Assignment 3

Sebastian Veuskens

Exercise 1

We divide the variables into three different groups that require different distance measures.

- For the first variable, simple matching is used.
- For the variables 2 to 4 Jaccard distance is used.
- For the 5th variable, (standard) Manhatten distance is used. The maximum of all values of that variable is used for standardization (parameter s).

The weights *w_l* from the formula are not calculated directly, but are indirectly included in the function.

```
v_1 = c("blue", "red", "red", "green")
v_2 = c(1, 0, 1, 1)
v_3 = c(1, 0, 0, 0)
v_4 = c(0, NA, NA, 0)
v_5 = c(12, NA, 17, 21)

data = data.frame(v_1, v_2, v_3, v_4, v_5)

gower = function(x, y, s) {
    numerator = 0
    denominator = 0
    for (k in 1:length(x)) {
        if (is.na(x[k]) || is.na(y[k])) {
            next
        } else if (k == 1) {
            numerator = numerator + ifelse(x[k] == y[k], 0, 1)
            denominator = denominator + 1
        } else if (k %in% 2:4) {
            numerator = numerator + ifelse(x[k] == y[k], 0, 1)
            denominator = denominator + ifelse(x[k] == 1 || y[k] == 1, 1, 0)
        } else {
            numerator = numerator + abs(x[k] - y[k]) / s
            denominator = denominator + 1
        }
    }
    return(as.numeric(numerator / denominator))
}

s = max(data[,5], na.rm = T) - min(data[,5], na.rm = T)

dist_mat = matrix(, 4, 4)
for (row in 1:4) {
    for (col in 1:4) {
        dist_mat[row, col] = gower(data[row,], data[col,], s)
    }
}

print(dist_mat)
```

```
##           [,1] [,2]      [,3]      [,4]
## [1,] 0.0000000  1.0 0.6388889 0.7500000
## [2,] 1.0000000  0.0 0.5000000 1.0000000
## [3,] 0.6388889  0.5 0.0000000 0.4814815
## [4,] 0.7500000  1.0 0.4814815 0.0000000
```

We receive the same distance matrix with the daisy package:

```
library(cluster)

data$v_1 = as.factor(data$v_1)
dist_mat = daisy(data, metric="gower", type=list(asymm=c(2, 3, 4)))

print(dist_mat)
```

```
## 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
```

Exercise 2 b

For Exercise 2 a see the handwritten part at the end of the file.

In the following we give two counterexamples for the triangular inequality for gower and correlation dissimilarity.

Gower

For Gower, simple matching is used. The dissimilarity between x and z is bigger than the sum of the dissimilarities between x to y and y to z. This contradicts the triangular inequality and thus the definition of a distance.

```
gower_simple = function(x, y) {
    numerator = 0
    denominator = 0
    for (k in 1:length(x)) {
        if (is.na(x[k]) || is.na(y[k])) next
        numerator = numerator + ifelse(x[k] == y[k], 0, 1)
        denominator = denominator + 1
    }
    return(numerator/denominator)
}

x = c(1, 0, 0, 1)
y = c(NA, NA, 0, 1)
z = c(0, 1, 0, 1)

x_y = gower_simple(x, y)
x_z = gower_simple(x, z)
y_z = gower_simple(y, z)

# Check if the triangular equality is violated
print(x_y + y_z >= x_z)
```

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

Correlation

For the correlation dissimilarity, only complete variables within the observations are considered. The dissimilarity between x and z is bigger than the sum of the dissimilarities between x to y and y to z. This contradicts the triangular inequality and thus the definition of a distance.

```
corr_dist <- function(x, y) {
    return(0.5 * (1 - cor(x, y, use = "complete.obs")))
}

x = c(1, 0, 1, 1)
y = c(1, 0, 7, 0)
z = c(0, 2, 0, -4)

x_y = corr_dist(x, y)
x_z = corr_dist(x, z)
y_z = corr_dist(y, z)

# Check if the triangular equality is violated
print(x_y + y_z >= x_z)
```

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

Exercise 3

3 a

The Jaccard should only be used if the absence (value of 0) is the “normal” value in some sense, and a value of 1 is in some sense rare or special (asymmetric binary variable). This might be the case in our example since option 0 denotes the current state and option 1 denotes a certain change. Depending on the objective of the study, the scientists might be interested especially in cases where people want to change something (which might be the exception, thus we have an asymmetric dataset). In this case, the Jaccard distance should be preferred since it gives more weight to cases where at least one of the answers is 1 (rare answer).

3 b

In the described case, I would prefer the euclidean distance on scaled data, while I think that also the manhattan distance on scaled data would be a suitable choice.

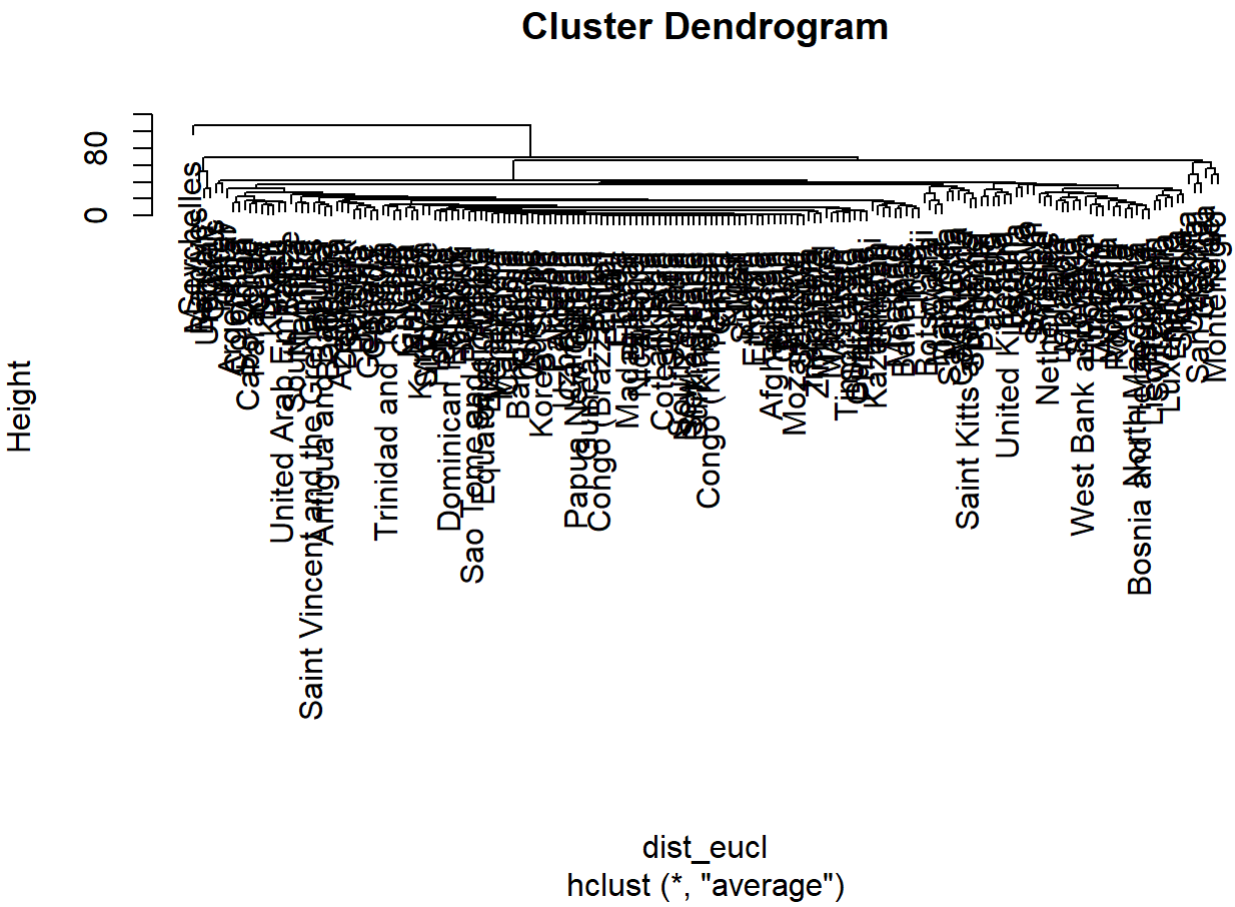
The data should be scaled because the different units of measurements between the variables are not meaningful to us with the information that is available to us. For example, the direct comparison of 10 invested Swiss Francs and 10 incidents in the area is not appropriate.

In addition, Mahalanobi distance downweights the correlated variables. But we might be interested in the correlation between correlated variables, e.g. is there a lot of areas where both the number of incidents and the number of avalanches is very high or do people tend to avoid these areas anyway and thus less incidents are reported there? In addition, the measure “danger” might be multidimensional, e.g. a region could have a lot of injuries (dangerous) but few avalanches (undangerous) or vice versa.

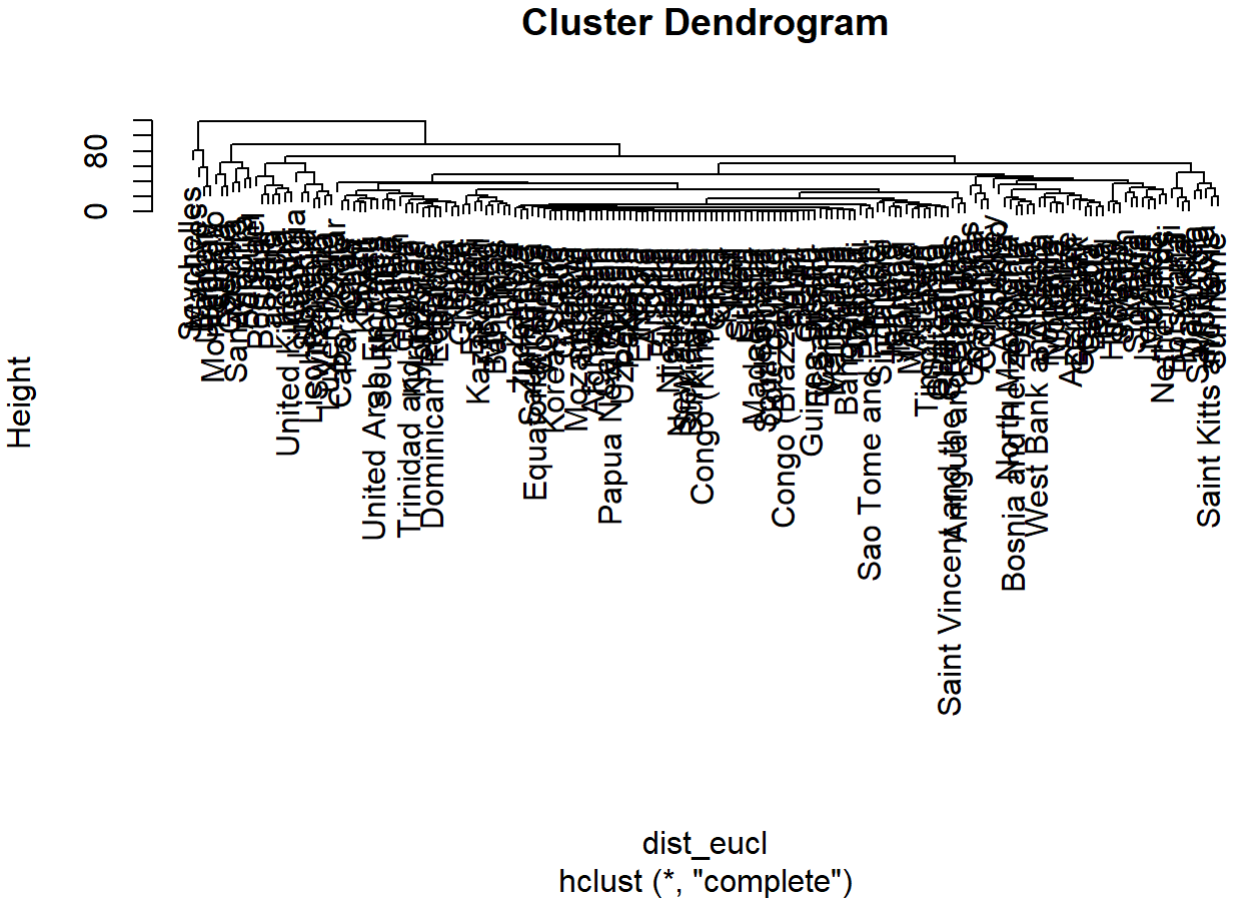
Because of this interest in correlated data and we dont have an imbalanced dataset (or at least dont know if it is), the Mahalanobi distance is not an appropriate choice and the Euclidean distance on scaled data should be used.

Exercise 4

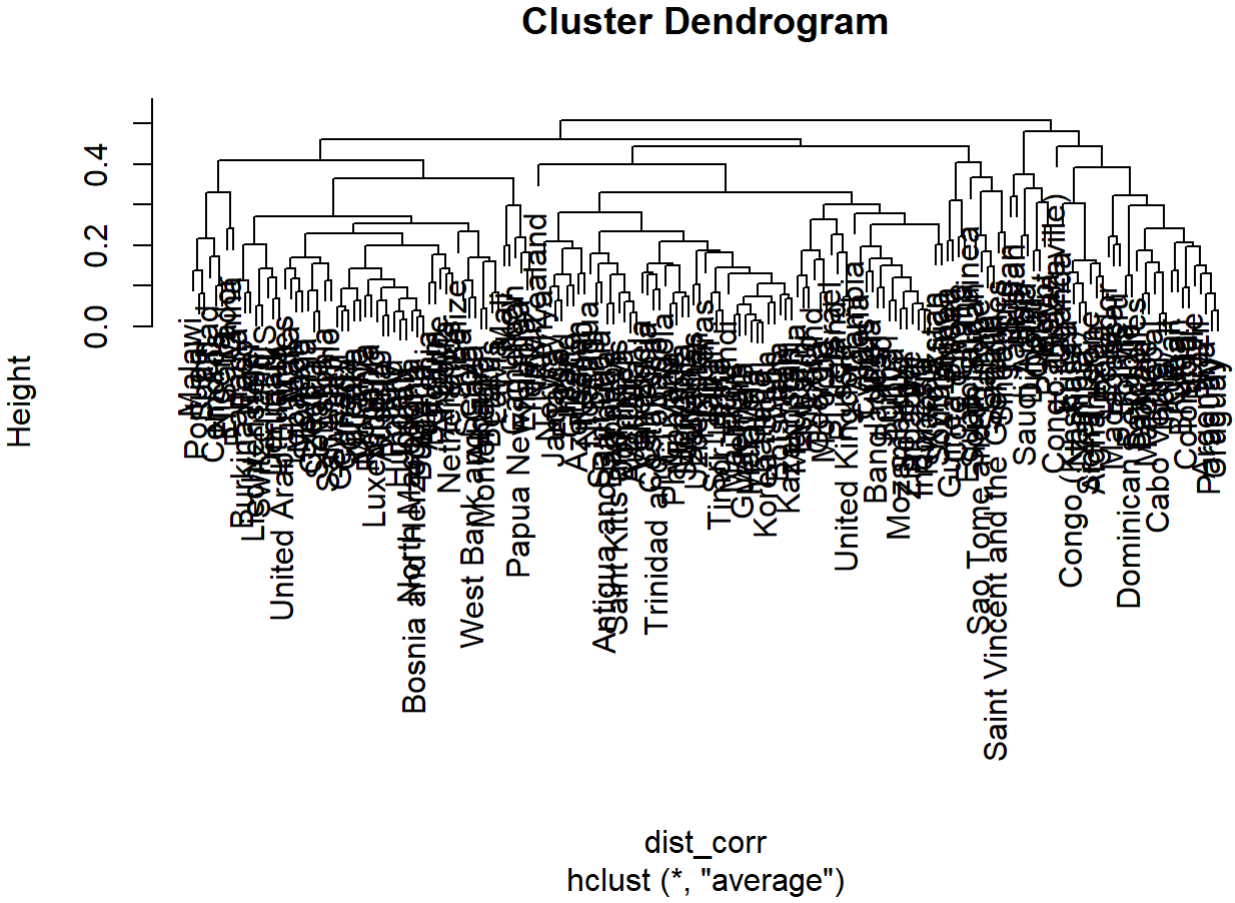
```
library(smacof)

covid2021 <- read.table("data/covid2021.dat")
covid2021cl <- covid2021[,5:559]

#Define the distances objects (euclidean and correlation)
dist_eucl <- dist(covid2021cl, method="euclidean")
dist_corr <- as.dist(0.5 - cor(t(covid2021cl)) / 2)

## Euclidean

eucl_avg <- hclust(dist_eucl, method="average")
eucl_com <- hclust(dist_eucl, method="complete")

## Correlation

corr_avg <- hclust(dist_corr, method="average")
corr_com <- hclust(dist_corr, method="complete")

# Plot the hierarchical clustering results with different dissimilarities and methods
plot(eucl_avg)
```



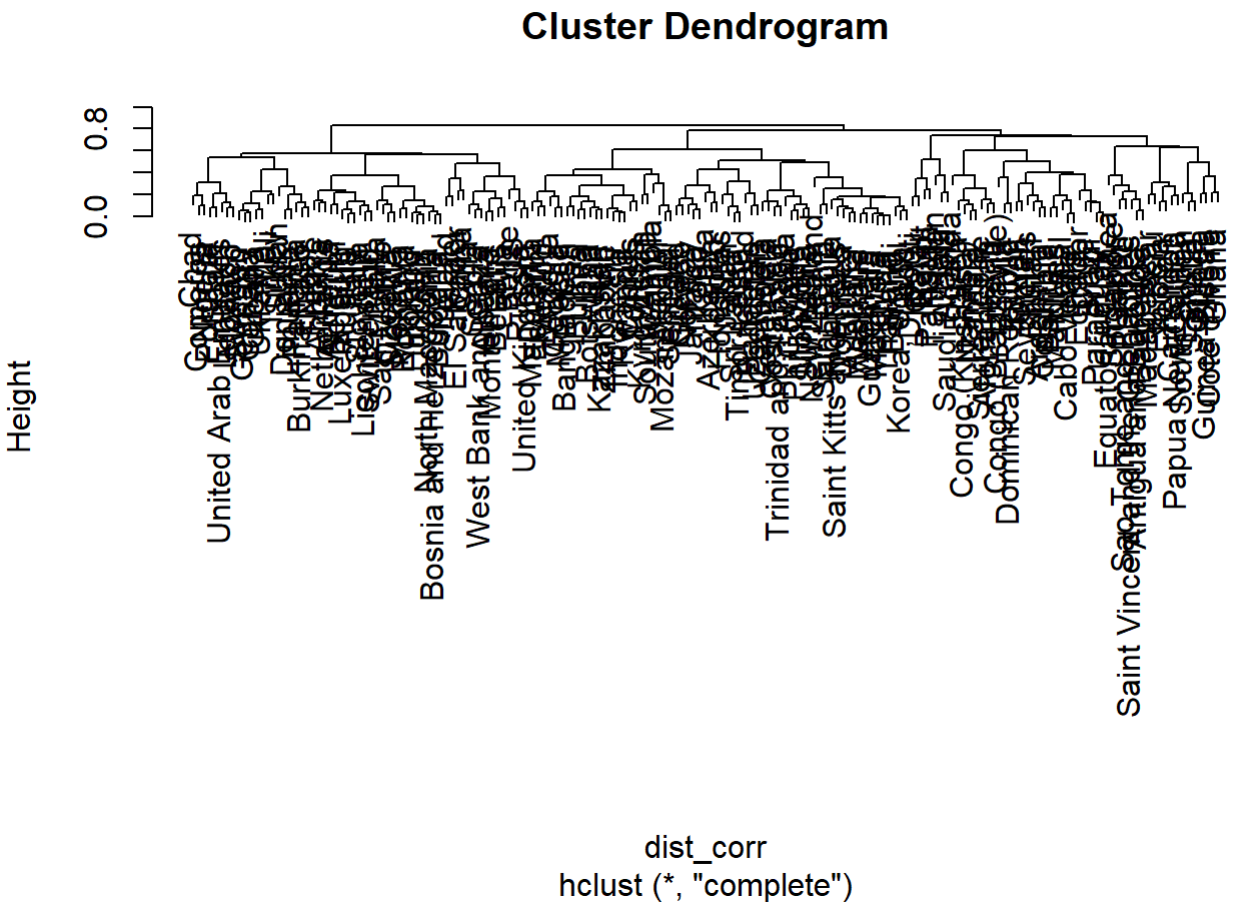

```
plot(eucl_com)
```



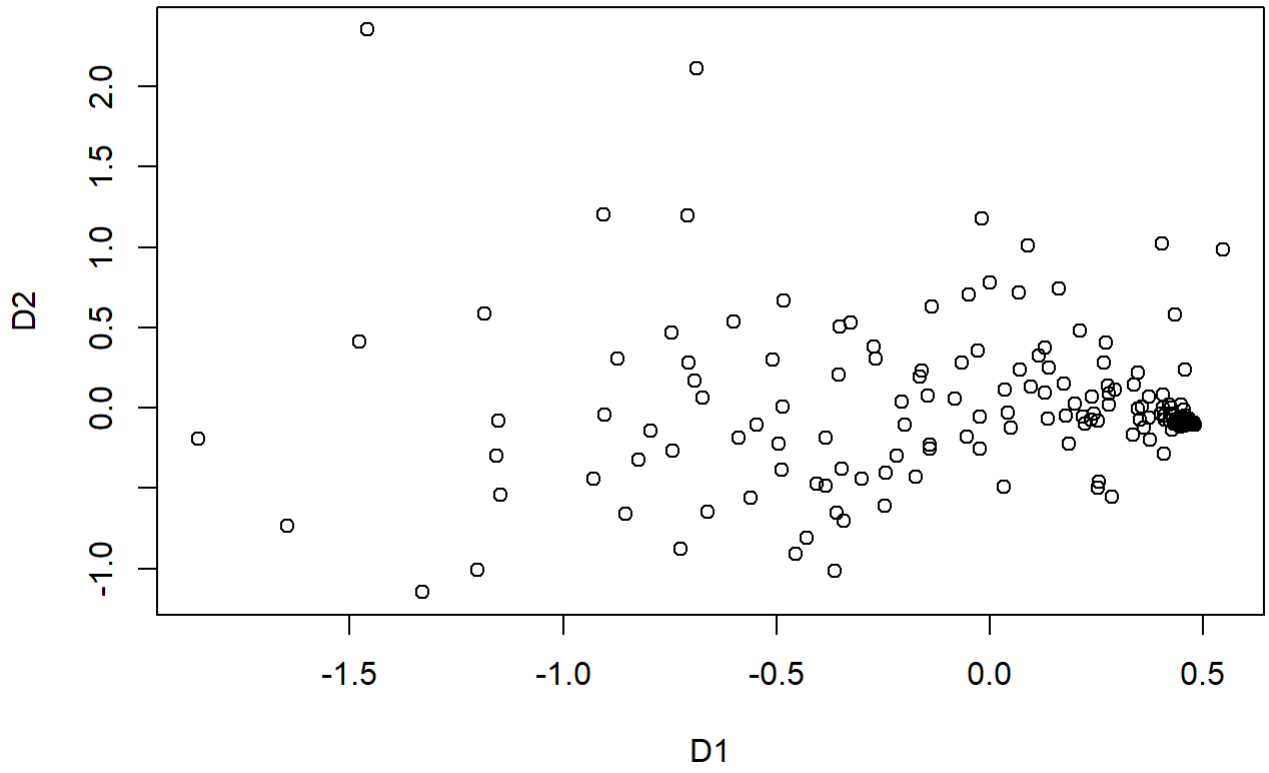
```
plot(corr_avg)
```



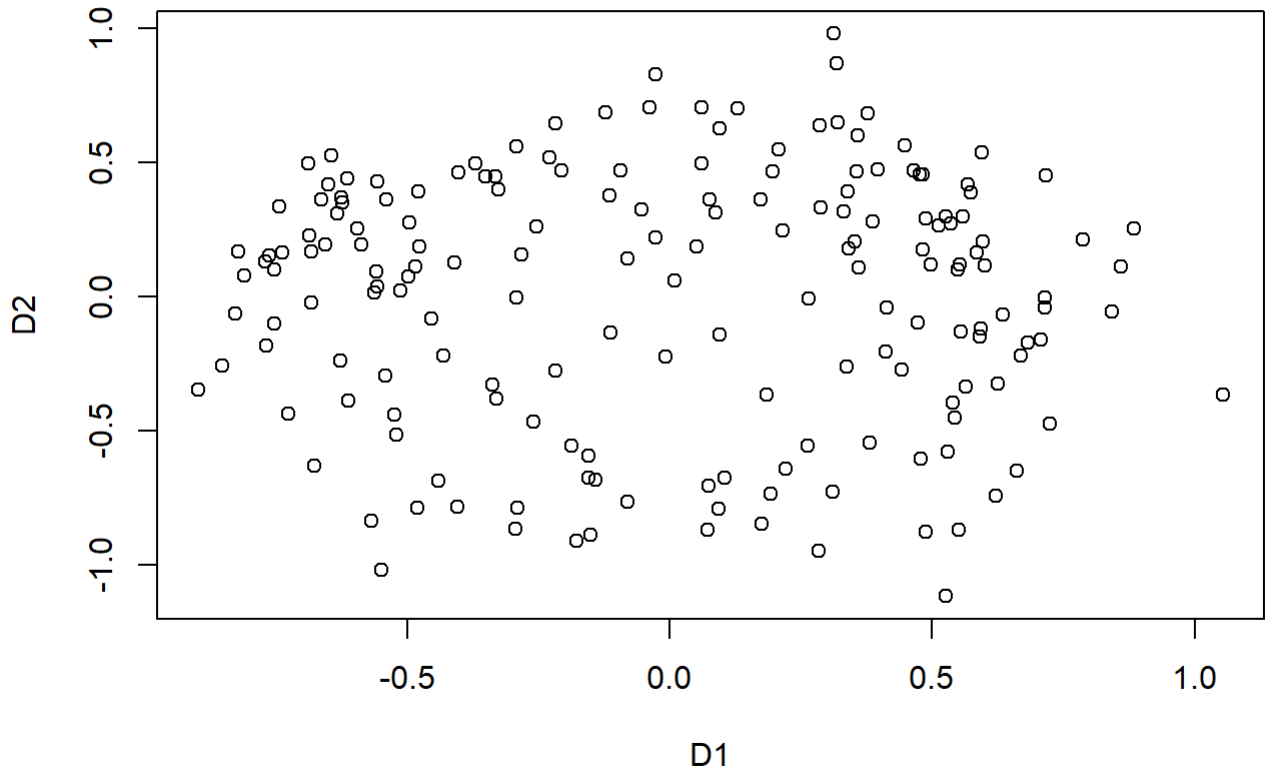

```
plot(corr_com)
```




```
# Plot the 2-dimensional multidimensionally scalled data with different dissimilarities
mds_eucl <- mds(dist_eucl, ndim=2)
mds_corr <- mds(dist_corr, ndim=2)
plot(mds_eucl$conf)
```



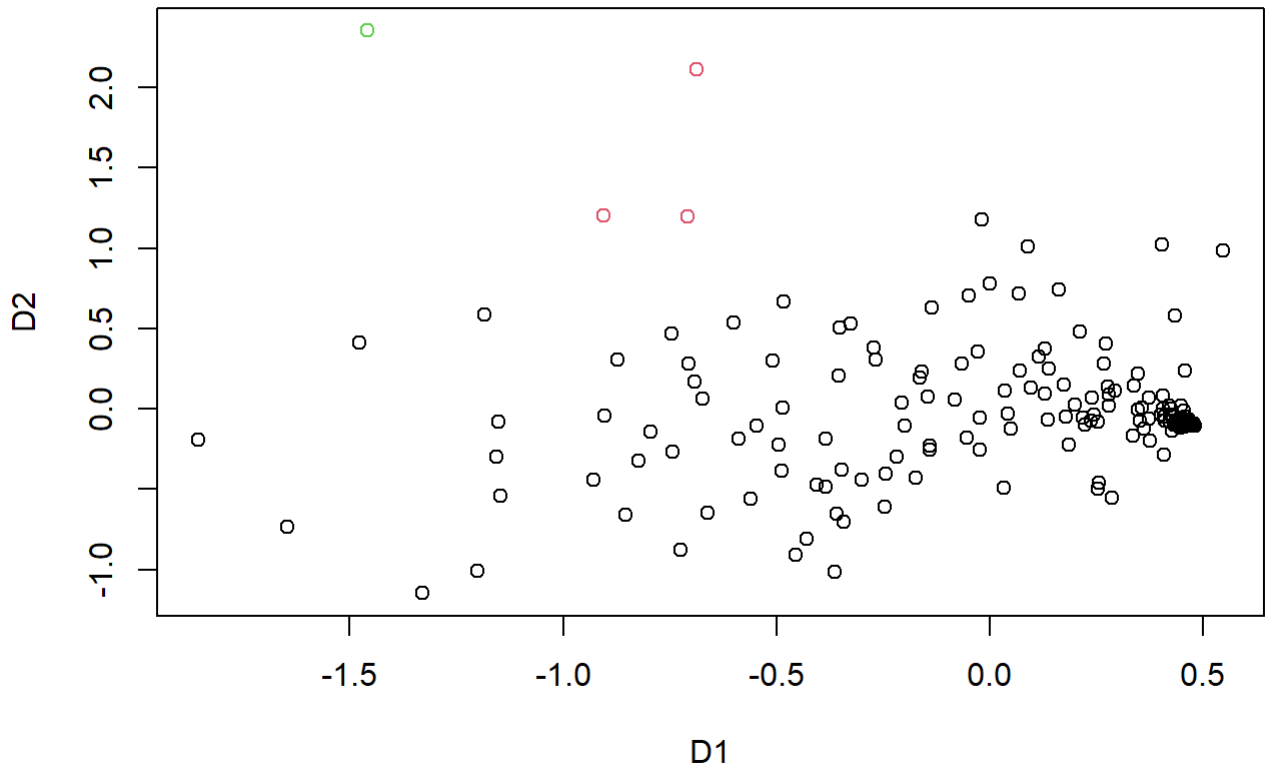

```
plot(mds_corr$conf)
```



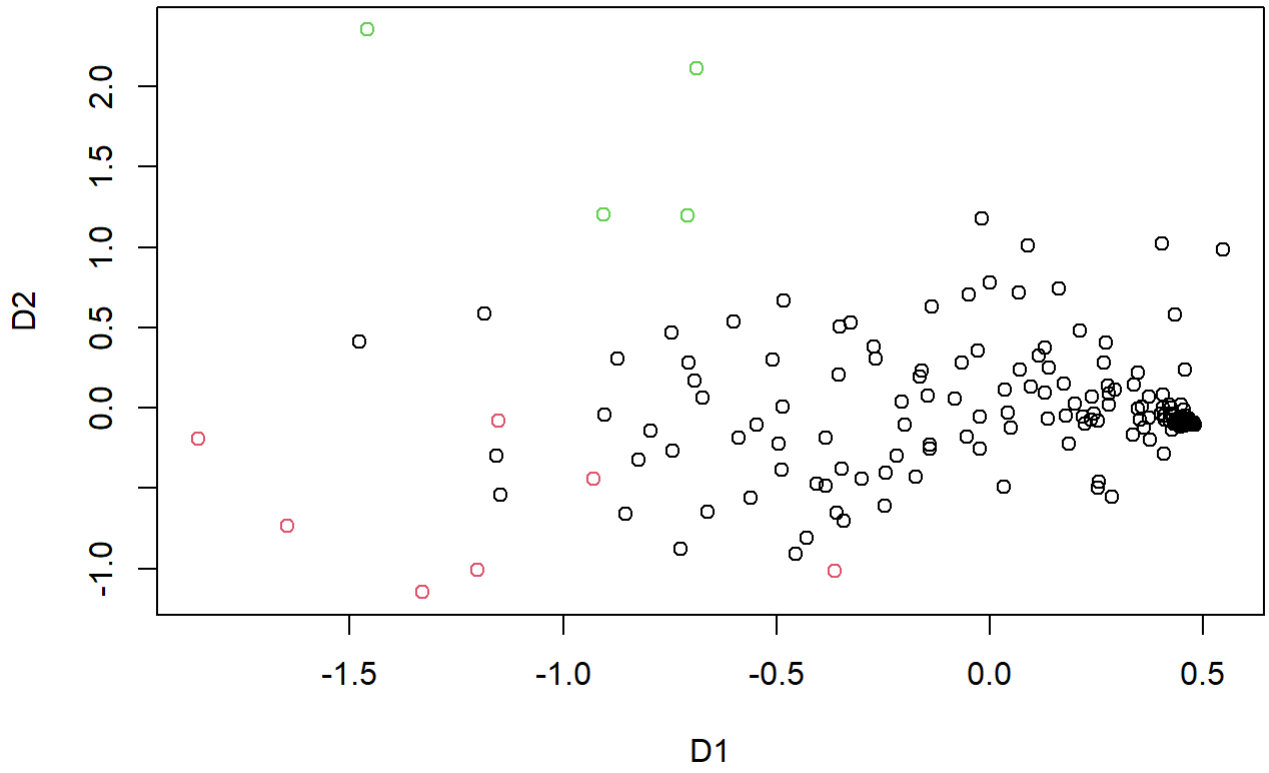
In the following, 3 seems a reasonable choice for clustering since it separates at least 2 big clusters for each distance and method clustering combination and does not result in separating single outliers only, according to the hierarchical clustering results.

The mds plots do not provide additional information about the number of clusters.

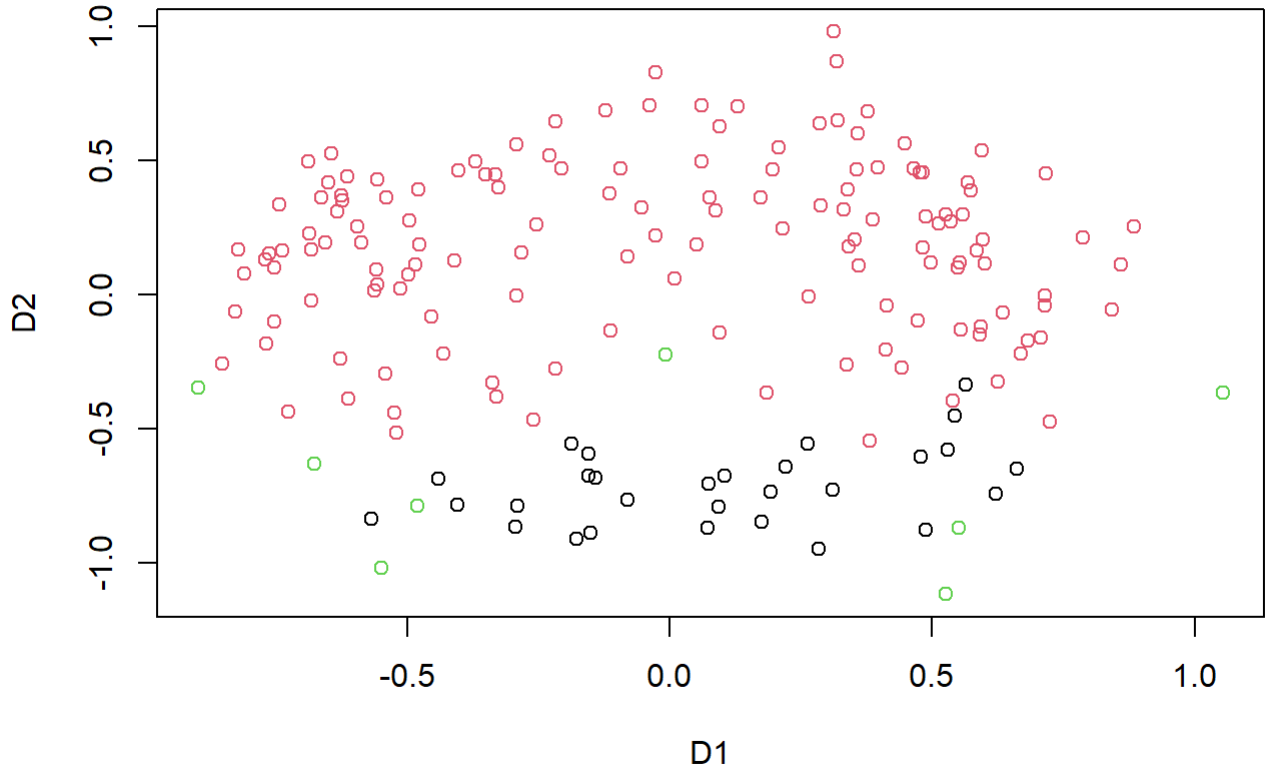
```
eucl_avg_3 <- cutree(eucl_avg, 3)
eucl_com_3 <- cutree(eucl_com, 3)

plot(mds_eucl$conf, col=eucl_avg_3)
```



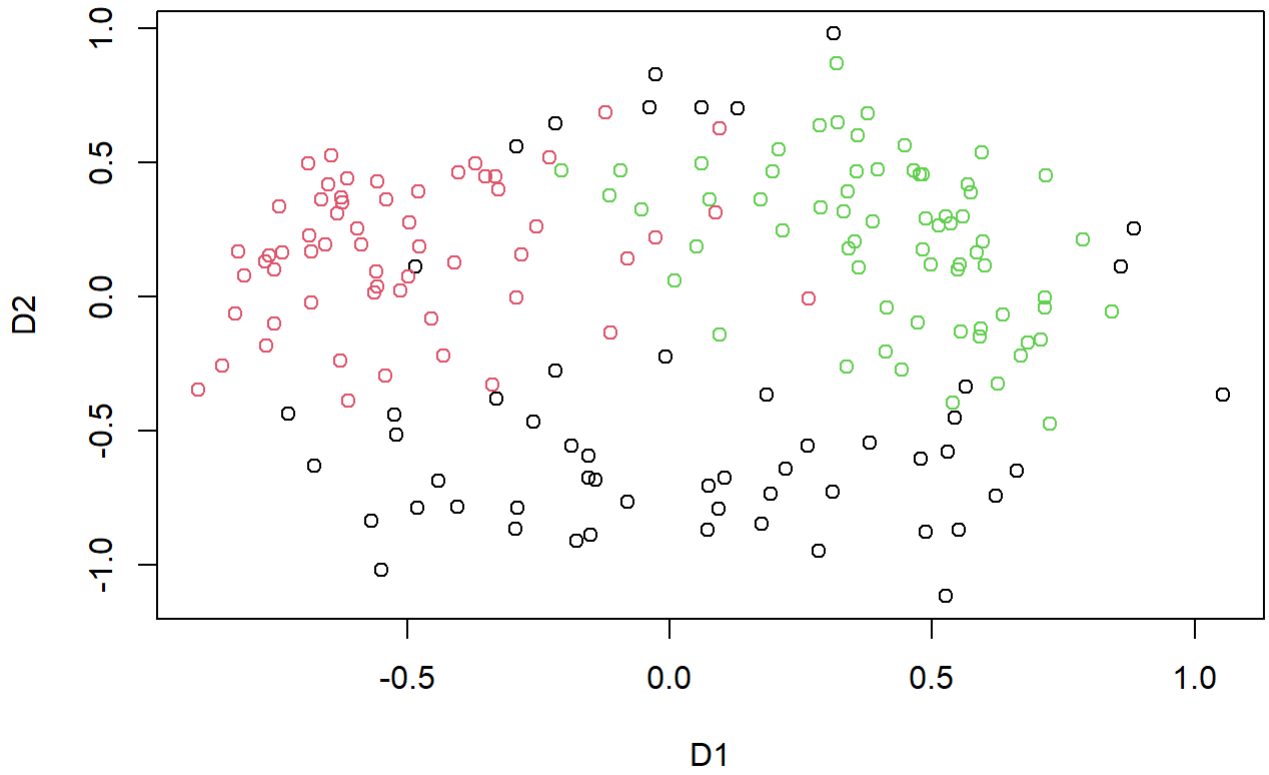

```
plot(mds_eucl$conf, col=eucl_com_3)
```




```
corr_avg_3 <- cutree(corr_avg, 3)
corr_com_3 <- cutree(corr_com, 3)

plot(mds_corr$conf, col=corr_avg_3)
```

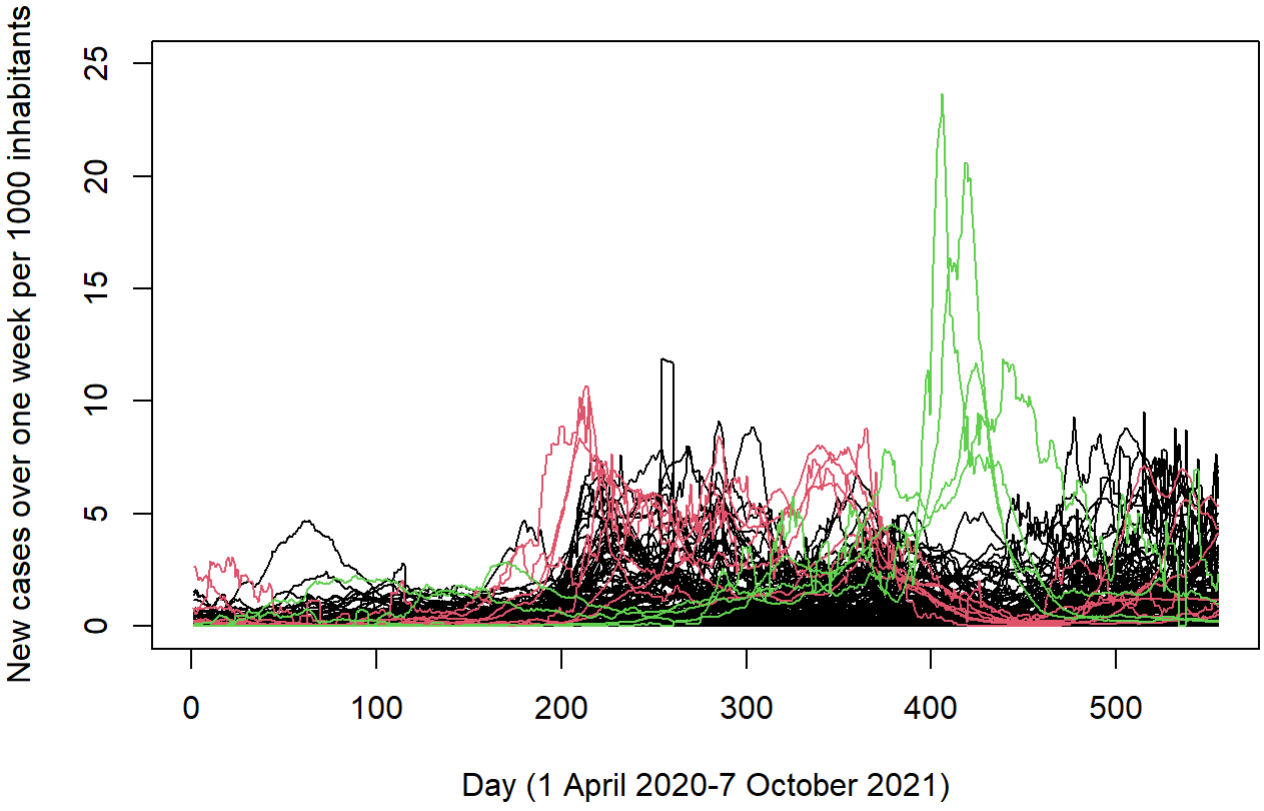



```
plot(mds_corr$conf, col=corr_com_3)
```

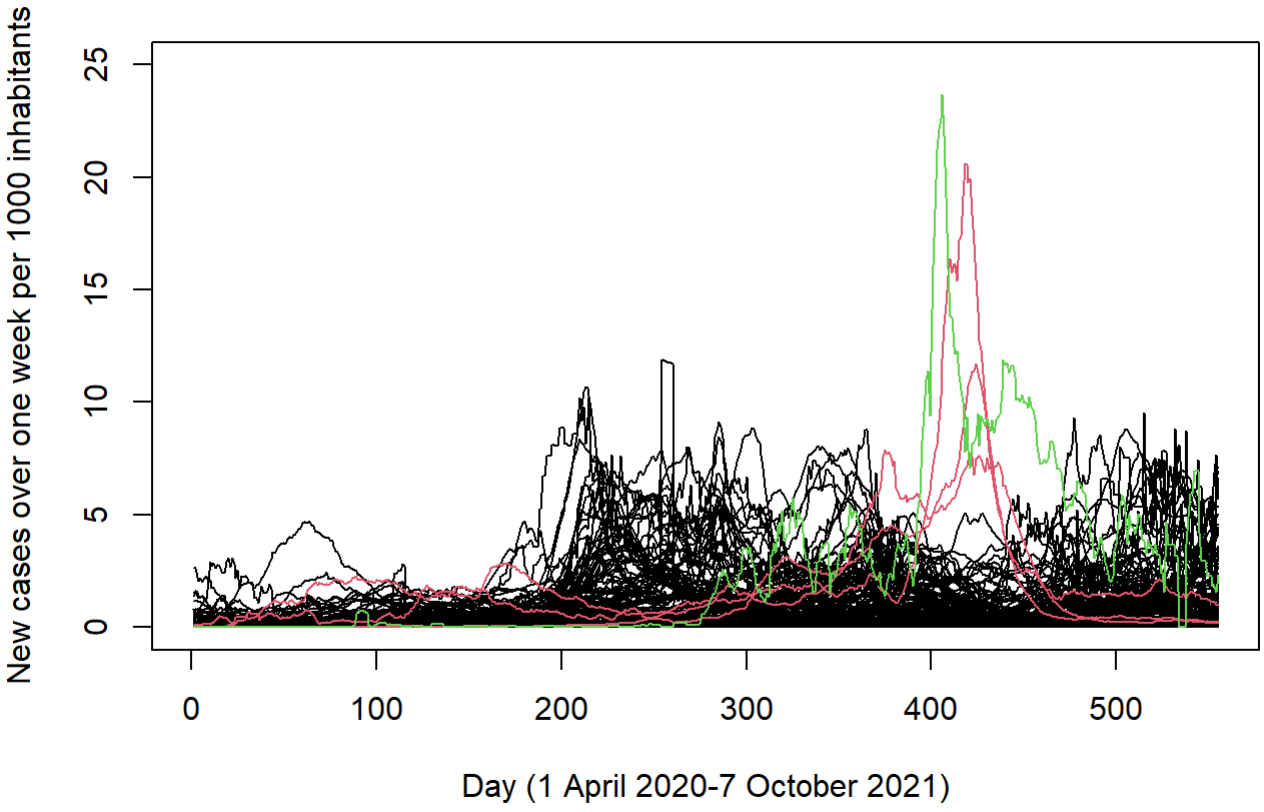


Notable is especially the differences in the number of observations per cluster between the euclidean dissimilarity and the correlation dissimilarity. While the first one focusses on outliers, the second one divides the dataset more evenly into clusters.

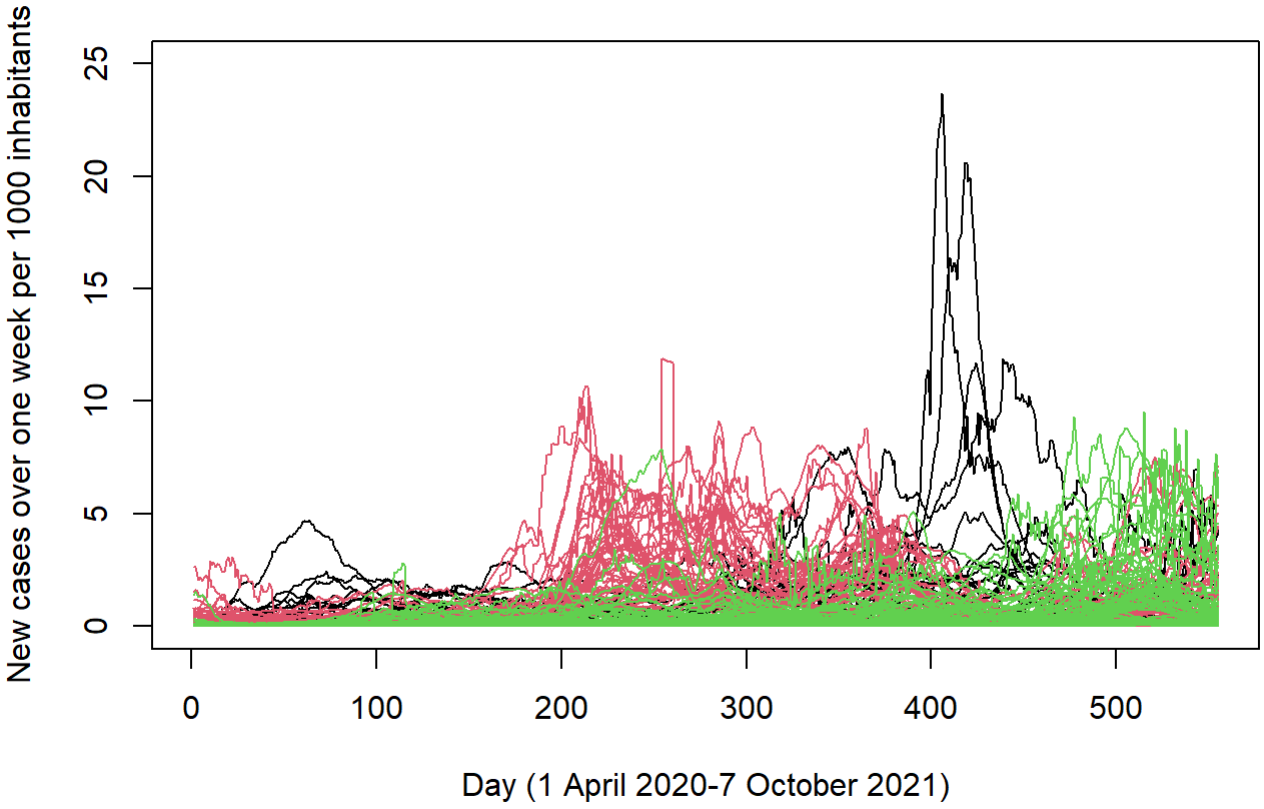
```
## Plot the time series with different colors for each cluster

timeseries_cluster <- function(data, clustering) {
    n_clusters <- max(unname(clustering))
    for (i in 1:n_clusters) {
        cluster = data[which(clustering == i),]
        if (i == 1) {
            plot(1:555,cluster[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(j in 2:179) {
                points(1:555,cluster[j,],type="l", col=i)
            }
        } else {
            for(j in 1:179) {
                points(1:555,cluster[j,],type="l", col=i)
            }
        }
    }
}

# Euclidean distance
timeseries_cluster(covid2021cl, eucl_com_3)
```



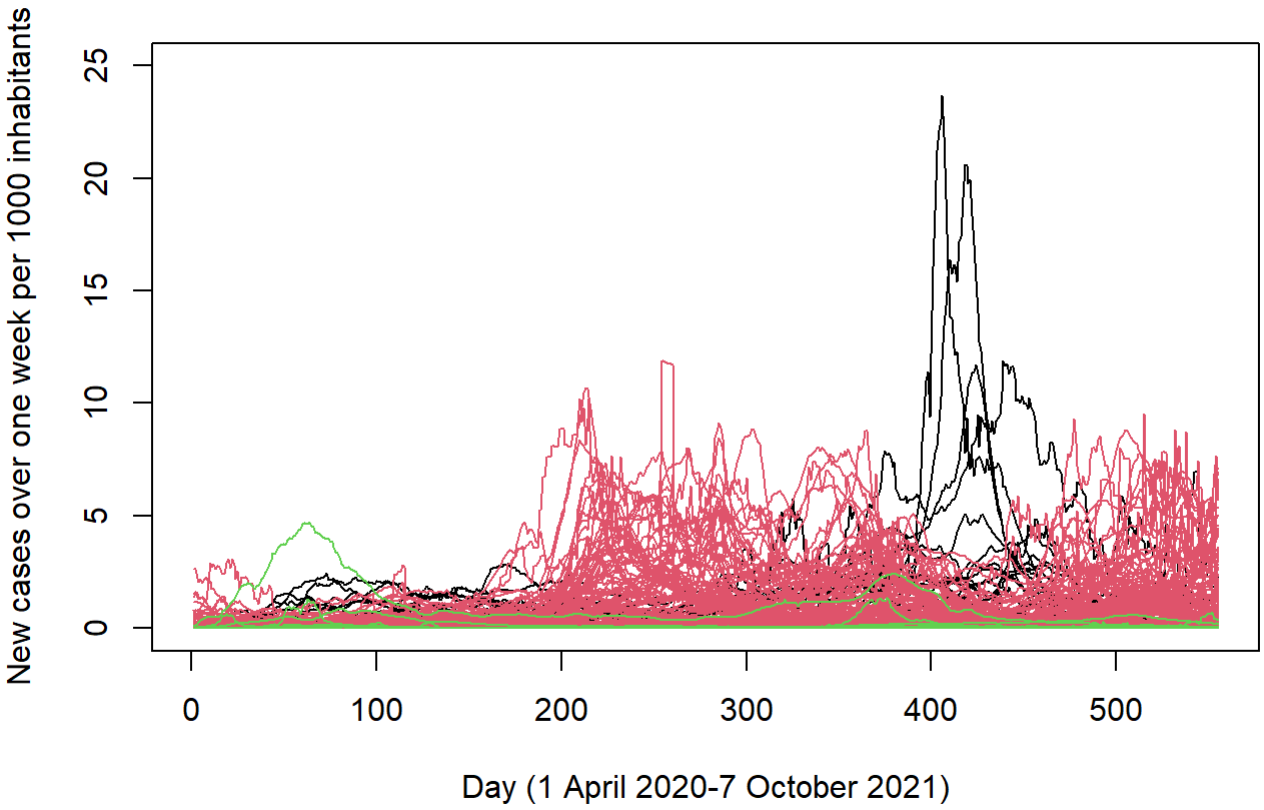
```
timeseries_cluster(covid2021cl, eucl_avg_3)
```




```
# Correlation distance
timeseries_cluster(covid2021cl, corr_com_3)
```




```
timeseries_cluster(covid2021cl, corr_avg_3)
```



According to the time series visualization, the correlation dissimilarity with an average linkage leads to rather good results with to some extent distinctive clusters. Separation is present by different countries having different peaks. It looks like in both cases, countries are grouped mostly according to the point in time when their highest peak occured.

These results should be treated with caution. A point of critic is that countries are only matched together/similar, if they show a similar pattern *at the same time*.

A better dissimilarity that should be preferred in the future is a dissimilarity that takes into account the different starting times for different countries. Maybe one country has a very similar dynamic in comparison to another country, but a few weeks later due to its geographical distance to the initial outbreak.

In that case, the countries should be grouped into the same cluster, but especially the euclidean distance will group these two observations further away from each other because the probably differ significantly for each day.

Assignment 3

18 October 2023 12:48

(2) (a)

Simple Matching:

Definition: $d_{SM}(x,z) = \frac{1}{p}\sum_{j=1}^{p} \mathbb{1}(x_j \neq z_j)$, with x, y, z p-dimensional vectors/listsleds.

For each $j \in \{1, \dots, p\}$ where $x_j \neq z_j$, we know that:

$x_j \neq y_j$ or $y_j \neq z_j$ (Proof: assume opposite $\Rightarrow x_j = y_j$ and $y_j = z_j \Rightarrow x_j = z_j$ ↯)

$\Rightarrow \{j : x_j \neq y_j \text{ or } y_j \neq z_j\} \supseteq \{j : x_j \neq z_j\}$ (superset)

$\Rightarrow |\{j : x_j \neq y_j \text{ or } y_j \neq z_j\}| \geq |\{j : x_j \neq z_j\}|$

$\Rightarrow \sum_{j=1}^{p} \mathbb{1}(x_j \neq z_j) \leq \sum_{j=1}^{p} \mathbb{1}(x_j \neq y_j \text{ or } y_j \neq z_j) = \sum_{j=1}^{p} \mathbb{1}(x_j \neq y_j) + \mathbb{1}(y_j \neq z_j) - \underbrace{\mathbb{1}(x_j \neq y_j \text{ and } y_j \neq z_j)}_{\leq 0} \leq \sum_{j=1}^{p} \left(\mathbb{1}(x_j \neq y_j) + \mathbb{1}(y_j \neq z_j)\right) = \sum_{j=1}^{p} \mathbb{1}(x_j \neq y_j) + \sum_{j=1}^{p} \mathbb{1}(y_j \neq z_j)$

$\Rightarrow d_{SM}(x,z) = \frac{1}{p}\sum_{j=1}^{p} \mathbb{1}(x_j \neq z_j) \leq \frac{1}{p}\sum_{j=1}^{p} \mathbb{1}(x_j \neq y_j) + \frac{1}{p}\sum_{j=1}^{p} \mathbb{1}(y_j \neq z_j) = d_{SM}(x,y) + d_{SM}(y,z)$ □

Mahalanobis distance:

(Here, the inequality for the squared distance is shown, but the inequality for the "normal" Mahalanobis distance is a direct consequence of that.)

Definition: $d_M(x,z)^2 = (x-z)^T S^{-1} (x-z)$, for all $x, y, z \in \mathbb{R}^p$

Set $\tilde{x} = D^{-\frac{1}{2}}(U^{-1})^T x$

$\tilde{y} = D^{-\frac{1}{2}}(U^{-1})^T y$

$\tilde{z} = D^{-\frac{1}{2}}(U^{-1})^T z$

$\Rightarrow d_M(x,z)^2 = (x-z)^T S^{-1} (x-z)$

(Eigenvalue theorem) $= (x-z)^T (UDU^T)^{-1} (x-z)$

(Algebra) $= (x-z)^T \left((UD^{\frac{1}{2}})(D^{\frac{1}{2}}U^T)\right)^{-1} (x-z)$

$= (x-z)^T (D^{\frac{1}{2}}U^T)^{-1} (UD^{\frac{1}{2}})^{-1} (x-z)$

$= (x-z)^T (U^{-1}D^{-\frac{1}{2}})(D^{-\frac{1}{2}}U^{-1})^T (x-z)$

$= \left((U^{-1}D^{-\frac{1}{2}})^T (x-z)\right)^T \left(D^{-\frac{1}{2}}(U^{-1})^T x - D^{-\frac{1}{2}}(U^{-1})^T z\right)$

$= \left(D^{-\frac{1}{2}}(U^{-1})^T (x-z)\right)^T (\tilde{x} - \tilde{z})$

$= (\tilde{x}-\tilde{z})^T (\tilde{x}-\tilde{z})$ ← Squared euclidean distance

$\Rightarrow d_M(x,y)^2 = (\tilde{x}-\tilde{y})^T (\tilde{x}-\tilde{y})$ and $d_M(y,z)^2 = (\tilde{y}-\tilde{z})^T (\tilde{y}-\tilde{z})$

$\Rightarrow d_M(x,z) = (\tilde{x}-\tilde{z})^T (\tilde{x}-\tilde{z}) \leq (\tilde{x}-\tilde{y})^T (\tilde{x}-\tilde{y}) + (\tilde{y}-\tilde{z})^T (\tilde{y}-\tilde{z}) = d_M(x,y) + d_M(y,z)$ □

(↑ known for (squared) euclidean distance)