Exercise 3

Riccardo Petrella

2023-10-13

1.

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 fifth is on a continuous scale. "NA" denotes missing values.

$$
\begin{aligned}
\mathbf{x}_1 &= (\text{ blue },1,1,0,12)\\
\mathbf{x}_2 &= (\text{ red}, 0,0,\mathrm{NA},\mathrm{NA})\\
\mathbf{x}_3 &= (\text{ red}, 1,0,\mathrm{NA},17)\\
\mathbf{x}_4 &= (\text{ green}, 1,0,0,21)
\end{aligned}
$$

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).

Solution: The formula to compute the Gower dissimilarities is represented as follows:

$$d_G(\mathbf{x}_i,\mathbf{x}_j)=\frac{\sum_{l=1}^{p}\frac{w_l}{s_l}\delta_{ijl}d_l(x_{il},x_{jl})}{\sum_{l=1}^{p}\delta_{ijl}w_l},$$

Its legend can be consulted at slide 139. However, two remarks are shown below:

$$\delta_{ijl}=1 \text{ (none of } \mathbf{x}_{il},\mathbf{x}_{jl} \text{ missing).}$$

and:

s_l : scaling of d_l; Gower: $s_l=\max d_l$, for every l such that $l=1,..,p$.

Remark: we assume $\frac{0}{0}=0$.

We separately compute all 6 distances between the vectors:

$$d(\mathbf{x}_1,\mathbf{x}_2)=\frac{1+\left(1-\frac{0}{1}\right)+\left(1-\frac{0}{1}\right)+0}{1\cdot1+1\cdot1+1\cdot1}=\frac{3}{3}=1$$

$$d(\mathbf{x}_1,\mathbf{x}_3)=\frac{1+\left(1-\frac{1}{1}\right)+\left(1-\frac{0}{1}\right)+\frac{|12-17|}{9}}{1+1+1+1}\simeq0.6389$$

$$d\left(\mathbf{x}_1,\mathbf{x}_4\right)=\frac{1+\left(1-\frac{1}{1}\right)+\left(1-\frac{0}{1}\right)+0\left(1-\frac{0}{0}\right)+\frac{|12-21|}{9}}{1+1+0+1+1}=\frac{3}{4}=0.75$$

$$d\left(\mathbf{x}_2,\mathbf{x}_3\right)=\frac{0+\left(1-\frac{0}{1}\right)+0\left(1-\frac{0}{0}\right)+0}{1+1}=\frac{1}{2}=0.5$$

$$d\left(\mathbf{x}_2,\mathbf{x}_4\right)=\frac{1+\left(1-\frac{0}{1}\right)+\left(1-\frac{0}{0}\right)+0}{1+1+1}=\frac{3}{3}=1$$

$$d\left(\mathbf{x}_3,\mathbf{x}_4\right)=\frac{1+\left(1-\frac{1}{1}\right)+0\left(1-\frac{0}{0}\right)+\frac{|17-21|}{9}}{3}=\frac{13}{27}\simeq 0.4815$$

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

Solution: We create:

- 1 vector to be used "as factor" (i.e. the qualitative $\mathbf{x}_1$)
- 3 binary vectors (i.e. $\mathbf{x}_2$,$\mathbf{x}_3$,$\mathbf{x}_4$)
- 1 vector "as numeric" for continuos values (i.e. $\mathbf{x}_5$).

Then we apply the "daisy" function:

```
x1 <- c('blue','red','red','green')
x2 <- c(1,0,1,1)
x3 <- c(1,0,0,0)
x4 <- c(0, NA, NA, 0)
x5 <- c(12, NA, 17, 21)

x <- data.frame(x1,x2,x3,x4,x5)
x[,1] <- as.factor(x[,1])
x[,2] <- as.factor(x[,2])
x[,3] <- as.factor(x[,3])
x[,4] <- as.factor(x[,4])

r_gower <- daisy(x , metric =" gower ", type = list(asymm=c(2,3,4)))

r_gower
```

```
## Dissimilarities :
##           1         2         3
## 2 1.0000000
## 3 0.6388889 0.5000000
## 4 0.7500000 1.0000000 0.4814815
##
## Metric :  mixed ;  Types = N, A, A, A, I
## Number of objects : 4
```

Here is the matrix with the handmade Gower dissimilarities:

$$\left[\begin{array}{ccccc} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_1 & 0 & & & \\ \mathbf{x}_2 & 1 & 0 & & \\ \mathbf{x}_3 & 0.6389 & 0.5 & 0 & \\ \mathbf{x}_4 & 0.75 & 1 & 0.4815 & 0 \end{array}\right]$$

Remark: the handmade-Gower dissimilarities and the ones computed with "Daisy" are fortunately equal. Daisy should manage NA values correctly and should apply the right type of dissimilarity depending on the type of data. This is particularly true when the command "type" is used inside the function. If we take a look at the output of the console, we notice that the differences between the rows of x are correctly labeled as "N", "A" and I" (i.e. Nominal, Asymmetric and Integer).

2. (a)

Show that Simple Matching and Mahalanobis distance are dissimilarities and also distances (i.e., fulfill the triangle inequality).

We obviously know that $d_{SM}(x,x) = 0$ and that $d_{SM}(x,y) = d_{SM}(y,x)\ \forall x, y \in \mathbb{R}^n$. We just need to show the triangular inequality.

By assuming the existence of 3 vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ whose elements are in $\mathbb{R}$, we define the Simple Matching dissimilarity for all of them:

$$d_{SM}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{p}\sum_{j=1}^{p} 1\left(x_{1j} \neq x_{2j}\right)$$

$$d_{SM}(\mathbf{x}_1, \mathbf{x}_3) = \frac{1}{p}\sum_{j=1}^{p} 1\left(x_{1j} \neq x_{3j}\right)$$

$$d_{SM}(\mathbf{x}_2, \mathbf{x}_3) = \frac{1}{p}\sum_{j=1}^{p} 1\left(x_{2j} \neq x_{3j}\right)$$

Assuming that $x_{1j} \neq x_{2j}$, $x_{1j} \neq x_{3j}$, $x_{2j} \neq x_{3j}$ are true for $j = 1, ..., p$, we get that all the indicator functions are equal to 1:

$$1\left(x_{1j} \neq x_{2j}\right) + 1\left(x_{2j} \neq x_{3j}\right) > 1\left(x_{1j} \neq x_{3j}\right)$$

$$\forall j$$

As a consequence:

$$d_{SM}(\mathbf{x}_1, \mathbf{x}_2) + d_{SM}(\mathbf{x}_2, \mathbf{x}_3) > d_{SM}(\mathbf{x}_1, \mathbf{x}_3)$$

$$\forall j$$

For the Mahalanobis Distance, we trivially know that: $d_M(x,x) = 0$ and $d_M(x,y) = d_M(y,x)$, $\forall x, y \in \mathbb{R}^n$.

It remains to prove the triangular inequality using the hint and some algebraic properties of matrices:

$$d_M^2(x,y) = (x-y)^T S^{-1}(x-y) =$$
$$(x-y)^T \left(UDU^T\right)^{-1}(x-y) =$$
$$(x-y)^T \left(UD^{-1/2}D^{-1/2}U^T\right)(x-y) =$$
$$\left(x^T UD^{-1/2} - y^T UD^{-1/2}\right)\left(D^{-1/2}U^T x - D^{-1/2}U^T y\right) =$$
$$\left(D^{-1/2}U^T x - D^{-1/2}U^T y\right)^T \left(D^{-1/2}U^T x - D^{-1/2}U^T y\right) =$$
$$d^2\left(D^{-1/2}U^T x, D^{-1/2}U^T y\right)$$

The last expression is nothing but an alternative way to express the Euclidean distance $d_{L2}(x,y)$.

It remains to prove that the Euclidean distance is actually a metric (i.e. satisfies the triangular inequality).

Given the definition of the Euclidean distance below:

$$d_{L2}(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\| = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

Let us focus on the sum inside the square root as the interesting part of our proof:

$$\begin{aligned}
\sum_{i=1}^{n}(x_i-y_i)^2 &= \sum_{i=1}^{n}(x_i-z_i+z_i-y_i)^2 \\
&= \sum_{i=1}^{n}(x_i-z_i)^2 + \sum_{i=1}^{n}(z_i-y_i)^2 + 2\sum_{i=1}^{n}(x_i-z_i)(z_i-y_i) \\
&\le \sum_{i=1}^{n}(x_i-z_i)^2 + \sum_{i=1}^{n}(z_i-y_i)^2 + 2\left(\sum_{i=1}^{n}(x_i-z_i)^2\right)^{\frac{1}{2}}\left(\sum_{i=1}^{n}(z_i-y_i)^2\right)^{\frac{1}{2}} \\
&= \left(\left(\sum_{i=1}^{n}(x_i-z_i)^2\right)^{\frac{1}{2}} + \left(\sum_{i=1}^{n}(z_i-y_i)^2\right)^{\frac{1}{2}}\right)^2
\end{aligned}$$

Finally:

$$d(x,y) \le d(x,z) + d(z,y)$$

2. (b)

Give counterexamples to show that the correlation dissimilarity (first version on slide 131) 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.

```
x1=c(0.02,3,0.03,0.4,1)
x2=c(7,3,-9,1,-5)
x3=c(16,-5,9,10,2)

d13=0.5-cor(x1,x3)/2
d13
```

```
## [1] 0.9583194
```

```
d12=0.5-cor(x1,x2)/2
d12
```

```
## [1] 0.3909199
```

```
d23=0.5-cor(x2,x3)/2
d23
```

```
## [1] 0.4147556
```

```
d12 + d23 > d13
```

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

The triangular inequality for the correlation dissimilarity is not satisfied.

Now let us show the example for the Gower dissimilarity:

```
y1 <-  c(0,1,1)
y2 <-  c(1,1,NA)
y3 <-  c(NA,NA,NA)
y3=as.integer(y3)

d <- data.frame(y1,y2,y3)
gow <- daisy(d,metric="gower")
gow
```

```
## Dissimilarities :
##     1   2
## 2 0.5
## 3 1.0 0.0
##
## Metric :  mixed ;  Types = I, I, I
## Number of objects : 3
```

```
dx1x2 <- 0.5
dx1x3 <- 1
dx2x3 <- 0

dx1x2+dx2x3 > dx1x3
```

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

We do not satisfy the triangular inequality with the Gower dissimilarity.

3. (a)

A political scientist wants to cluster the respondents of a questionnaire using a distance-based method. The questionnaire has preference questions with two response options of the type "do you prefer the current level of taxes, or do you prefer higher taxes with all money from the higher taxes being invested in the health system?" Generally the option "I prefer the current situation" is coded 0 and the option that suggests something else is coded 1. Would you prefer the simple matching distance or the Jaccard distance here? Why?

Assuming that binary variables can be measured both with the Simple Matching Distance and Jaccard Distance, which is probably correct in this case, and assuming that the main interest of the questionnaire is a democratic vote that will be used by a certain government to make a decision involving the public wellbeing, I would personally prefer the former. In facts in this scenario it might not be useful to give more importance to joint presences rather than joint absences. Indeed, if we assume that the respondents' decisions do have the same democratic value (i.e. weight), it is preferable to use the Simple Matching Distance since it treats all elements equally.

However, if we assume that the political scientist is doing the research just to see if the respondents are interested in changing the current situation (i.e. voting 1), it is better to give more weight to the respondents who want to increase the taxes. Therefore with this assumption, the binary variable becomes asymmetric and the Jaccard Distance is a better choice.

3. (b)

Geographers want to cluster areas in the Swiss alps according to danger from avalanches in order to produce a map with different colour codes for different danger levels, using a distance-based method. Their variables are, all for the year 2019: (i) the number of avalanches in the area, (ii) the average percentage of the area covered by an avalanche, (iii) number of persons injured or dead in incidents involving avalanches in the area, (iv) Swiss Francs investment in the security of the ski slopes in the area, (v) Swiss Francs budget for emergency rescue in the area. Would you prefer the Euclidean distance on raw data, the Euclidean distance on scaled data, the Manhattan distance on raw data, the Manhattan distance on scaled data, or the Mahalanobis distance for these data? Why? (Probably more than one option can convincingly be argued.)

I would personally avoid the Euclidean distance on raw data, since these variables obviously have very different scales and in Minkowski distances d_{Lq}, the higher the q the higher the weights of variables with larger distance. By using the Manhattan distance, we can reduce this problem on raw data but I do not see the advantage of using a rotation variant distance (i.e. Manhattan) in this case over a rotation invariant distance (i.e. Euclidean). Before using any Minkowski distance, with these variables it is definitely a good practice to standardize our data.

As a consequence, the Euclidean distance on scaled variables may perform well in cluster analysis. I would avoid to use the Manhattan distance regardless the standardization because the latter performs better than the Euclidean distance when the dataset is afflicted by high dimensionality which is probably not our situation since there are only 5 variables and it is likely to assume that the number of observations is much bigger.

If I had to choose only one distance to use on this dataset, I would pick the Mahalanobis distance, despite the fact that it is rarely used in cluster analysis.

Indeed, it presents a couple of advantages over Minkowski distances that are effective with these data. These pros are listed as follows:

- it is scale and rotation invariant. That means that we do not need to standardize the variables and their difference in ranges is not a problem anymore because the scaling is computed by computing the Mahalanobis distance itself (with S^{-1} in its expression).

- it is very useful when the variables are highly correlated and there is a sort of redundance in the information provided by them. That is because the Mahalanobis distance focuses on the direction of the variables where they provide the biggest variance, since S is usually a sample covariance matrix.

4.

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.

Choose a number of clusters, try to understand and interpret the clusters as good as you can, using the information in the data (using visualisation as you see fit), and built yourself an opinion which of the tried out clusterings is most appropriate, and how appropriate they are in general (you may not be happy with any of them, in which case you may think about what went wrong, and how a better clustering could be achieved, even if you currently don't know how to put such ideas into practice).

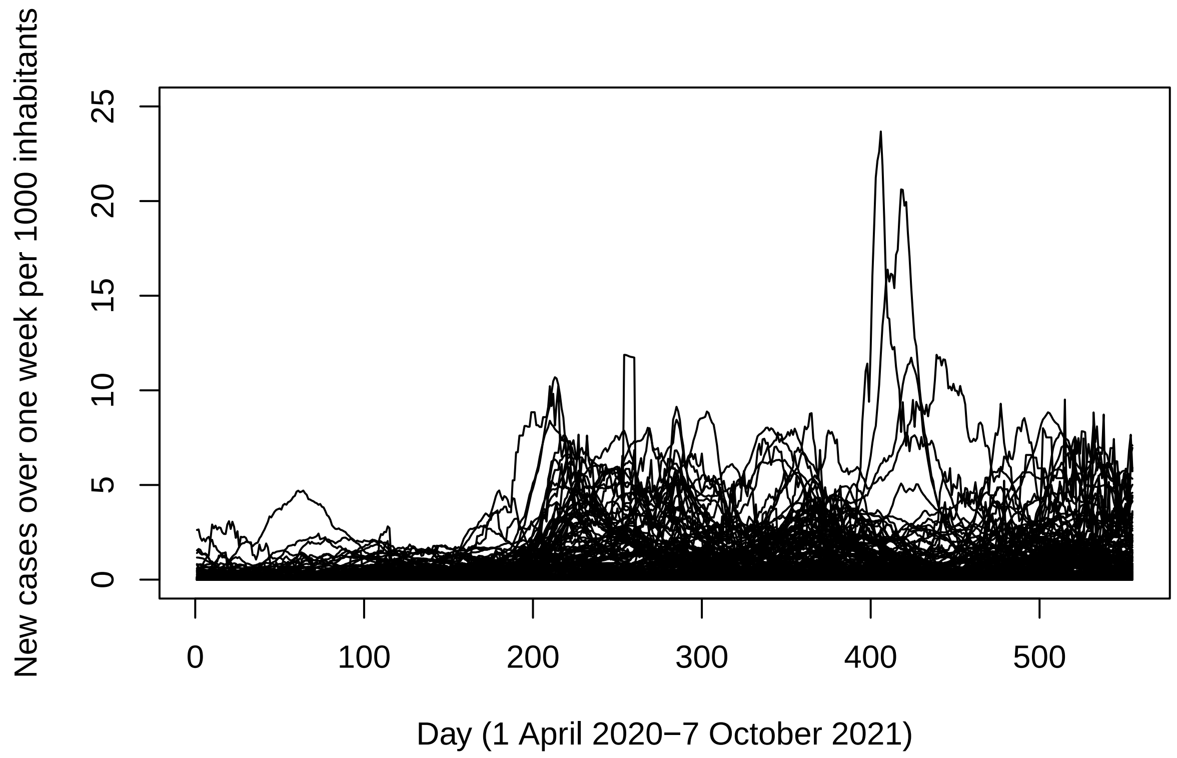
Visualise the data set using multidimensional scaling (different dissimilarities will different MDS plots), coloring the observations according to the clusters. You may also try to produce a good heatmap of the data

Let us proceed by uploading the data frame. We filter out the variables that are not useful to perform the cluster analysis.

```r
covid2021 <- read.table("covid2021.dat")
covid2021cl <- covid2021[,5:559] # This selects the variables for clustering
covid2021cl <- data.frame(covid2021cl)
```

Remark: this dataframe has a number of observation (179) which is much smaller than the number of variables (555). There may be a problem with high dimensionality which will have a negative impact on cluster analysis. The next chunk of code is used to plot the data

```r
plot(1:555,covid2021cl[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)  # the remaining observations (i.e. countries)
  points(1:555,covid2021cl[i,],type="l")
```

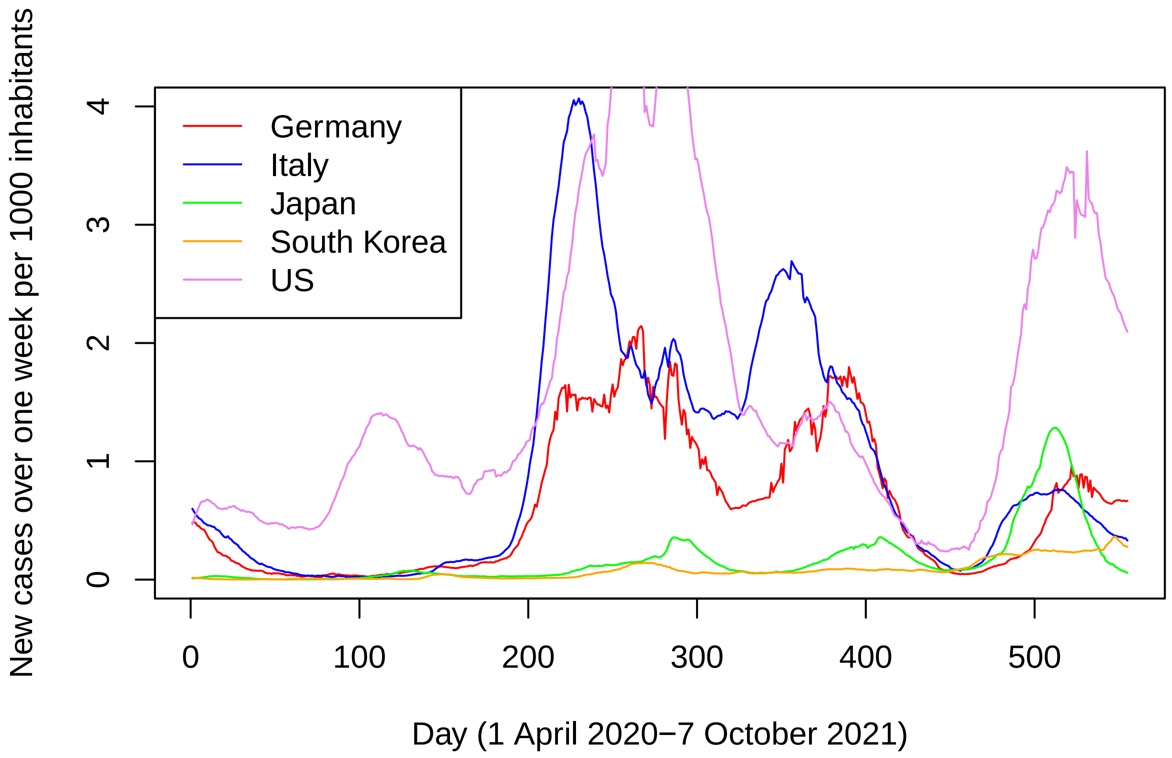


Unfortunately, this plot is not very informative.

For instance, we take 3 countries and we plot them to see the trends of their time series:

- Italy (79th row)
- Germany (62th row)
- US (164th row)
- South Korea (85th row)
- Japan (81st row)

```r
plot(1:555, covid2021cl[62,], type='l', col = 'red',
     ylab="New cases over one week per 1000 inhabitants",
     xlab="Day (1 April 2020-7 October 2021)", ylim=c(0, 4))

points(1:555, covid2021cl[79,], type = 'l', col='blue')
points(1:555, covid2021cl[81,], type = 'l', col='green')
points(1:555, covid2021cl[85,], type = 'l', col='orange')
points(1:555, covid2021cl[164,], type = 'l', col='violet')

legend("topleft", legend = c("Germany", "Italy", "Japan", "South Korea", "US"),
       col = c("red", "blue", "green", "orange", "violet"),
       lty = 1)
```



Here is a brief example on how to interpret the plot.

There are some periods where there is an increase in new weekly cases over 1000 inhabitants for the three western countries taken into account (for instance around the 200th day). Around day 400, there is also a decrease of cases in all the countries but South Korea, which is almost stable around 0 cases throughout the time series.

The two highest peaks above all the countries are reached by US around day 300 and day 500.

Optimal K with clusGap

Despite out target being the use of various methods of hierarchical clustering, for the sake of having another method of comparison, we use the k-means and the "clusGap" function to compute the optimal K with $K = 1, ..., 10$.

```
gap <- function(data,FUNcluster=kmeans, K.max=10, B = 50, d.power = 2,
              spaceH0 ="scaledPCA",method ="globalSEmax", SE.factor = 2,...){

          gap1 <- clusGap(data,kmeans,K.max, B, d.power,spaceH0,...)

          nc <- maxSE(gap1$Tab[,3],gap1$Tab[,4],method, SE.factor)

          kmopt <- kmeans(data,nc,...)
          out <- list()
          out$gapout <- gap1
          out$nc <- nc
          out$kmopt <- kmopt
          out
}

gap_covid <- gap(covid2021cl,nstart = 100)
gap_covid$nc
```

```
## [1] 10
```

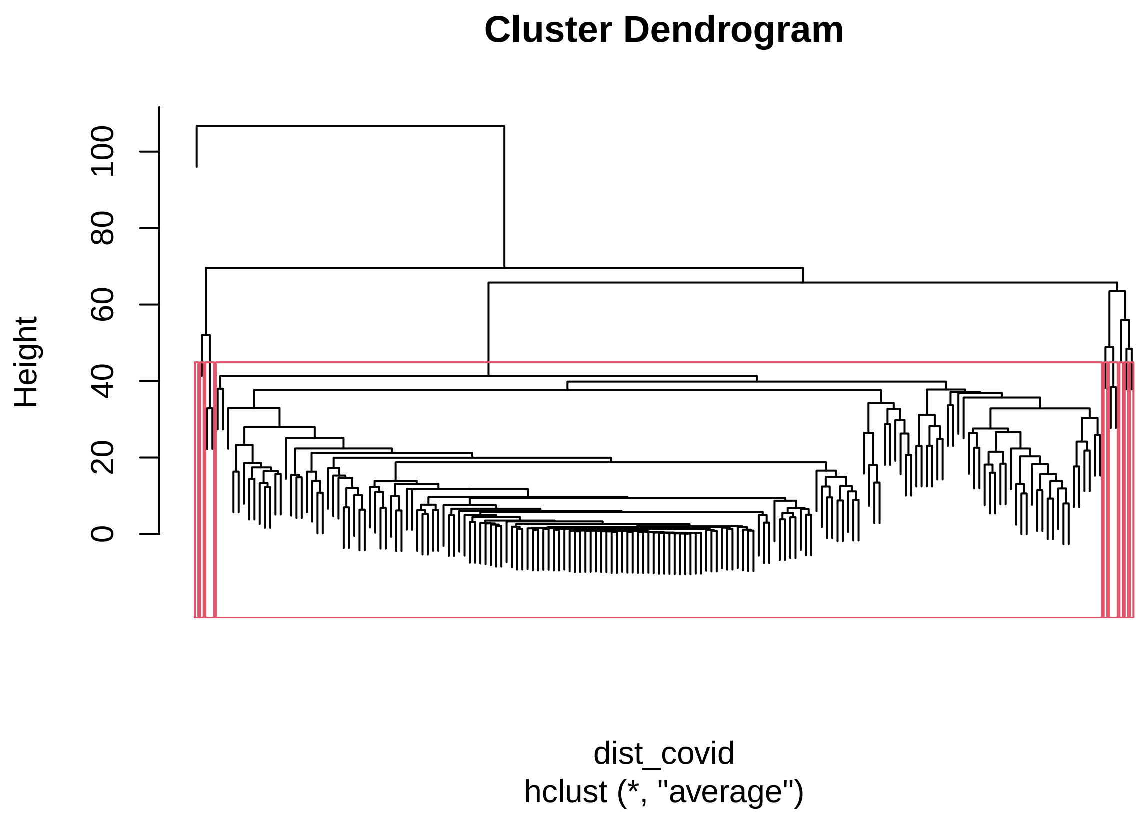
The suggested K is 9. Let us see what we get with the same K in hierarchical clustering.

Agglomerative hierarchical clustering with Euclidean distance

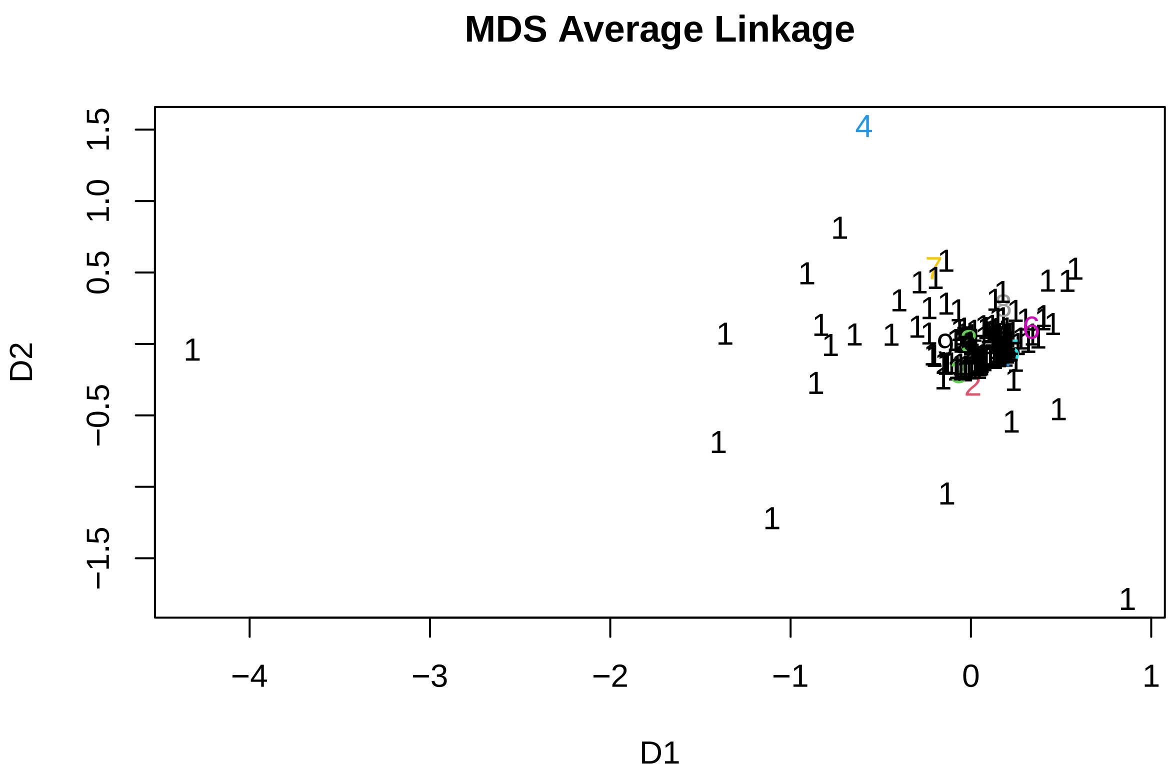
We want to create a matrix with Euclidean distances and use it as input for the three types of linkage discussed in class (average, single, complete).

Since the clusters are multi-dimensional, we use MDS to plot the clusters in two dimensions.

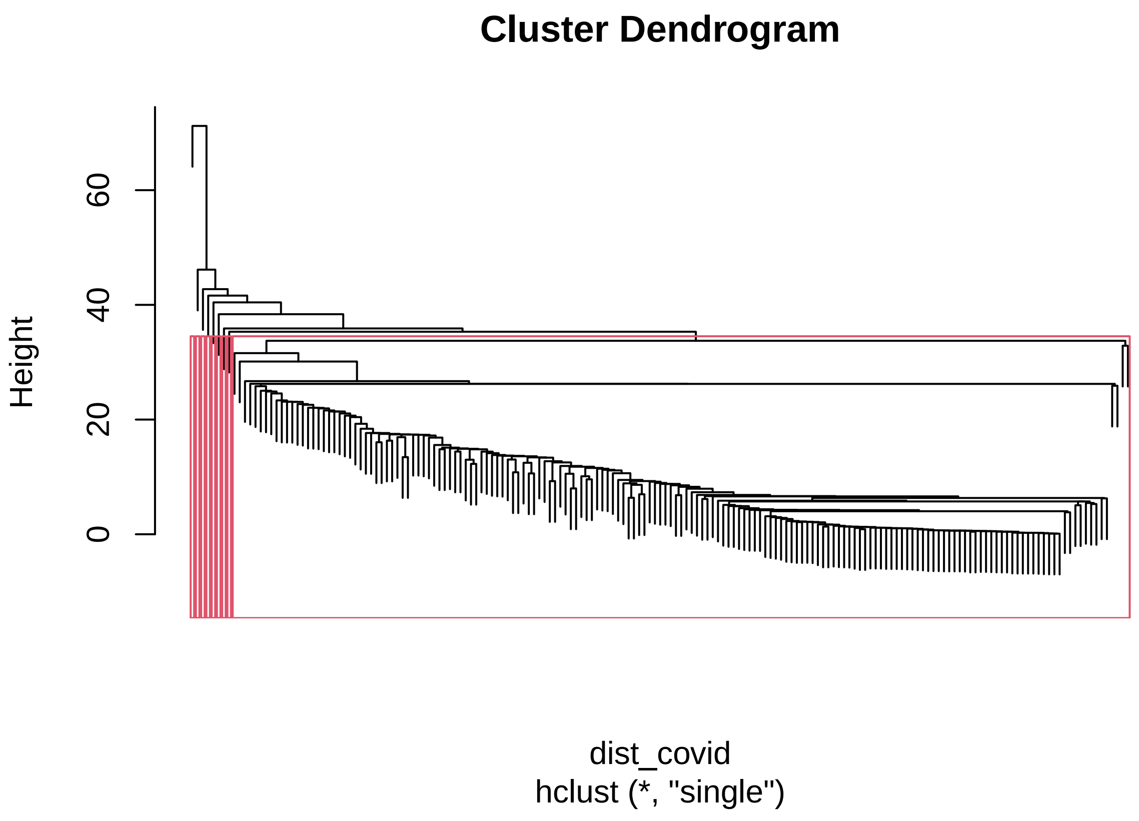
```
mds_covid <- mds(covid2021cl)
dist_covid <- dist(covid2021cl, method = "euclidean")

# Average Linkage
covid_clust <- hclust(dist_covid,method="average")
# Plot the dendrogram:
plot(covid_clust, labels = FALSE)

rect.hclust(covid_clust, k = 9)
```



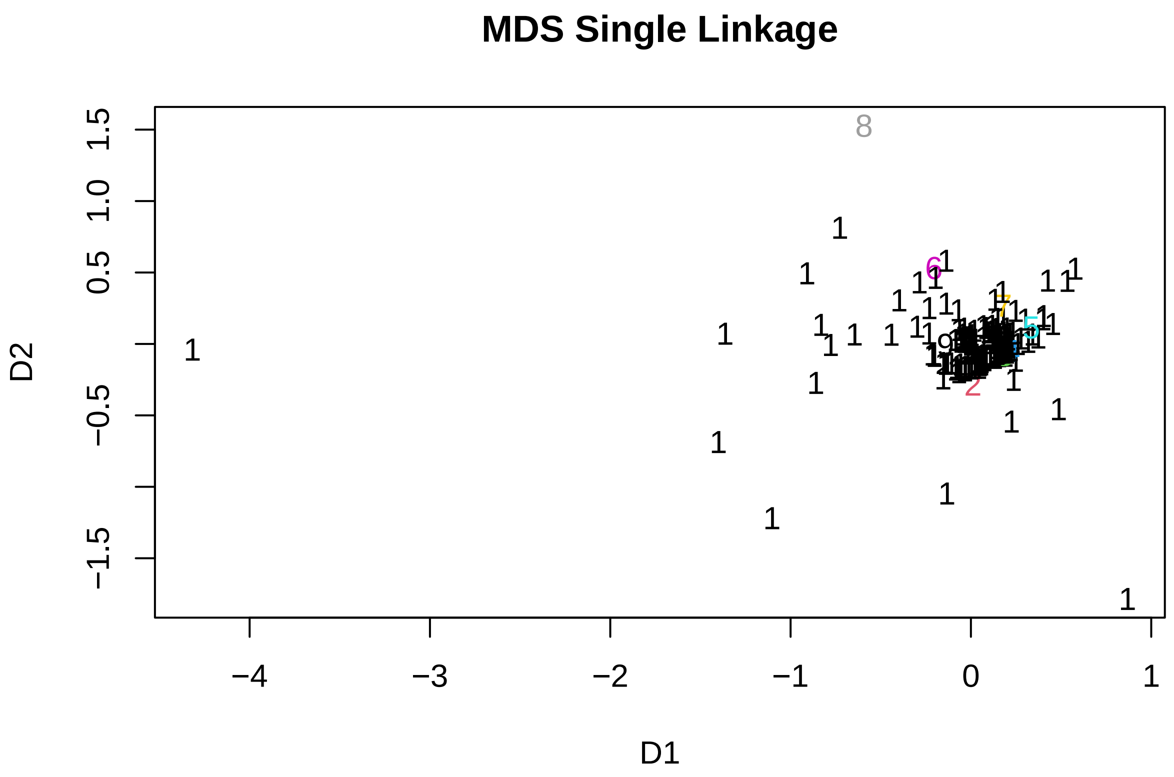
```
covid_clust9 <- cutree(covid_clust,9)
# covid_clust9 to see which cluster number the countries are associated to

# Visualisation using MDS:
plot(mds_covid$conf,col=covid_clust9,pch=clusym[covid_clust9])
title(main = "MDS Average Linkage")
```



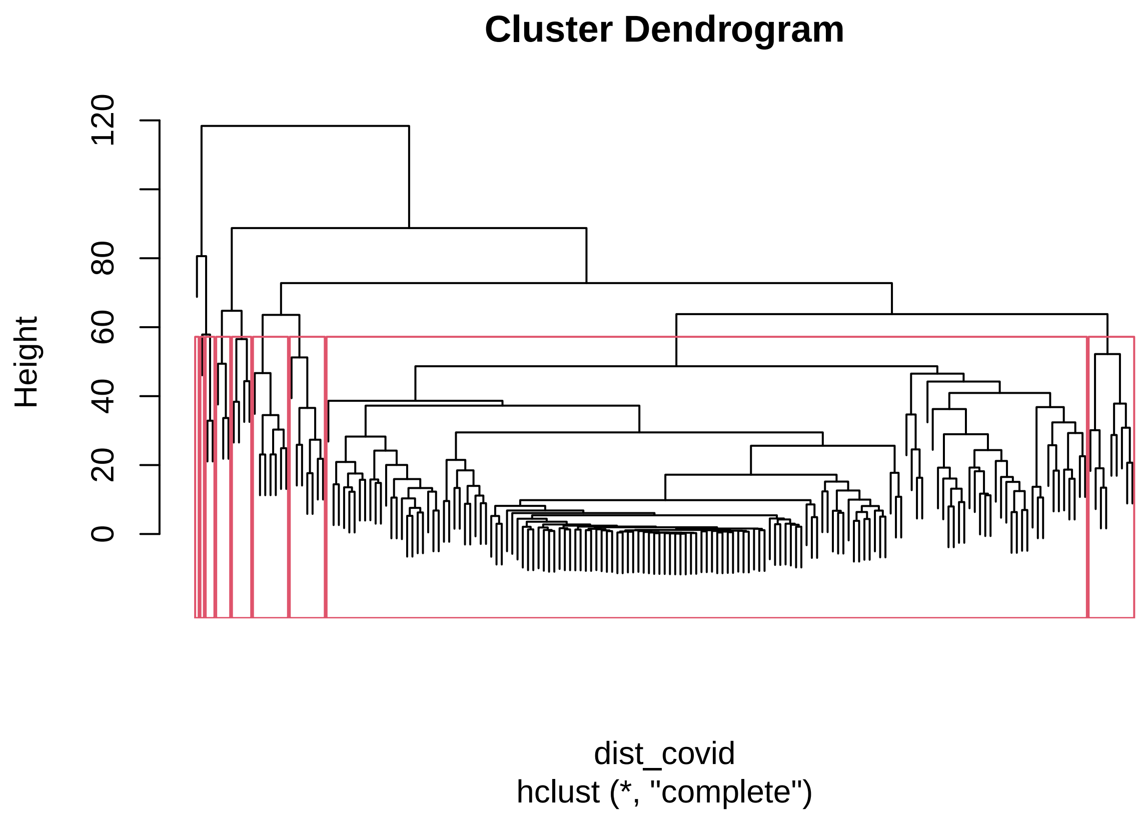
```
# Single Linkage
covid_clusts <- hclust(dist_covid, method="single")
plot(covid_clusts, labels=FALSE)
rect.hclust(covid_clusts, k = 9)
```



```r
covid_clusts9 <- cutree(covid_clusts,9)
plot(mds_covid$conf,col=covid_clusts9,pch=clusym[covid_clusts9])
title(main = "MDS Single Linkage")
```



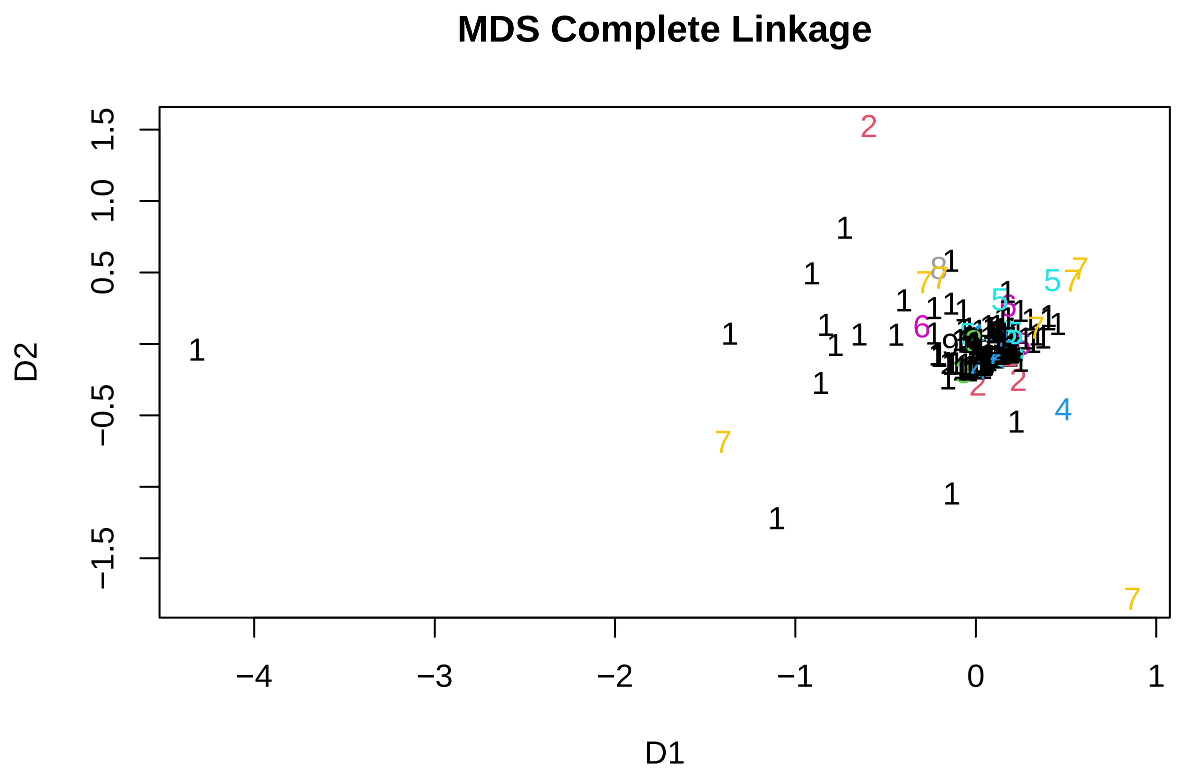
```
# Complete Linkage
covid_clustc <- hclust(dist_covid,method="complete")
plot(covid_clustc, labels=FALSE)
rect.hclust(covid_clustc, k = 9)
```



```r
covid_clustc9 <- cutree(covid_clustc,9)
covid_clustc9
```

```
##                     Afghanistan                         Albania
##                               1                               1
##                         Algeria                         Andorra
##                               1                               2
##                          Angola             Antigua and Barbuda
##                               1                               1
##                       Argentina                         Armenia
##                               1                               1
##                         Austria                      Azerbaijan
##                               1                               1
##                         Bahamas                         Bahrain
##                               1                               3
##                      Bangladesh                        Barbados
##                               1                               1
##                         Belarus                         Belgium
##                               1                               2
##                          Belize                           Benin
##                               1                               1
##                          Bhutan                         Bolivia
##                               1                               1
##          Bosnia and Herzegovina                        Botswana
##                               1                               4
##                           Brazil                           Brunei
##                                1                                1
##                         Bulgaria                     Burkina Faso
##                                1                                1
##                            Burma                          Burundi
##                                1                                1
##                       Cabo Verde                         Cambodia
##                                1                                1
##                         Cameroon                             Chad
##                                1                                1
##                            Chile                         Colombia
##                                1                                1
##                          Comoros              Congo (Brazzaville)
##                                1                                1
##                 Congo (Kinshasa)                       Costa Rica
##                                1                                1
##                    Cote d’Ivoire                          Croatia
##                                1                                5
##                             Cuba                           Cyprus
##                                4                                1
##                          Czechia                          Denmark
##                                2                                1
##                         Djibouti                         Dominica
##                                1                                4
##               Dominican Republic                          Ecuador
##                                1                                1
##                            Egypt                      El Salvador
##                                1                                1
##                Equatorial Guinea                          Eritrea
##                                1                                1
##                          Estonia                         Eswatini
##                                6                                1
##                         Ethiopia                             Fiji
##                                1                                4
##                          Finland                           France
##                                1                                1
##                            Gabon                           Gambia
##                                1                                1
##                          Georgia                          Germany
##                                5                                1
##                            Ghana                           Greece
##                                1                                1
##                        Guatemala                           Guinea
##                                1                                1
##                    Guinea-Bissau                           Guyana
##                                1                                1
##                            Haiti                         Honduras
##                                1                                1
##                          Hungary                          Iceland
##                                1                                1
##                            India                        Indonesia
##                                1                                1
##                             Iran                             Iraq
##                                1                                1
##                Ireland                 Israel
##                      7                      7
##                  Italy                Jamaica
##                      1                      1
##                  Japan                 Jordan
##                      1                      1
##             Kazakhstan                  Kenya
##                      1                      1
##           Korea, South                 Kosovo
##                      1                      1
##                 Kuwait             Kyrgyzstan
##                      1                      1
##                   Laos                 Latvia
##                      1                      1
##                Lebanon                Liberia
##                      1                      1
##                  Libya          Liechtenstein
##                      1                      5
##              Lithuania             Luxembourg
##                      5                      5
##             Madagascar                 Malawi
##                      1                      1
##               Malaysia               Maldives
##                      4                      8
##                   Mali                  Malta
##                      1                      1
##             Mauritania              Mauritius
##                      1                      1
##                 Mexico                Moldova
##                      1                      1
##                 Monaco               Mongolia
##                      1                      4
##             Montenegro                Morocco
##                      6                      1
##             Mozambique                Namibia
##                      1                      1
##                  Nepal            Netherlands
##                      1                      1
##            New Zealand              Nicaragua
##                      1                      1
##                  Niger                Nigeria
##                      1                      1
##        North Macedonia                 Norway
##                      1                      1
##                   Oman               Pakistan
##                      1                      1
##                 Panama       Papua New Guinea
##                      7                      1
##               Paraguay                   Peru
##                      1                      1
##            Philippines                 Poland
##                      1                      1
##               Portugal                  Qatar
##                      7                      1
##                          Romania                           Russia 
##                                1                                1 
##                           Rwanda            Saint Kitts and Nevis 
##                                1                                4 
##                      Saint Lucia Saint Vincent and the Grenadines 
##                                4                                1 
##                       San Marino            Sao Tome and Principe 
##                                2                                1 
##                     Saudi Arabia                          Senegal 
##                                1                                1 
##                           Serbia                       Seychelles 
##                                6                                9 
##                     Sierra Leone                        Singapore 
##                                1                                1 
##                         Slovakia                         Slovenia 
##                                1                                5 
##                          Somalia                     South Africa 
##                                1                                1 
##                      South Sudan                            Spain 
##                                1                                7 
##                        Sri Lanka                            Sudan 
##                                1                                1 
##                         Suriname                           Sweden 
##                                4                                1 
##                      Switzerland                            Syria 
##                                5                                1 
##                       Tajikistan                         Thailand 
##                                1                                1 
##                      Timor-Leste                             Togo 
##                                1                                1 
##              Trinidad and Tobago                          Tunisia 
##                                1                                1 
##                           Turkey                               US 
##                                1                                7 
##                           Uganda                          Ukraine 
##                                1                                1 
##             United Arab Emirates                   United Kingdom 
##                                1                                7 
##                          Uruguay                       Uzbekistan 
##                                3                                1 
##                        Venezuela                          Vietnam 
##                                1                                1 
##               West Bank and Gaza                            Yemen 
##                                1                                1 
##                           Zambia                         Zimbabwe 
##                                1                                1 
##                        Australia                           Canada 
##                                1                                1 
##                            China 
##                                1
```

```r
plot(mds_covid$conf,col=covid_clustc9,pch=clusym[covid_clustc9])
title(main = "MDS Complete Linkage")
```



Remark: we can perform clusterization of the data in two ways:

- we choose a cutoff value of H to split the dendrogram or alternatively:
- we set K so that the dendrogram is split at a certain H to yield the desired number of clusters.

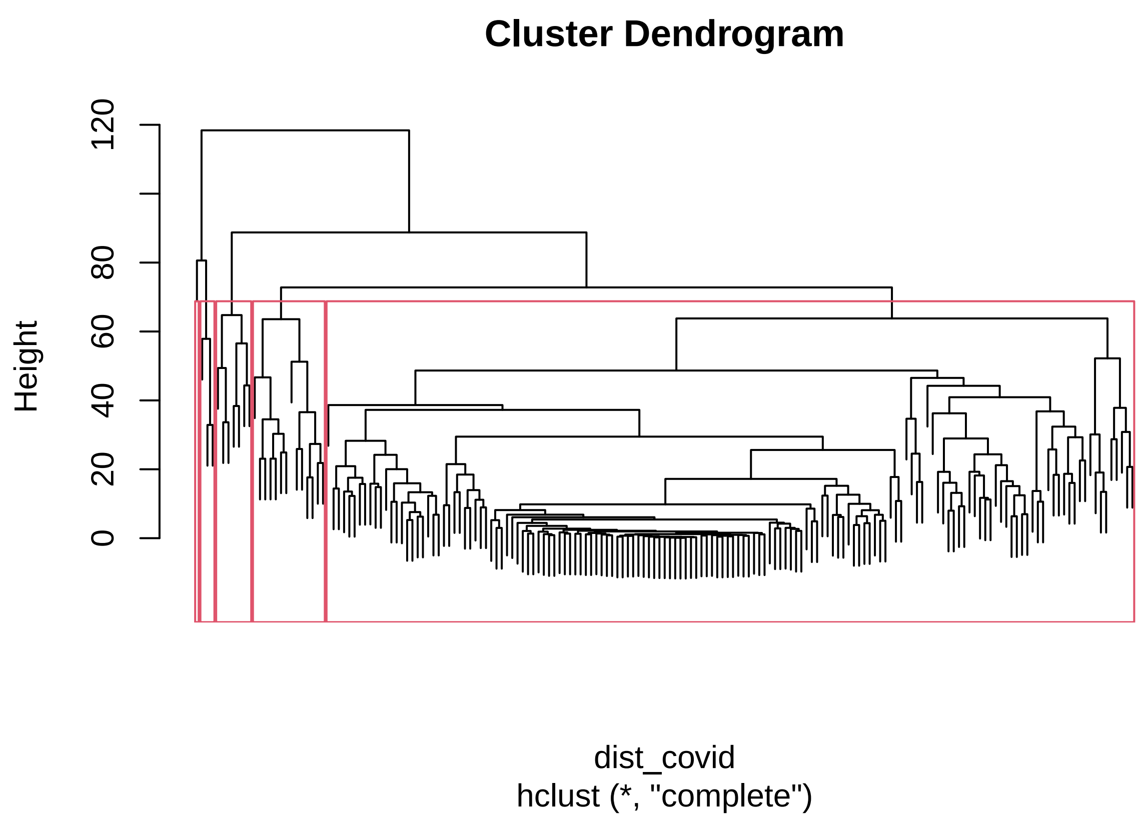
Here we have decided to set K in "rect.clust" as our method.

All the scatterplots are created with multi-dimensional scaling and their visualization must be considered as an approximation. The points in these scatterplots represent the countries of the dataset. As we can notice, the points do not appear to be grouped in some sort of homogeneous clusters, and as a consequence, it is very difficult to perform cluster analysis of any kind. The single linkage seems to perform the worst.

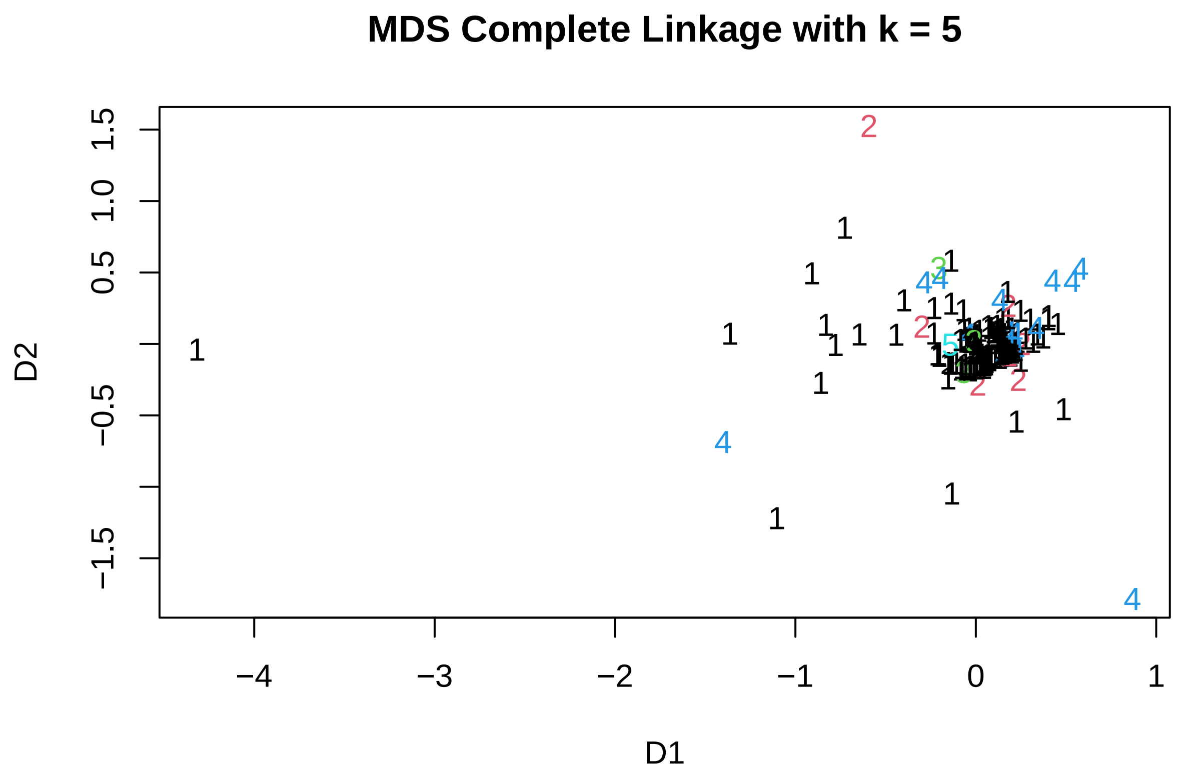
However, the Hierarchical clustering computed with the complete linkage seems to be the best of all, despite most of the observations are still linked to cluster 1 (however it is better because there is only one observation as outlier while in the other linkages there are more of them). With "covid_clustc9", we have printed a table showing the cluster association to the specific country.

Let us try the complete linkage with a smaller K and see if the clusterization is visually better:

```r
# Complete Linkage
plot(covid_clustc, labels=FALSE)
rect.hclust(covid_clustc, k = 5)
```



```r
covid_clustc5 <- cutree(covid_clustc,5)
plot(mds_covid$conf,col=covid_clustc5,pch=clusym[covid_clustc5])
title(main = "MDS Complete Linkage with k = 5")
```



Sadly, there is still one outlier and the cluster 1 is even bigger. These two details make us think that it may be better to stick with $K = 9$.

Agglomerative hierarchical clustering with correlation dissimilarity

We now want to perform clustering analysis with the same three linkage methods but this time we want to use the correlation dissimilarity.

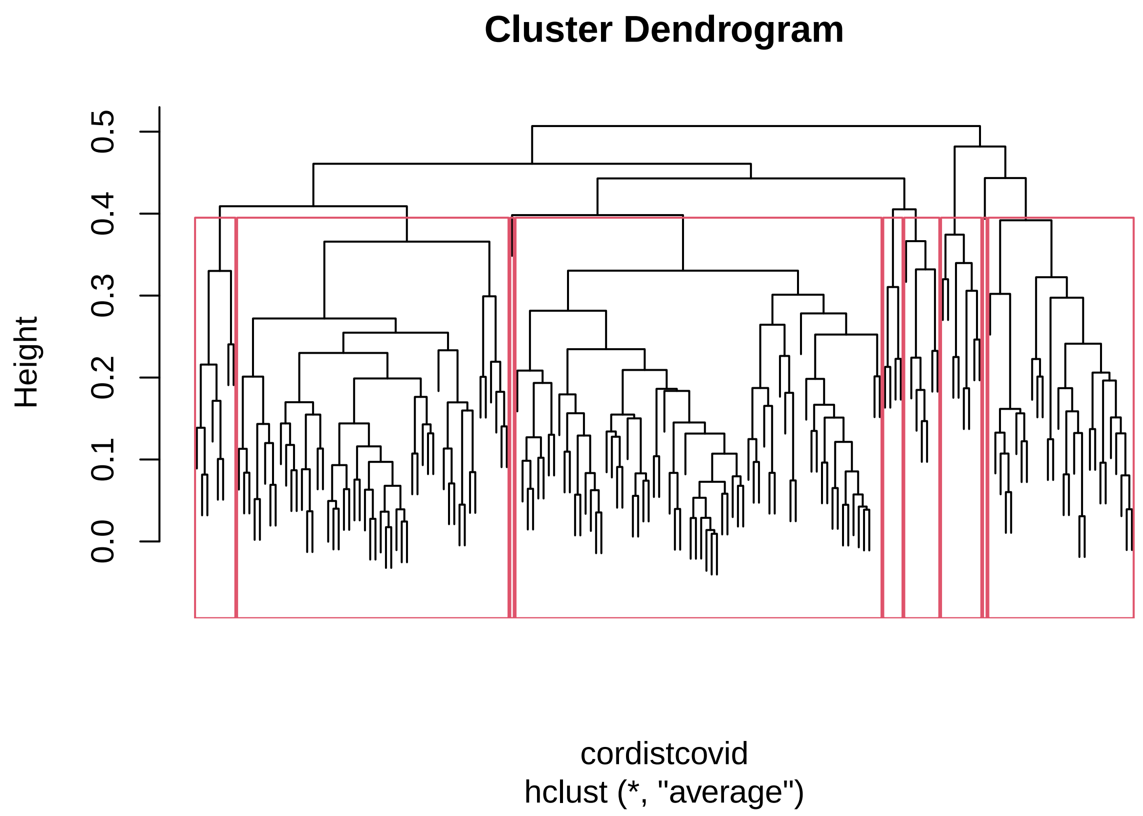
Firs of all, we have to compute the correlation matrix of the transposed dataframe because we want to compute the correlation between countries instead of variables. Then we use the formula of the correlation dissimilarity and we convert it with "as.dist" to use it as argument of hclust. The rest is the same as before.

Remark: we may check the information loss with mds by checking the stress of the correlation dissimilarity. 26% is not bad actually, however, we could improve the result by increasing the dimensionality with ndim.

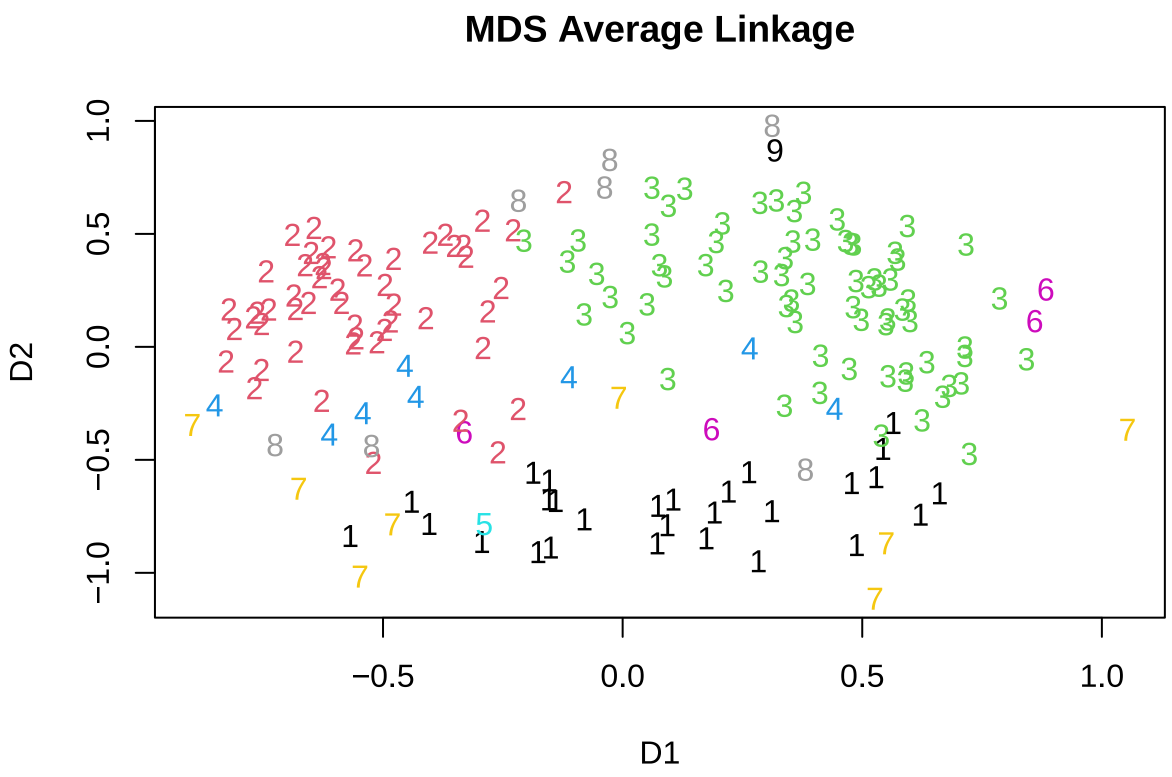
```
corcovid <- cor(t(covid2021cl))


cordistcovid <- 0.5 - corcovid/2 # Correlation dissimilarity
cordistcovid <- as.dist(cordistcovid)


mdscorcovid <- mds(cordistcovid, ndim=2)
# ndim is the number of dimensions here. 2 is default.
conf_mds <- mdscorcovid$conf

mdscorcovid$stress
```

```
## [1] 0.2649387
```

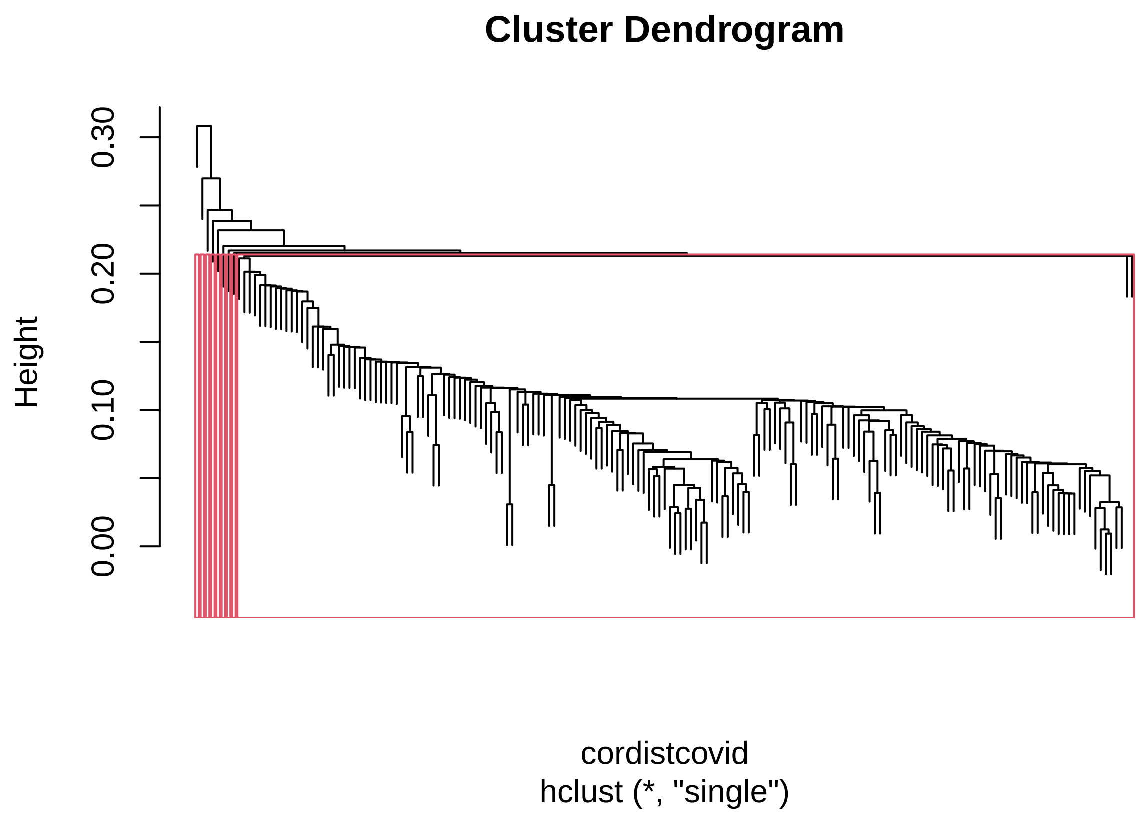
```
# Average Linkage
corcovid_clust <- hclust(cordistcovid,method="average")
# Plot the dendrogram:
plot(corcovid_clust, labels = FALSE)

rect.hclust(corcovid_clust, k = 9)
```



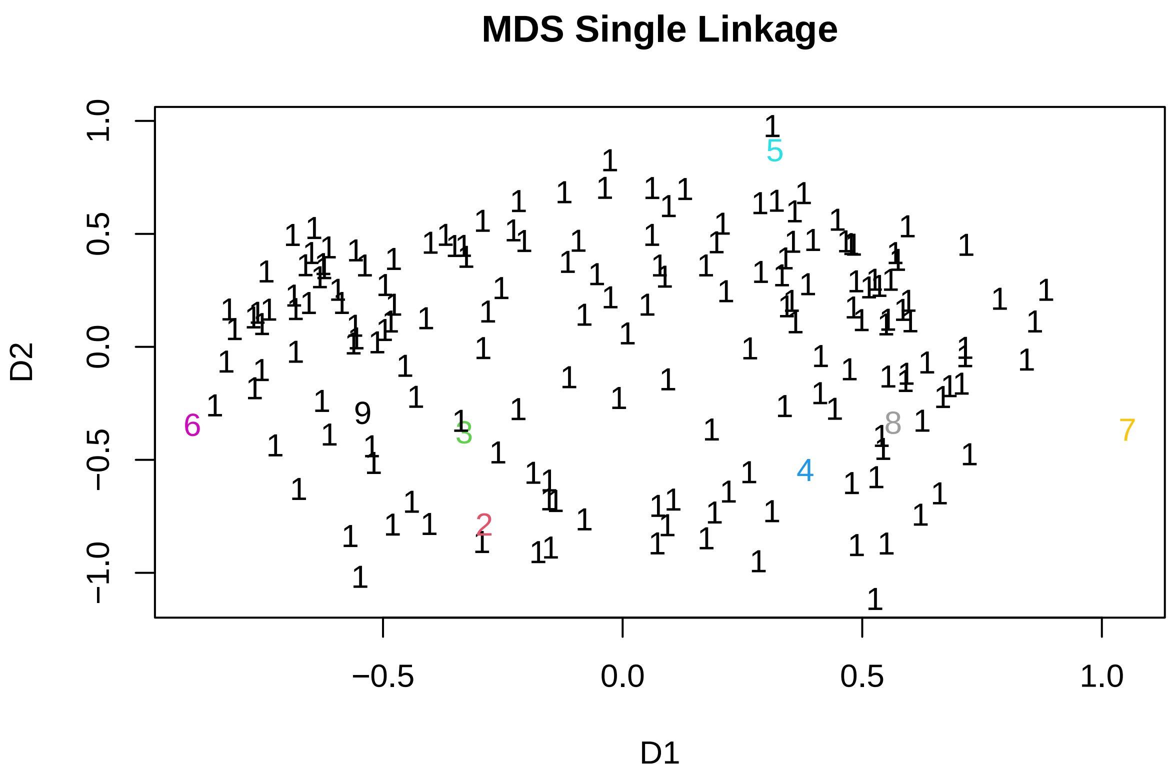
```
corcovid_clust9 <- cutree(corcovid_clust,9)

# Visualisation using MDS:
plot(mdscorcovid$conf,col=corcovid_clust9,pch=clusym[corcovid_clust9])
title(main = "MDS Average Linkage")
```



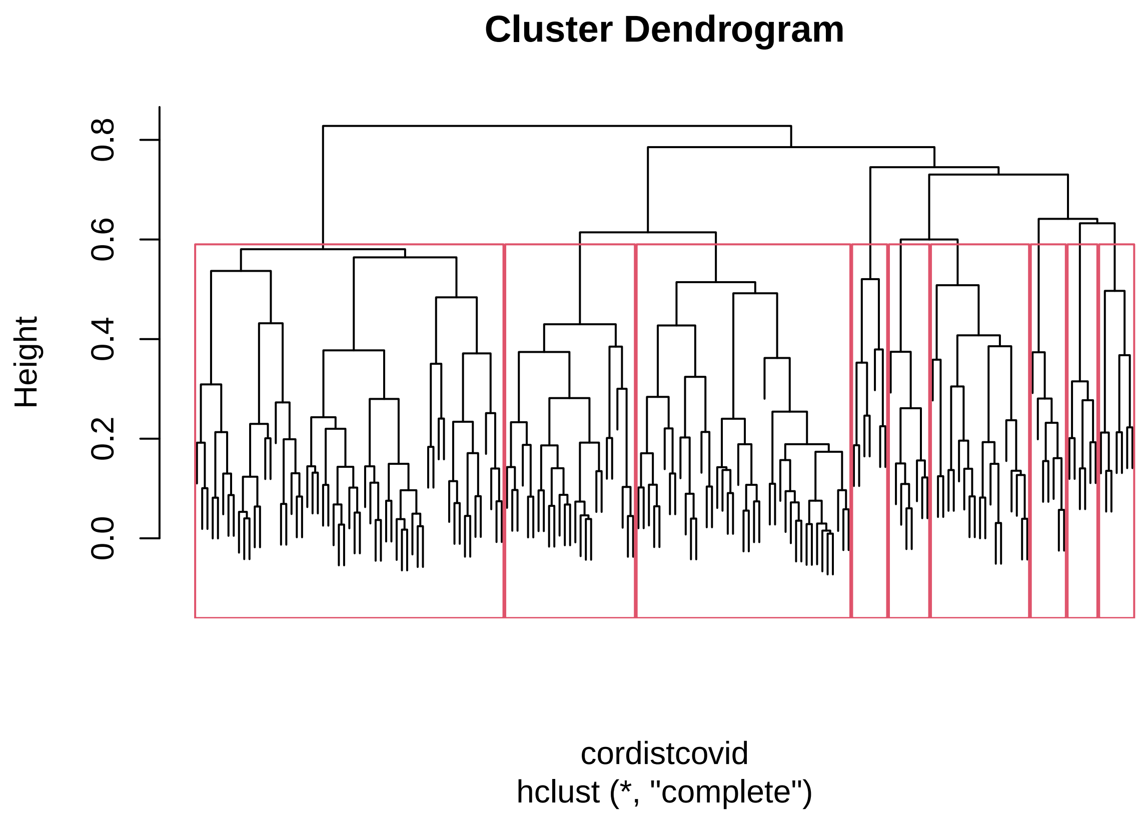
```
# Single Linkage
corcovid_clusts <- hclust(cordistcovid,method="single")
plot(corcovid_clusts, labels=FALSE)
rect.hclust(corcovid_clusts, k = 9)
```



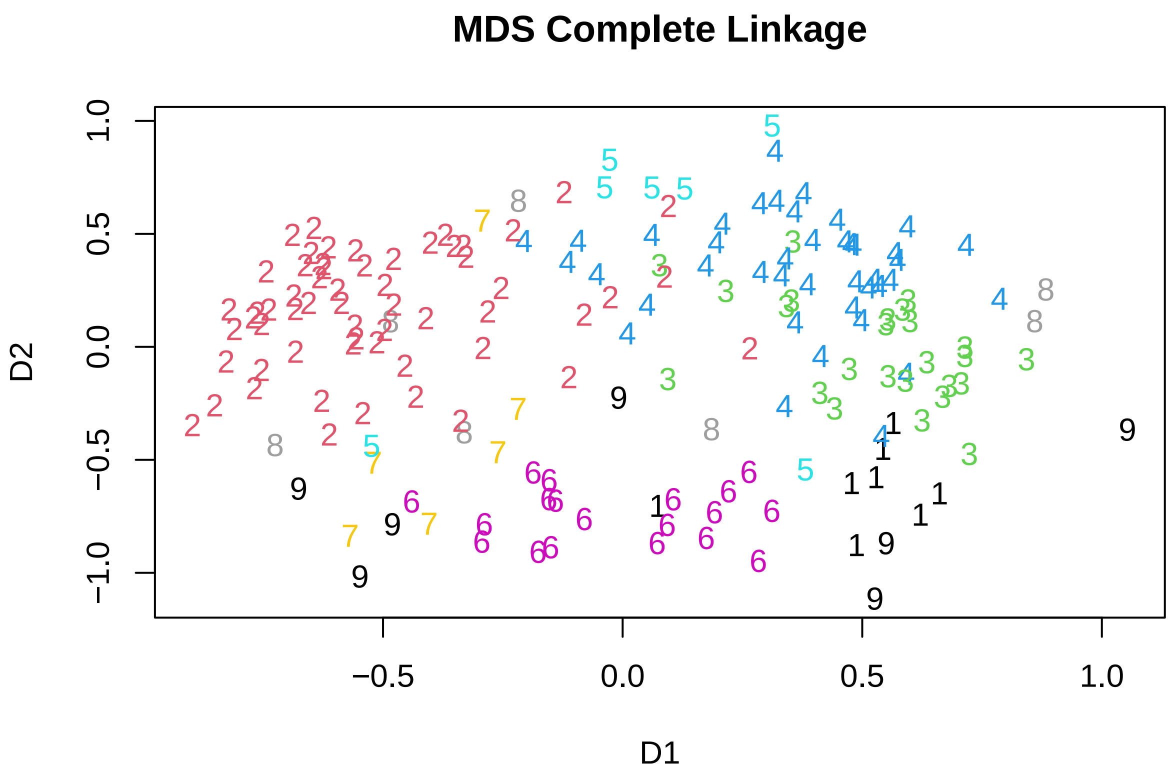
```
corcovid_clusts9 <- cutree(corcovid_clusts,9)

plot(mdscorcovid$conf,col=corcovid_clusts9,pch=clusym[corcovid_clusts9])
title(main = "MDS Single Linkage")
```



```
# Complete Linkage
corcovid_clustc <- hclust(cordistcovid,method="complete")
plot(corcovid_clustc, labels=FALSE)
rect.hclust(corcovid_clustc, k = 9)
```



```
corcovid_clustc9 <- cutree(corcovid_clustc,9)

plot(mdscorcovid$conf,col=corcovid_clustc9,pch=clusym[corcovid_clustc9])
title(main = "MDS Complete Linkage")
```



The single linkage performs poorly. There is one big cluster with many countries and several clusters composed of just single observations (cluster 2,3,4,5,6,7,8). Overall, there is an improvement in the other two linkages with respect to the Euclidean method.

In the average linkage clusters usually contain many countries except only for one outlier (cluster 4).

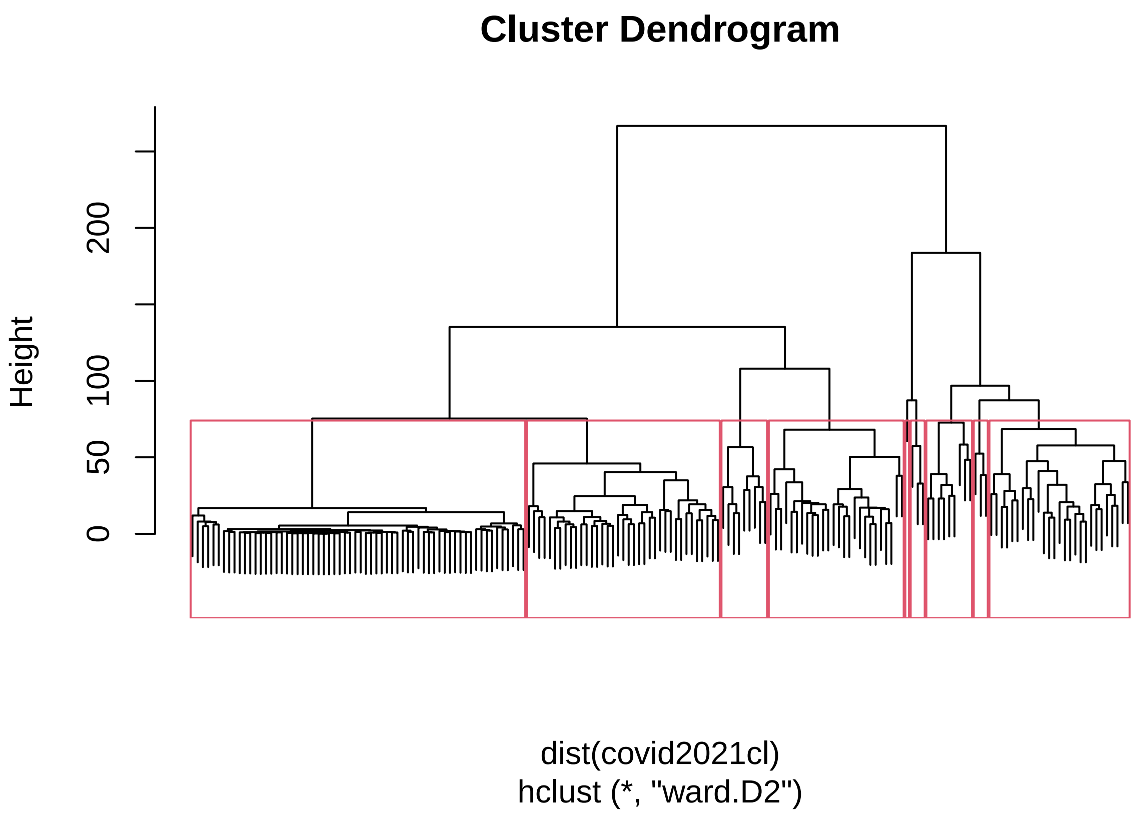
The complete linkage is once again the best without outliers and pretty homogeneous clusters.

Remark: it may be useful to create a heatmap with the correlation matrix between the countries but I have not been able to plot a working correlation heatmap for so many countries.

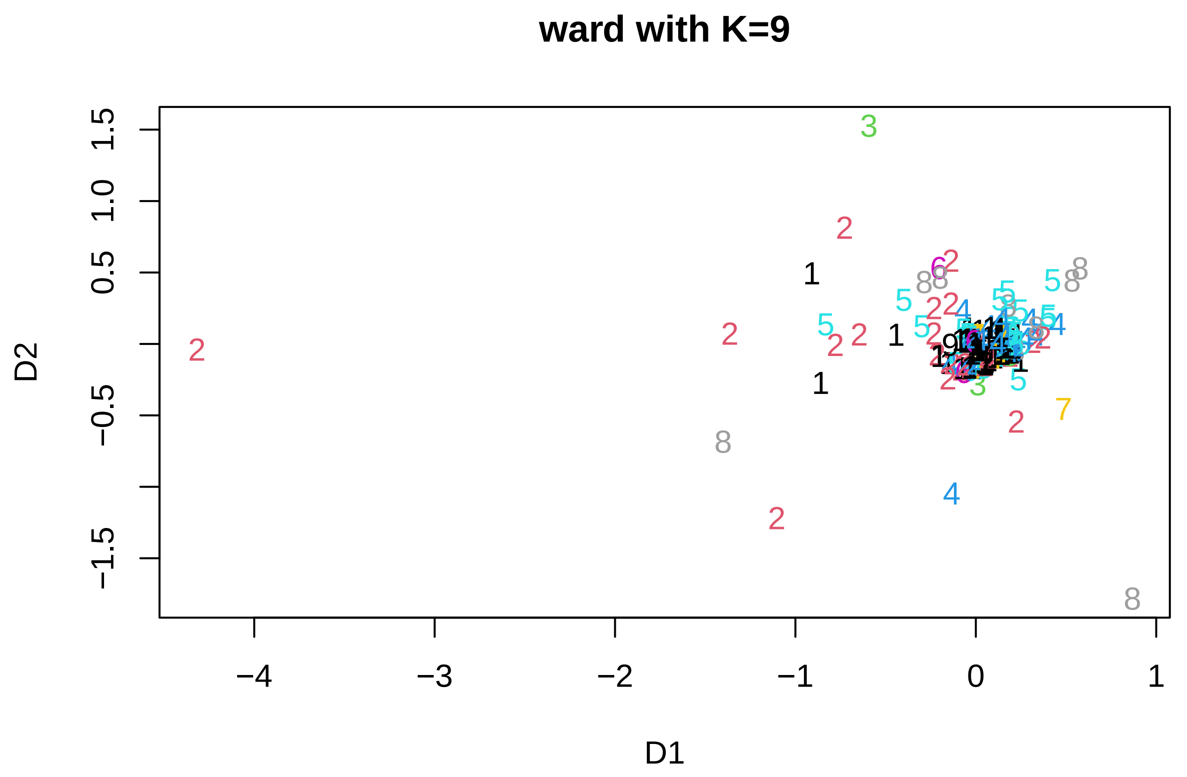
Hierarchical Clustering: Ward's method

As the last form of hierarchical clustering we try Ward's method. We just need to use the "dist" function on the dataset and select method="ward.D2".

```
# Ward's method
ward_covid <- hclust(dist(covid2021cl),method="ward.D2")
plot(ward_covid, labels=FALSE)
rect.hclust(ward_covid, k= 9)
```



```
ward9_covid <- cutree(ward_covid,9)
plot(mds_covid$conf,col=ward9_covid,pch=clusym[ward9_covid])
title(main="ward with K=9")
```



The scatterplot created using MDS is pretty unclear. As before, there is a peak in density around the coordinates (0,0). That makes the patterns of the clusters not visible from a simple look. If we check the dendrogram, we notice that clusters are fairly decent. There is just one outlier that is not visible in the scatterplot and without the notation on the dendrogram it is difficult to determine what country is the outlier. Anyway, the other clusters seems to not be too big or too small.

Let us compare some of the best clusterings using the Adjusted Rand Index to see the similarity between these methods.

```
adj.rand.index(corcovid_clustc9, ward9_covid)
```

```
## [1] 0.09596396
```

```
adj.rand.index(corcovid_clustc9, covid_clustc9)
```

```
## [1] -0.04427331
```

Since the Adjusted Rand Index has values between -1 and 1, 0.0959 represents a neutral level of similarity between complete linkgage with correlation dissimilarity and Ward's clustering. We could say the same thing if we compare the former with the complete linkage with euclidean distance, since - 0.0443 is close to 0.

Conclusion: Overall the cluster analysis of the data is underwhelming. The reason behind it can be linked to the fact that the number of variables is much bigger than the number of observations. A solution to this problem is to use one of the dimensionality reduction methods commonly used in Statistics. For instance, we could apply PCA on the dataset, or we could simply build a correlation matrix and delete the variables with a correlation coefficient whose absolute value is higher than a certain threshold. We have indeed computed a correlation matrix on the transposed dataset, which helped to reduce this issue and in fact it led to a better clustering. After the dimensionality reduction, we could perform the cluster analysis with the same methods, hoping for a solid improvement.