Homework3_modernstat

Federico Veronesi

2023-10-13

Exercise 1:

Manual computation

Gower coefficient:

$$d_G(x_i, x_j) = \frac{\sum_{l=1}^{p} \frac{w_l}{s_l} \delta_{ijl} d_l(x_{il}, x_{jl})}{\sum_{l=1}^{p} \delta_{ijl} w_l}$$

Let's recall that δ_{ijl} assumes value 0 if there's at least one missing value for the variable l. Here there are missing values for variables 4 and 5.
Plus, variables 2,3,4 are binary (the Jaccard distance has to be used): so the double zeros will count as 0 in the denominator and in the numerator.

$$d_G(x_1, x_2) = \frac{1+1++1+0+0}{1+1+1+0+0} = \frac{3}{3} = 1$$
$$d_G(x_1, x_3) = \frac{1+0+1+0+5/9}{1+1+1+0+1} = \frac{23/9}{4} = \frac{23}{36} = 0.639$$
$$d_G(x_1, x_4) = \frac{1+0+1+0+9/9}{1+1+1+0+1} = \frac{3}{4} = 0.75$$
$$d_G(x_2, x_3) = \frac{0+1+0+0+0}{1+1+0+0+0} = \frac{1}{2} = 0.5$$
$$d_G(x_2, x_4) = \frac{1+1+0+0+0}{1+1+0+0+0} = \frac{2}{2} = 1$$
$$d_G(x_3, x_4) = \frac{1+0+0+0+4/9}{1+1+0+0+1} = \frac{13/9}{3} = \frac{13}{27} = 0.481$$

R computation in R using 'daisy'

```
library(cluster)
#weights = 1
#delta_ij = 0 for variables 4,5 in distances involving x2 and/or x3

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)
m <- rbind(x1, x2, x3, x4)
df <- as.data.frame(m)
df$V1 <- as.factor(df$V1)
df$V5 <- as.numeric(df$V5)
#V1: simple matching
```

```r
#V2,V3,V4: Jaccard (type="asymm")
#V5: continuous: minkovsky
gower <- daisy(df, metric = "gower", type = list(asymm=c(2:4)))
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
```

Exercise 2

a) Simple matching distance

In order to prove that the simple matching is a distance, it's sufficient to take in exam the triangle inequality and to show that it is fulfilled for every possible set of observations. First, let's consider the simple matching for binary data:

Binary variables

$$\frac{1}{p}\sum_{j=1}^{p}\mathbf{1}\left(x_{1j}\neq x_{2j}\right)+\frac{1}{p}\sum_{j=1}^{p}\mathbf{1}\left(x_{2j}\neq x_{3j}\right)\geq\frac{1}{p}\sum_{j=1}^{p}\mathbf{1}\left(x_{1j}\neq x_{3j}\right)$$

We can collect the term $\frac{1}{p}$ in the left member and divide both sides by $\frac{1}{p}$ (which is a positive number).
Now, let's consider all the possible combinations of values for our binary variables:
a = number of variables for which $x_{1j}\neq x_{2j}$ and $x_{2j}=x_{3j}$ (or viceversa) -> this means that $x_{1j}\neq x_{3j}$: these variables bring a contribution of 1 to both the first and the second member of the disequality.
b = number of variables for which $x_{1j}=x_{2j}$ and $x_{2j}=x_{3j}$ -> this means that $x_{1j}=x_{3j}$: these variables bring a contribution of 0 to both the first and the second member of the disequality.
c = number of variables for which $x_{1j}\neq x_{2j}$ and $x_{2j}\neq x_{3j}$ -> this means that $x_{1j}=x_{3j}$: these variables bring a contribution of 2 to the first member and a contribution of 0 to second member of the disequality.

Let's rewrite the inequality according to these combinations:

$$a*1+b*0+c*1+a*0+b*0+c*1\geq a*1+b*0+c*0$$
$$a+c+c\geq a$$
$$2c\geq 0$$
$$c\geq 0$$

This is always true, since c could take integer values greater or equal than 0.

Categorical variables with more than 2 levels

Let's reconsider all the possible combinations:
a = number of variables for which $x_{1j} \neq x_{2j}$ and $x_{2j} = x_{3j}$ (or viceversa) -> this means that $x_{1j} \neq x_{3j}$: these variables bring a contribution of 1 to both the first and the second member of the disequality.
b = number of variables for which $x_{1j} = x_{2j}$ and $x_{2j} = x_{3j}$ -> this means that $x_{1j} = x_{3j}$: these variables bring a contribution of 0 to both the first and the second member of the disequality. (The situations of a and b are exactly the same of having binary variables)
c = number of variables for which $x_{1j} \neq x_{2j}$ and $x_{2j} \neq x_{3j}$ -> this will not necessary imply that $x_{1j} = x_{3j}$, since we have more than 2 categories. The "worst" scenario for what we are trying to demonstrate is that $x_{1j} \neq x_{3j} \forall j$: these variables bring a contribution of 2 to the first member and a contribution of 1 to the second member of the inequality.

$$a * 1 + b * 0 + c * 1 + a * 0 + b * 0 + c * 1 \geq a * 1 + b * 0 + c * 1$$
$$a + c + c \geq a + c$$
$$c \geq 0$$

This is always true, since c could take integer values greater or equal than 0.

b) Counter example for the correlation-based dissimilarity

As a counter example, showing that this dissimilarity doesn't fulfill the triangle inequality, let's take the data from the last point of exercise 2. We stressed that, using this measure, units 1 and 3 should be very similar, while unit 2 should be dissimilar from both units 1 and 3.

```
y1 <- c(1, 4, 5, 4, 2, 1, 1, 4)
y2 <- c(2, 3, 2, 2, 3, 3, 3, 3)
y3 <- c(7, 11, 11, 12, 9, 8, 8, 12)

d12 <- (1-cor(y1, y2))/2
d13 <- (1-cor(y1, y3))/2
d23 <- (1-cor(y2, y3))/2

d12

## [1] 0.6447072

d13

## [1] 0.03577897

d23

## [1] 0.5522233
```

In this case, $d_{12} > d_{13} + d_{23}$, so the triangle inequality isn't fulfilled.

b) Counter example for the Gower coefficient

Here, let's take the data described in point 1, for which the Gower coefficient is already computed (by hand and by means of the daisy function).

```
gower23 <- gower[4]
gower24 <- gower[5]
gower34 <- gower[6]
gower24 <= gower23+gower34

## [1] FALSE
```

Exercise 3

a) Jaccard or Simple Matching?

Jaccard distance -> basically, the dissimilarity between two people will depend only on:
a) The n° of responses in which both answered "something else" (double 1), and
b) the n° of responses in which only one people expressed a preference for "something else".

Let's make an example on 6 questions:
q1=c(0,0,0,0,0,0) q2=c(0,0,0,0,0,1)

In this case, Jaccard = 1, and simple matching = 1/6 = 0.17. Using simple matching, probably these two individuals will be put together in a cluster, while using Jaccard they will certainly be placed in two different clusters.
As a political reseacher, I would like to study the differences in what people WANT TO CHANGE: i would be not much interested on the fact that they agree on something that already happens. For this reason, in the example i would like to have units q1 and q2 in separate clusters, since there are 0 topics that they both want to be different.
To conclude, I would prefer a distance-based method that uses the Jaccard distance.

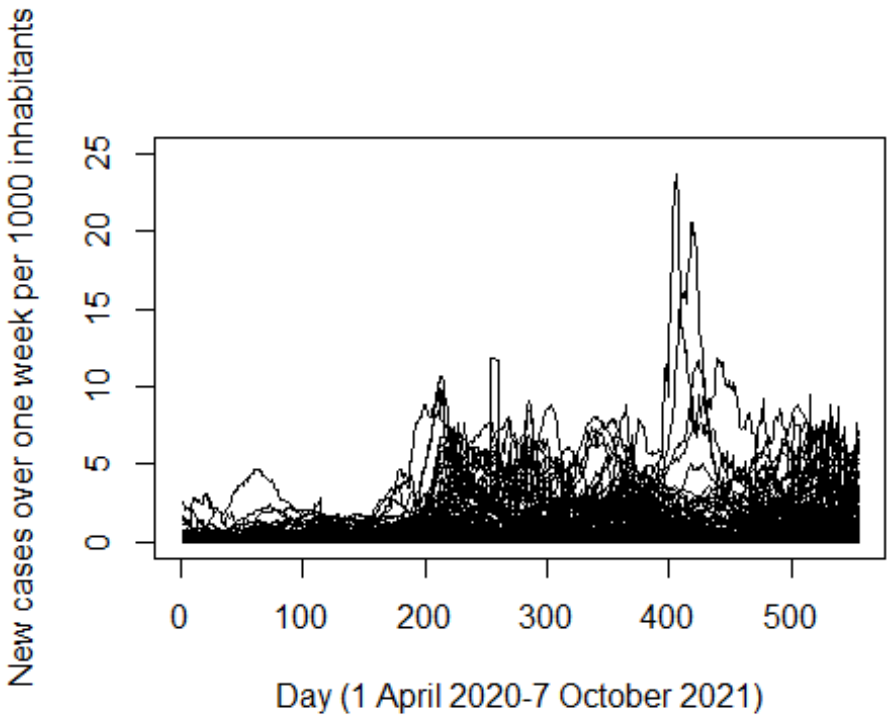
b) Avalanches data

First of all, I would exclude the use of distances (Manhattan/euclidean) on raw data, because the variables have different size and measure. For example, the investments in security are measured in Swiss Francs and may probably have a size of (at least) hundreds of thousands, while the average % of the area covered by avalanches is a percentage and vary from 0 to 100. So the choice is restricted to Euclidean on scaled data, Manhattan on scaled data and Mahalanobis.
An important question has to be asked in order to decide if the Mahalanobis distance is appropriate for this situation: does the correlation between variables constitute a relevant information which we want to keep? If not, Mahalanobis is appropriate since correlated variables are treated as carrying redundant information. If we are interested in keeping correlated variables and underline that in our cluster analysis, then the use of a Minkovsky measure (euclidean or Manhattan) is more suitable. Here, the first three variables will probably be positively correlated. With the last two variables it's more difficult to think of a correlation: maybe a negative correlation with the first three could emerge. My decision is that we want to keep correlation as a relevant information, and look in the resulting clusters if a low budget for security and emergency would be related with a higher n°of victims or in a bigger area. So let's exclude the Mahalanobis and reduce the choice among only two distance: Mahnattan or Euclidean.
Important question in distinguishing Euclidean and Manhattan: do we want to assign higher weights to variables with higher distances? If yes, the exponent of Minkovsky has to be higher (-> Euclidean distance, q=2). If not, the Manhattan (q=1) is preferable. Here, this information could

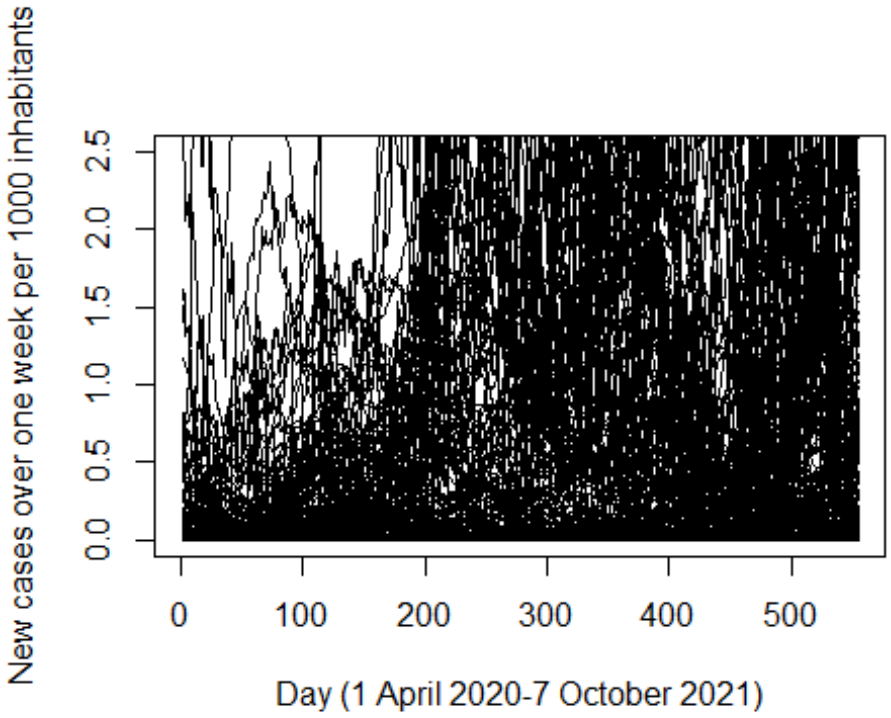
be very interesting. There could be areas with many avalanches and other areas with no avalanches. Maybe for some regions a lot of money is stored, while for others the security could be underestimate. Recall that Swiss is a federal country! To conclude, i would prefer the euclidean distance (on SCALED DATA) in order to give importance to higher distances.

Exercise 4

```
covid2021 <- read.table("covid2021.dat")
covid2021cl <- covid2021[,5:559] # This selects the variables for clustering
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)
points(1:555,covid2021cl[i,],type="l")
```

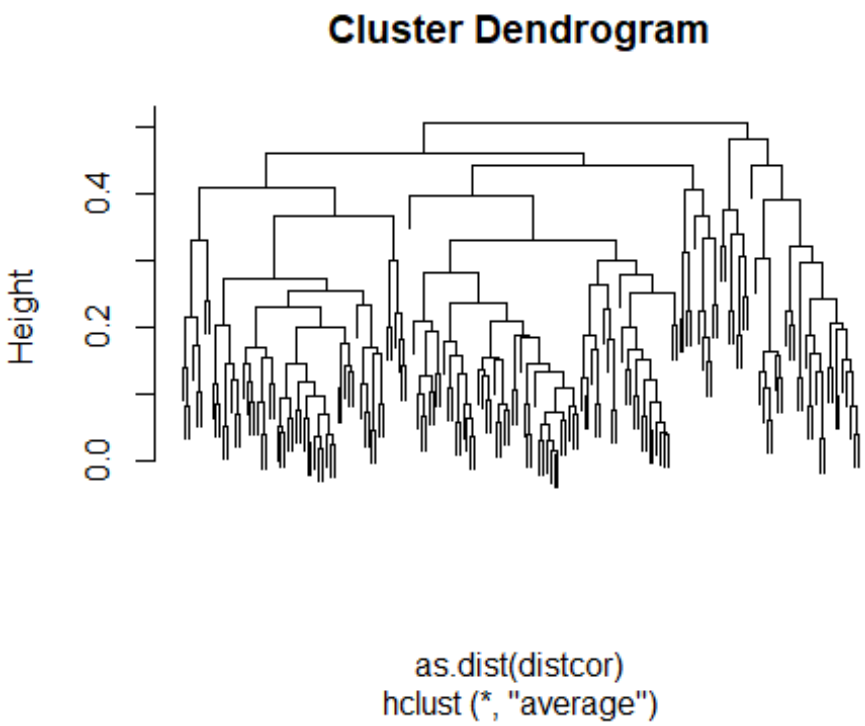



```
# low values
plot(1:555,covid2021cl[1,],type="l",ylim=c(0,2.5),
ylab="New cases over one week per 1000 inhabitants",
xlab="Day (1 April 2020-7 October 2021)")
for(i in 2:179)
points(1:555,covid2021cl[i,],type="l")
```

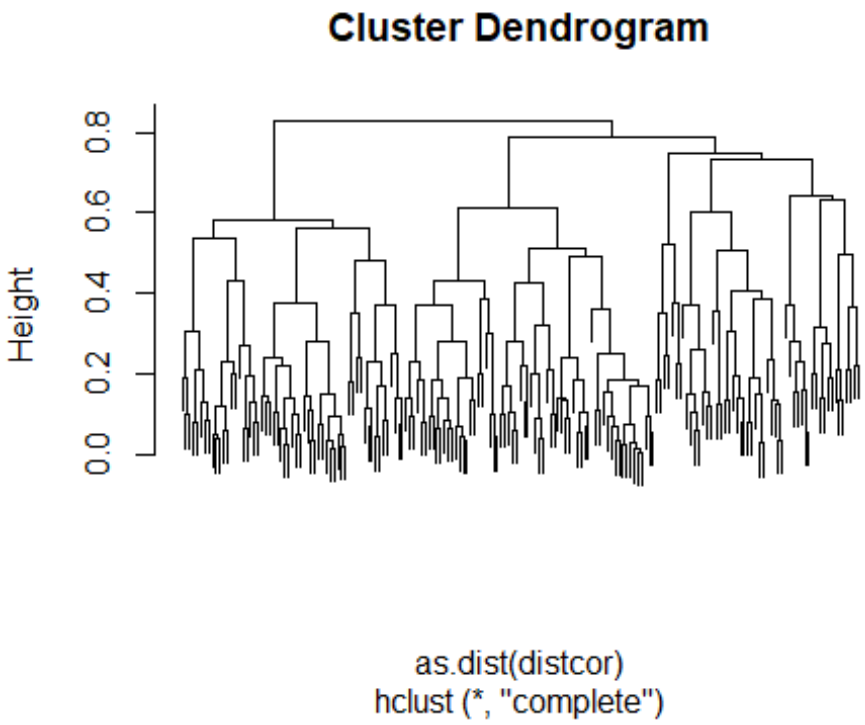


Correlation-based distance

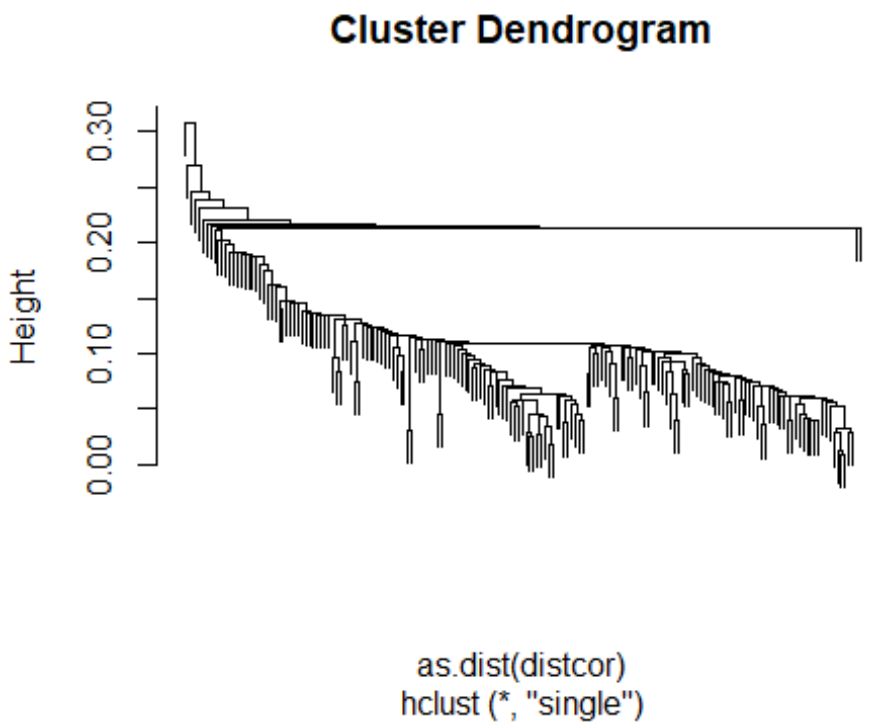
```
cormatrix <- cor(t(as.matrix(covid2021cl)))
distcor <- 0.5-cormatrix/2
hierarchical.avg <- hclust(as.dist(distcor), method = "average")
plot(hierarchical.avg, labels = F)
```



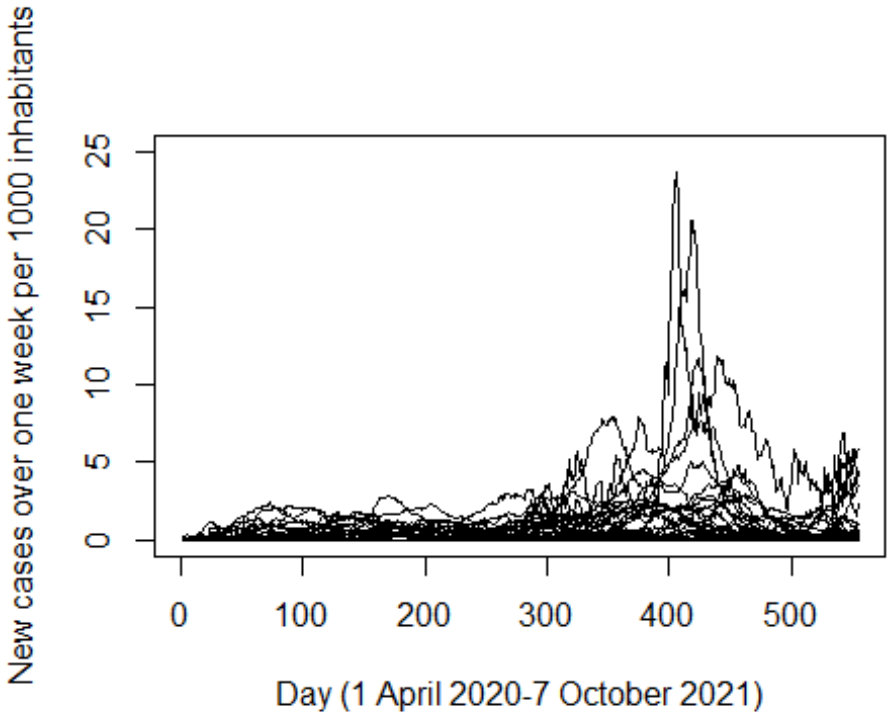

```
hierarchical.compl <- hclust(as.dist(distcor), method = "complete")
plot(hierarchical.compl, labels = F)
```



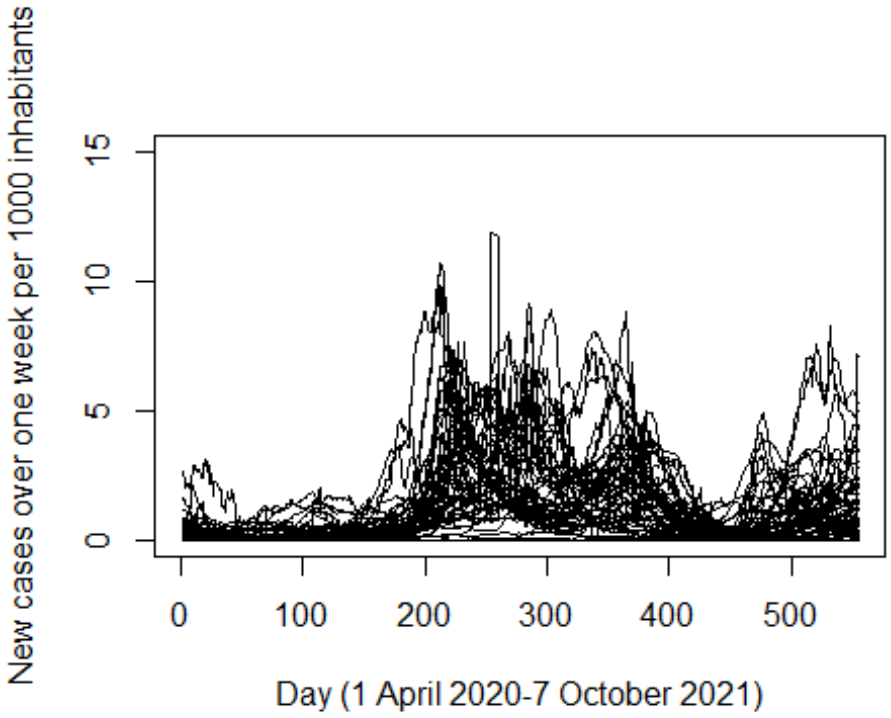

```
hierarchical.single <- hclust(as.dist(distcor), method = "single")
plot(hierarchical.single, labels = F)
```



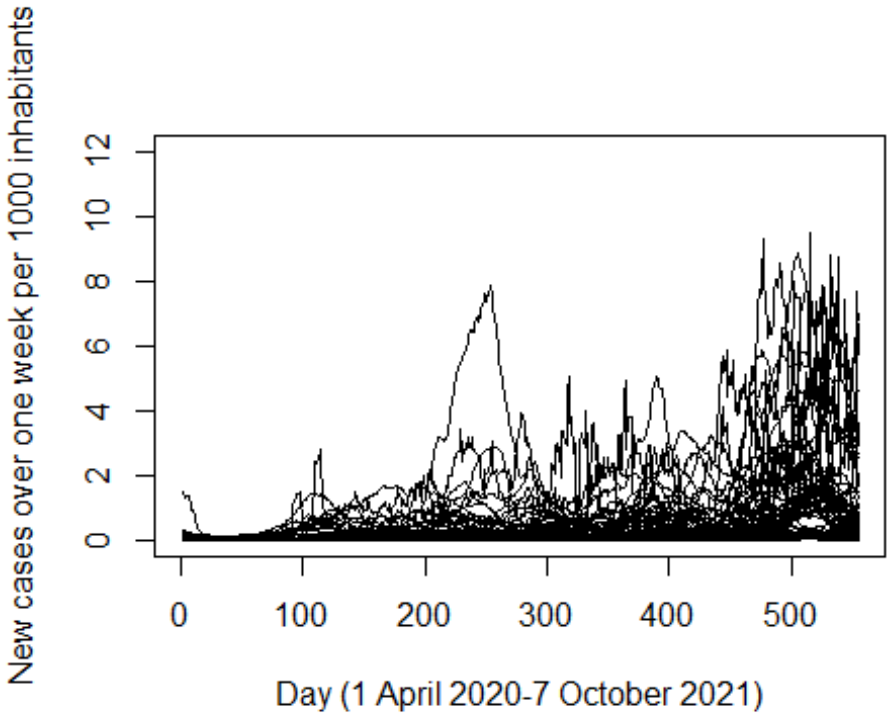

```
avg3 <- cutree(hierarchical.avg, k=3)
compl4 <- cutree(hierarchical.compl, k=4)
covid2021cl$cluster.avg3 <- avg3
covid2021cl$cluster.compl4 <- compl4

cl1 <- covid2021cl[covid2021cl$cluster.compl4 == 1,]
plot(1:555,cl1[1,1:555],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:nrow(cl1))
points(1:555,cl1[i,1:555],type="l")
```



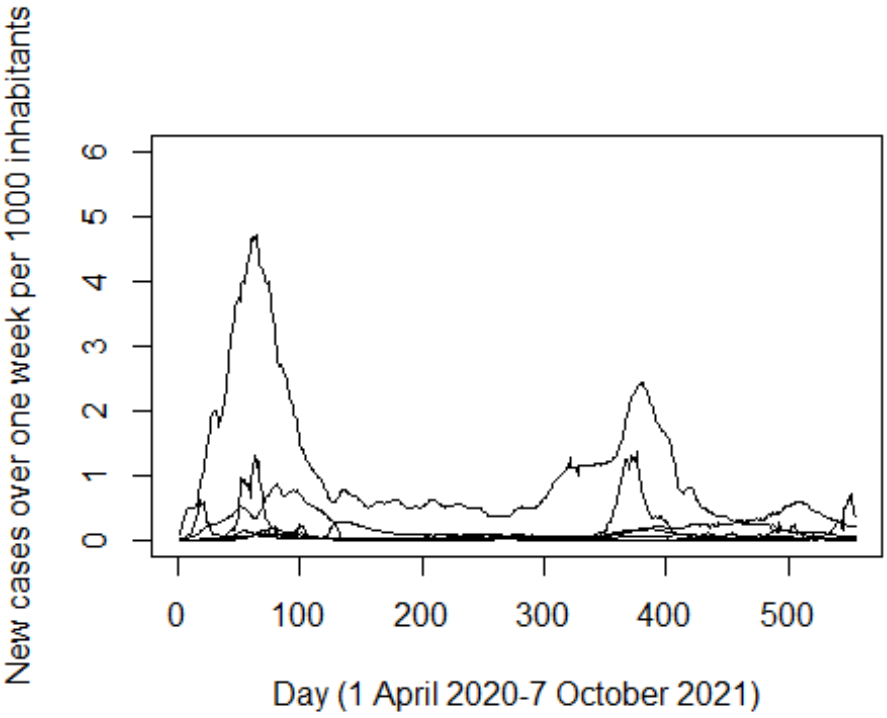

```
cl2 <- covid2021cl[covid2021cl$cluster.compl4 == 2,]
plot(1:555,cl2[1,1:555],type="l",ylim=c(0,15),
ylab="New cases over one week per 1000 inhabitants",
xlab="Day (1 April 2020-7 October 2021)")
for(i in 2:nrow(cl2))
points(1:555,cl2[i,1:555],type="l")
```




```
cl3 <- covid2021cl[covid2021cl$cluster.compl4 == 3,]
plot(1:555,cl3[1,1:555],type="l",ylim=c(0,12),
ylab="New cases over one week per 1000 inhabitants",
xlab="Day (1 April 2020-7 October 2021)")
for(i in 2:nrow(cl3))
points(1:555, cl3[i,1:555],type="l")
```




```
cl4 <- covid2021cl[covid2021cl$cluster.compl4 == 4,]
plot(1:555,cl4[1,1:555],type="l",ylim=c(0,6),
ylab="New cases over one week per 1000 inhabitants",
xlab="Day (1 April 2020-7 October 2021)")
for(i in 2:nrow(cl4))
points(1:555,cl4[i,1:555],type="l")
```


New cases over one week per 1000 inhabitants
0
1
2
3
4
5
6
0
100
200
300
400
500
Day (1 April 2020-7 October 2021)