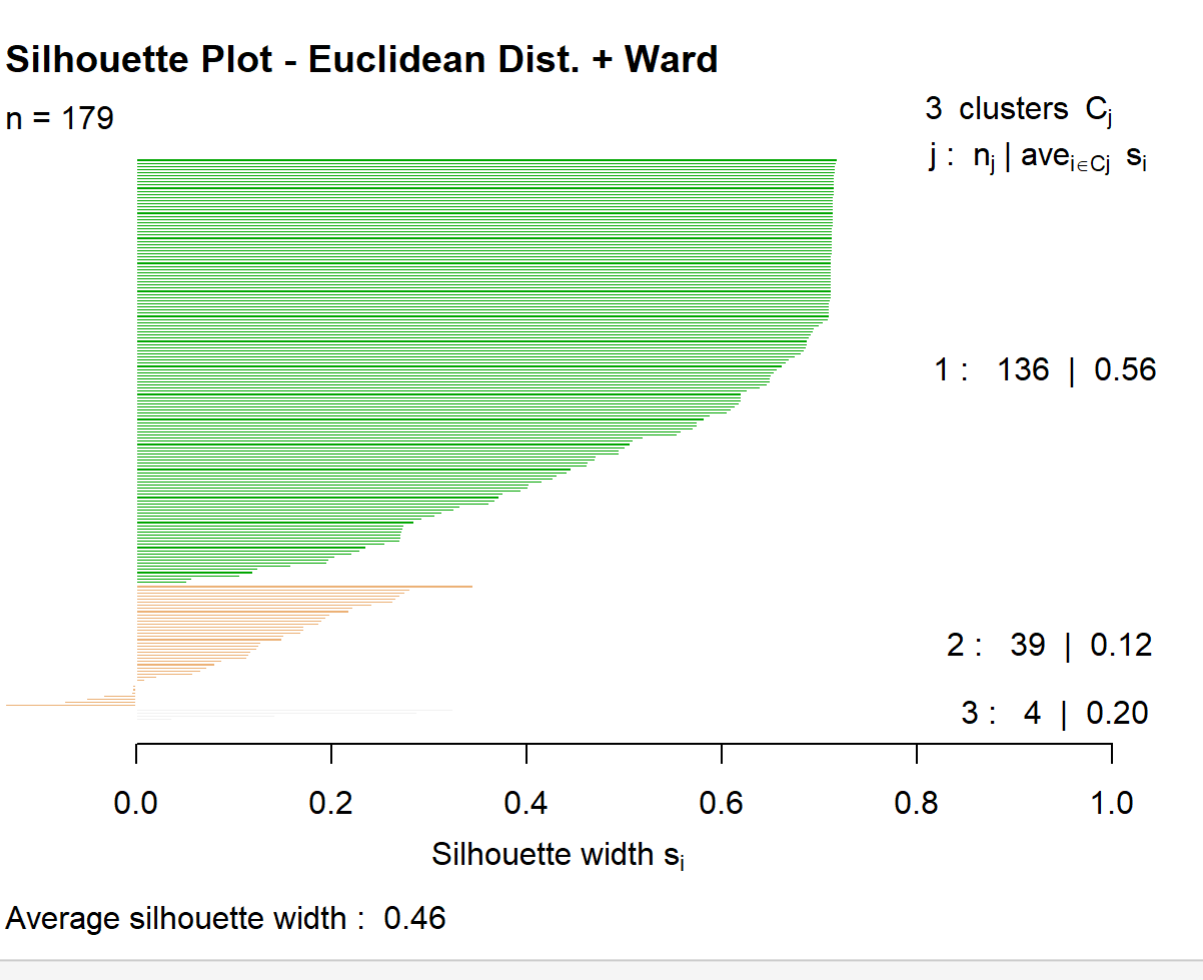
Last Aggregations

	-Unit	Group	Height
[166,]	159	160	57.77
[167,]	-61	161	58.36
[168,]	157	162	68.09
[169,]	153	166	68.40
[170,]	154	167	72.73
[171,]	108	158	75.37
[172,]	-142	165	87.23
[173,]	163	169	87.25
[174,]	170	173	96.86
[175,]	164	168	108.00
[176,]	171	175	135.26
[177,]	172	174	183.68
[178,]	176	177	266.66



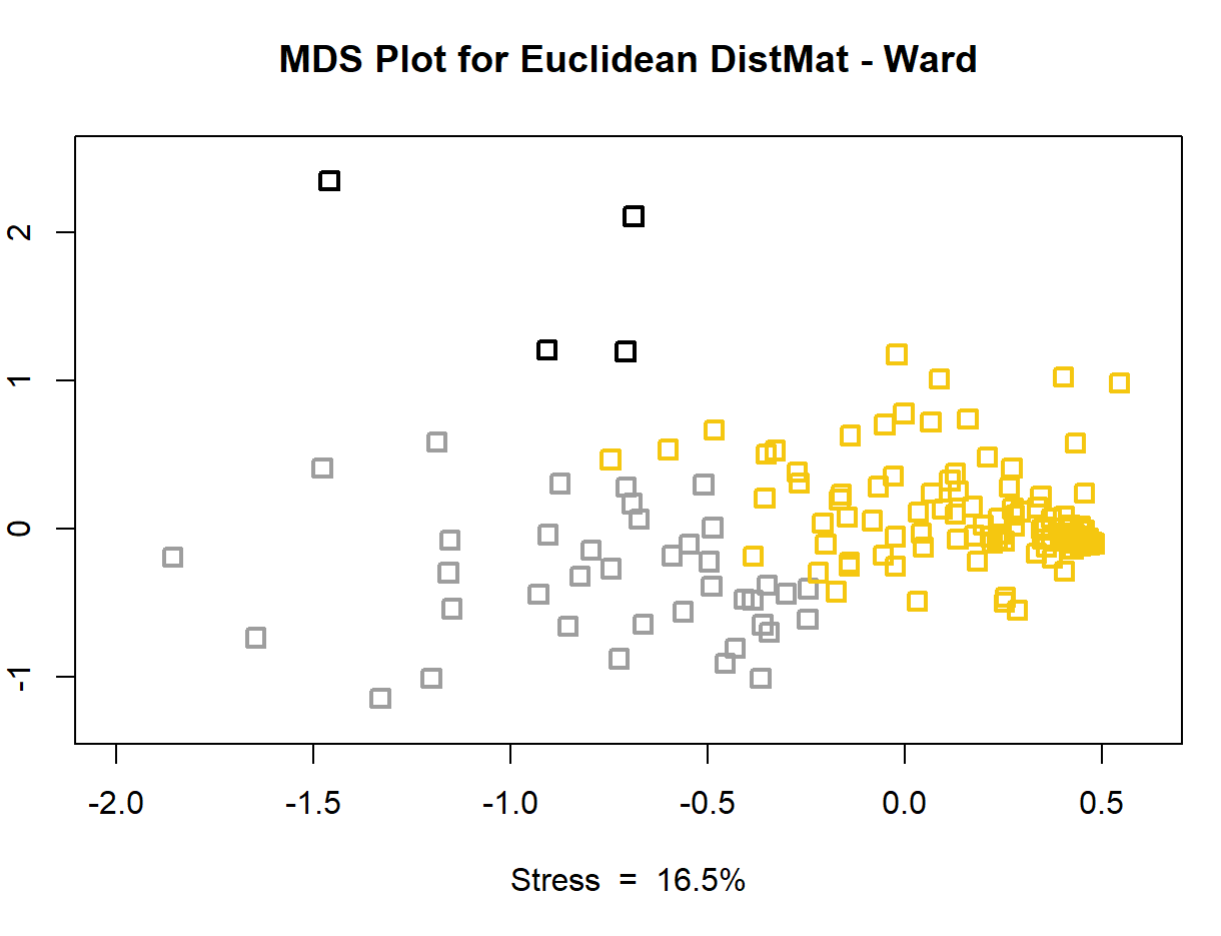
The decision here was between *K = 3*, that is the value with the highest ASW, and *K = 8*, which may represent the elbow of the scree plot. However, the second alternative led to several small clusters (5 of them with at most 9 units) with very low average silhouette values, and a significant reduction in the global clustering ASW. So I opted for *K = 3*. It is immediately clear that the right main branch of the dendrogram, representing European and East-European countries, is quite heterogeneous and doesn't show any clear group structure (*ASW = 0.08*). The same can also be said for the center-right South-American countries cluster (*ASW = -0.07*).

```
# Storing the 3 Clusters Partition
groups.eucl=cutree(out.eucl, k=3)
# Plotting Silhouette Plot for the Partition Obtained
plot(tsil[[3]], col=terrain.colors(3),
     main="Silhouette Plot - Euclidean Dist. + Ward")
```




```
# Applying Ratio MDS on the Euclidean Dist.Matrix
library(smacof)
mds.eucl=mds(d.eucl, type = 'ratio')

# Plotting a 2D Representation
plot(mds.eucl$conf, main="MDS Plot for Euclidean DistMat - Ward",
     col=(groups.eucl+6), cex=1.2,pch=0,lwd=2, xlim=c(-2,0.6),ylim=c(-1.3,2.5),
     xlab="Stress  =  16.5%",ylab='')
```



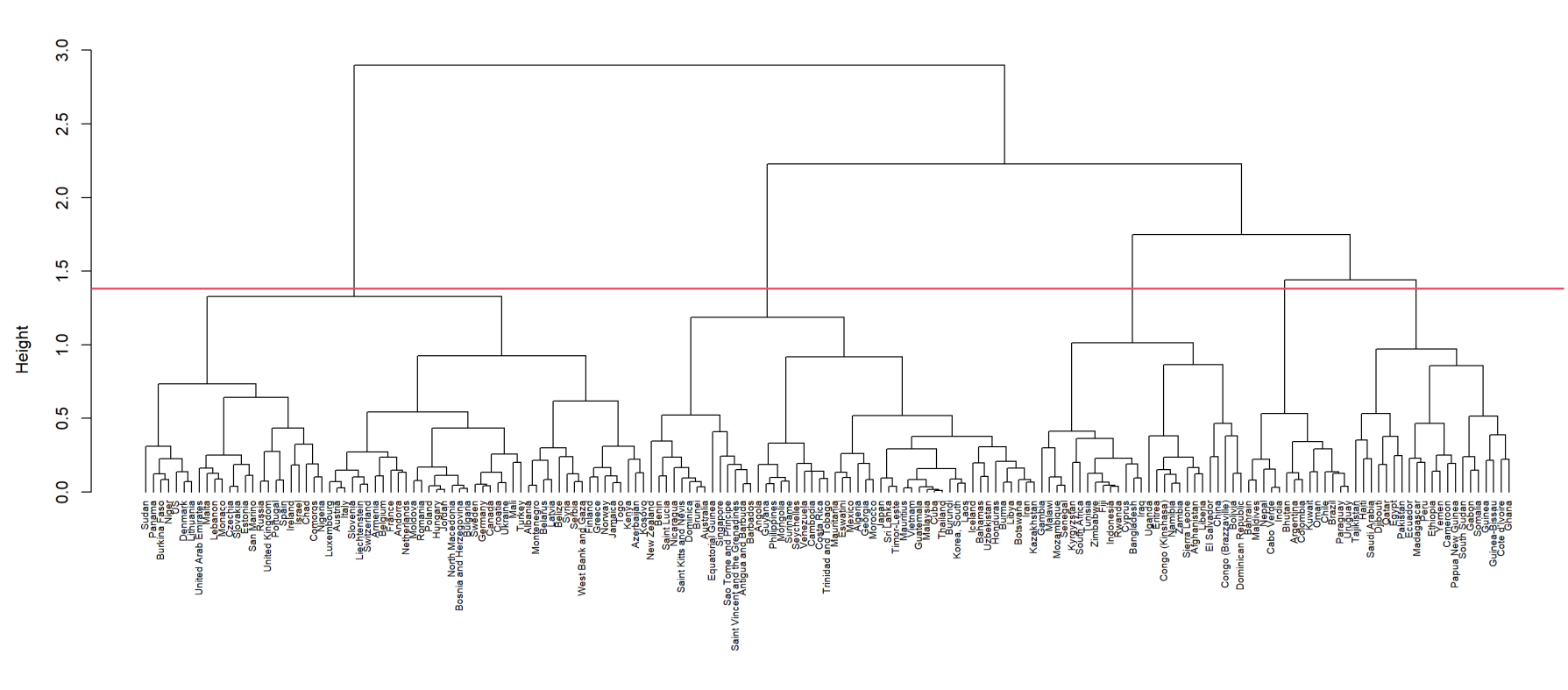
My interpretation of these clusters is the following: we have 39 European and Mediterranean countries in cluster #2 (39 units, in grey) vs. the rest of the world (Cluster #1, in yellow) which includes Third World countries that officially suffered limited Covid-19 outbreaks (very homogeneous group, we can also find AUS, NZ and Singapore) and two other groups collecting Asian and South-American countries. In the 3rd cluster, Ward method isolated 4 "anomalous" countries (in black): Seychelles, Maldives, Uruguay and Bahrain. There exists a weak correspondence between clusters and the Continent variable (as measured by the Adjusted Rand Index, up to 0.25 for k=5).

Correlation Dissimilarity + Ward Method

The correlation dissimilarity I've chosen is the "asymmetric" version from slide 126, for which inverse correlation is indicative of dissimilarity:

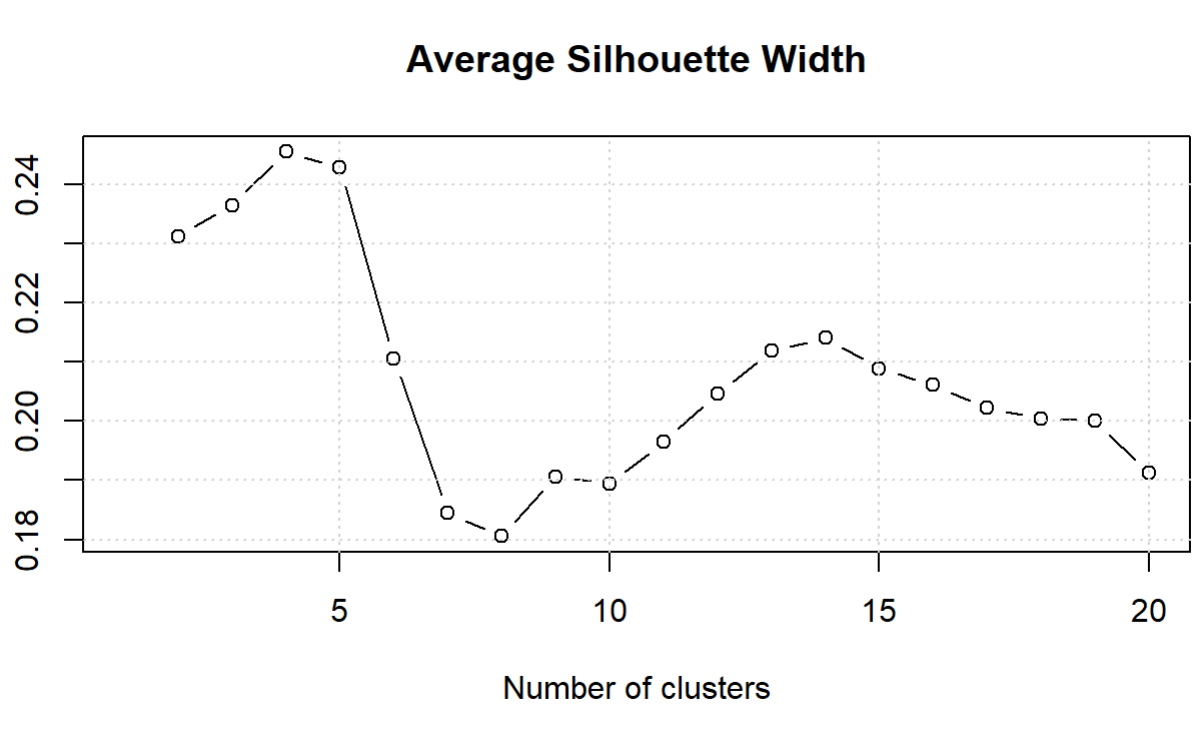
$$d(x,z) = \frac{1}{2}(1 - r(x,z))$$

```
# Computing the distance matrix
r.mat=cor(t(x))
d.corr=as.dist(0.5*(1-r.mat))
# Performing Hierarchical Cluster Analysis
out.corr=hclust(d.corr,method = "ward.D2")
# Plotting Dendrogram
plot(out.corr, hang=-1, xlab="",sub="" ,cex=0.6, cex.axis=1, main="")
abline(h=1.38, col=2, lwd=2)
```



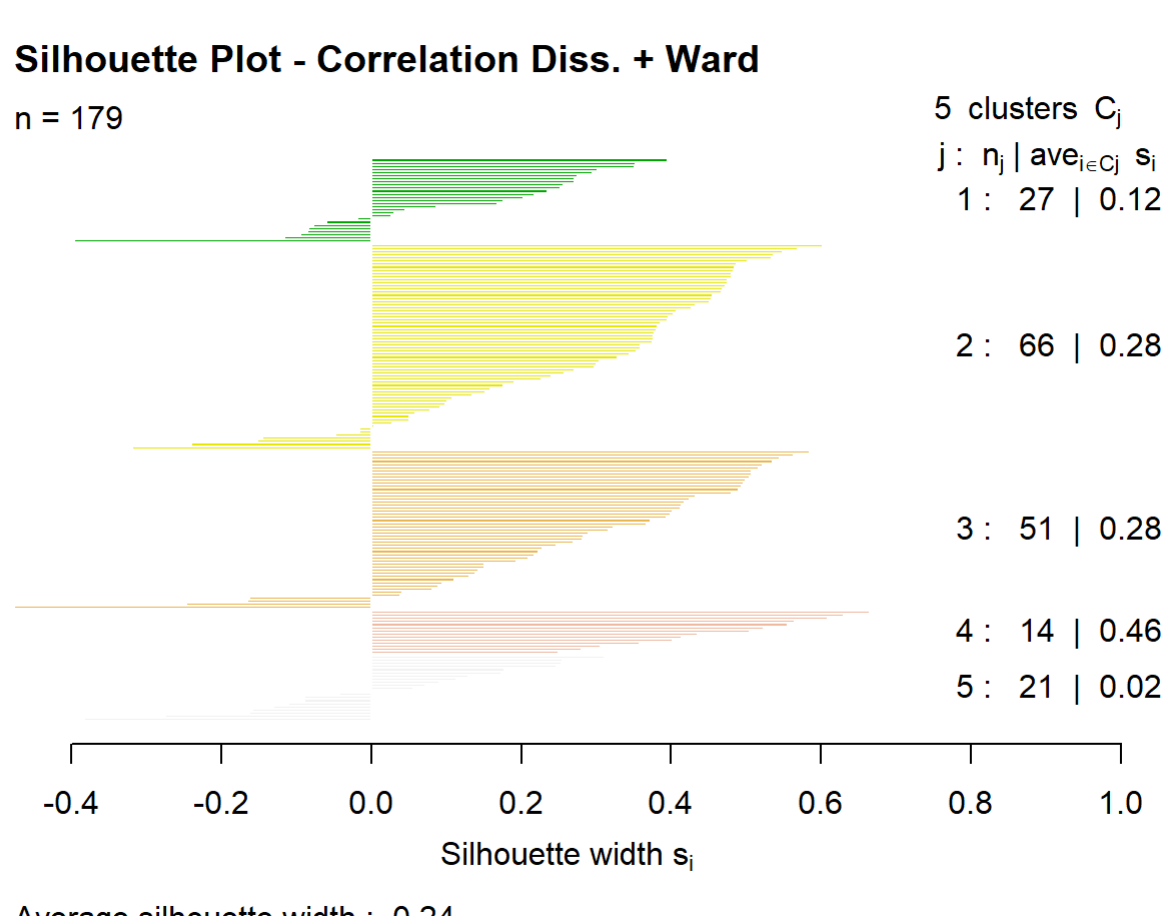
Last Aggregations

	-Unit	Group	Height
[166,]	140	165	0.73
[167,]	156	158	0.86
[168,]	150	157	0.87
[169,]	142	159	0.92
[170,]	163	164	0.93
[171,]	161	167	0.97
[172,]	153	168	1.01
[173,]	160	169	1.18
[174,]	166	170	1.33
[175,]	162	171	1.44
[176,]	172	175	1.75
[177,]	173	176	2.23
[178,]	174	177	2.90



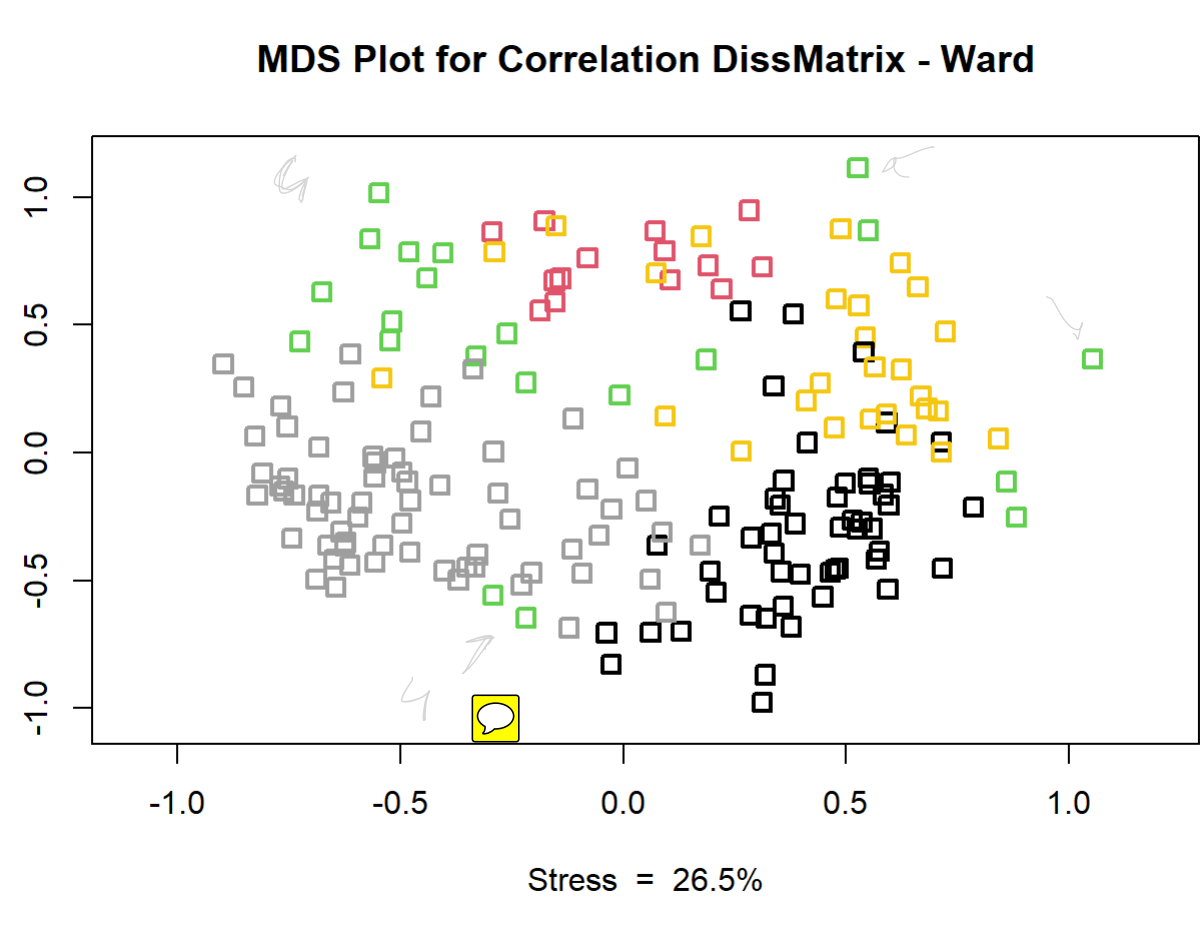
The decision here was between *K = 5*, for which we have the second highest ASW, and *K = 8*, which may represent more clearly the elbow of the scree plot. As you can imagine I preferred the 5 clusters solution, even over *K = 4* since the separation of the 4th cluster from this last partition led to a quite homogeneous group (#4, 14 units). The usage of a correlation dissimilarity between the rows of our dataset produced poorer results in terms of Average Silhouette Width, which indicates weak and unstable partitions.

```
# Storing the 3 Clusters Partition
groups.corr=cutree(out.corr, k=5)
# Plotting Silhouette Plot for the Partition Obtained
plot(tsil[[5]], col=terrain.colors(5),
     main="Silhouette Plot - Correlation Diss. + Ward")
```




```
# Applying Ratio MDS on the Euclidean Dist.Matrix
library(smacof)
mds.corr=mds(d.corr, type = 'ratio')

# Plotting a 2D Representation
plot(mds.corr$conf, main="MDS Plot for Correlation DissMatrix - Ward",
     col=(groups.corr+6), cex=1.2,pch=0,lwd=2,xlim=c(-1.1,1.2),ylim=c(-1.05,1.15),
     xlab="Stress  =  26.5%",ylab='')
```



My suspects about the absence of a clear structure found further confirmation from the analysis of this 2-Dimensional plot. Even though such representation suffers a significant loss of information (26% of "stress"), it is clear that our partition does not guarantee neither homogeneity nor separation. Also, interpretation of the groups is harder: we have weaker correspondence with continents (Adj. Rand Index = 0.21) but also clusters with a more comparable homogeneity and size than those produced by means of the euclidean distance.

```
library(mclust)
# Checking Similarity between Clusterings
adjustedRandIndex(groups.corr,groups.eucl)
```

```
## [1] 0.07589103
```

```
# Checking Correspondence with Continent Grouping
adjustedRandIndex(groups.corr,covid2021$continent)
```

```
## [1] 0.2117788
```

Interestingly, according to the adjusted Rand Index, the partition obtained earlier and this one based on correlation dissimilarities are very different. This maybe tells us that these represent a different approach to clustering this kind of data: the euclidean one let us analyze countries from the point of view of the overall Covid-19 impact, while correlation focuses more on the "patterns" of the pandemic waves. I'm a bit skeptical about the suitability of simple Pearson's correlation coefficients between cases curves, since the pandemic showed complex and often unclear dynamics. Practically, the partition based on euclidean distances seems more convincing to me, kind of faithful to reality, apart from being better. Of course such result could be improved by including more and more useful indicators about Covid-19, since new cases statistics proved to be unreliable and significantly biased.

Exercise 5

Focusing just on Complete Linkage clustering, what is the best distance in these experiments? Explain how this can be seen from the plots regarding the Complete Linkage results

This research paper shows that the L1-distance has a lot of largely unexplored potential for clustering in high dimensional data. Simulations were run in five different setups, with 100 replicates generated as 100 units from two balanced classes and *p = 2000* independent dimensions (Gaussian or t-distributed, proportion of noise variables different according to the setting). Results were plotted as interaction lines showing the mean results over different standardization and aggregation methods, based on 4 p-Minkowski distances. These clustering results are compared with the true 2-class clustering by means of the Adjusted Rand Index, which is equal to 1 if identical. The L1 aggregation is the best in almost all respects, often with a big distance to the others and often delivering a certain number of perfect results (ARI = 1); in general, using city block distance leads to results that are closer to the truth, in a way that is independent both from the setting (very noisy or simple) and the standardization method, whose optimal choice depends on the setting and has a consistent weight.

Explain roughly the basic idea of boxplot transformation

The boxplot transformation is a new standardization technique which also tame the influence of outliers on any variable. For outliers, here we mean outlying values on single variables, which are very common in high dimensional data, rather than full outlying p-dimensional observations (as are treated in the robust statistics field). Usual standardization techniques do not necessarily solve the issue of outliers, and may even be influenced themselves by extreme values (*std to unit variance or unit range*); distance-based statistical techniques do suffer from the presence of outliers. The idea of boxplot transformation is to standardise the lower and upper quantile *linearly* to [-0.5, 0.5]; the same happens to all the other observations if there are no outliers over or below the median. If there are outliers below the median, all the observations on the lower external side of "the box" are then transformed by a non-linear transformation that maps to [-2, -0.5], so that outliers are brought closer to the sample and these are no longer anomalous by the boxplot definition. The same happens to observations between the 3rd quartile and the maximum, since they are mapped to [0.5,2] (if there are upper outliers). Such boxplot transformation proved to perform well in simulated settings where there was a strong contrast between many noise variables and few variables with strongly separated classes.

EXERCISE 3

Given two p-dimensional binary vectors x, y, without missing values, the following dissimilarities are EQUIVALENT:

$$d_G(x,y) = \frac{\sum_{i=1}^{p} \delta_i \, d_i(x_i, y_i)}{\sum_{i=1}^{p} \delta_i}$$

SINCE $w_i = 1$ for $i = 1, \dots, p$

$s_i = \max d_i = 1$

(NO SCALING FOR BINARY DATA)

$$d_J(x,y) = \frac{\sum_{i=1}^{p} 1(x_i = 1 \text{ or } y_i = 1) - \sum_{i=1}^{p} 1(x_i = 1 \,\&\, y_i = 1)}{\sum_{i=1}^{p} 1(x_i = 1 \text{ or } y_i = 1)}$$

NUMERATOR:

Joint absences ($x_i = 0 \,\&\, y_i = 0$) are handled as missing values in the computations of Gower's coefficient, i.e.

$$\delta_i = 0 \quad \text{if} \quad x_i = 0 \,\&\, y_i = 0$$

This means that we only count disagreements, for which $d_i(x_i, y_i) = d_J(x_i, y_i) = 1$. This is equivalent to Jaccard's numerator:

$$\sum_{i=1}^{p} \delta_i d_i(x_i, y_i) = \sum_{i=1}^{p} 1(x_i = 1 \,\&\, y_i = 0) + \sum_{i=1}^{p} 1(x_i = 0 \,\&\, y_i = 1)$$

DENOMINATOR:

$$\sum_{i=1}^{p} \delta_i = \sum_{i=1}^{p} 1(x_i = 1 \text{ or } y_i = 1)$$

They both count pairs for which none of x_i, y_i is 0. Both denominators can be zero, overall, if we only observe joint absences, in that case $d_J(x,y)$ and $d_G(x,y)$ would be undefined