FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Comparison of national COVID-19 time series
Student:	Michael Kolínský
Supervisor:	Ing. Kamil Dedecius, Ph.D.
Study program:	Informatics
Branch / specialization:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	until the end of summer semester 2022/2023

Instructions

Aktuální epidemie COVID-19 je z hlediska informatiky doprovázena bezprecendentní dostupností národních časových řad vývoje velké řady různých epidemiologicky významných ukazatelů. Cílem bakalářské práce je tyto národní řady porovnat pomocí vhodných metrik, diskutovat podobnosti či odlišnosti, případně i vhodnost vybraných metrik.

V bodech:
1) nastudujte a popište metody pro porovnávání časových řad z hlediska vývoje, struktury a podobně,
2) nastudujte vlastnosti vybraných národních časových řad souvisejících s epidemií COVID-19,
3) aplikujte vhodné vybrané metody a diskutujte své výsledky a pozorování.

Electronically approved by doc. Ing. Jan Janoušek, Ph.D. on 17 February 2021 in Prague.

Bachelor thesis

COMPARISON OF NATIONAL COVID-19 TIME SERIES

Michael Kolínský

Faculty of information technology CTU in Prague
Department of computer science
Supervisor: Ing. Kamil Dedecius, Ph.D.
May 12, 2021

Contents

List of Figures

List of Tables

Seznam výpisů kódu

I would like to sincerely thank to Ing. Kamil Dedecius, Ph.D. for the time given, valuable advice and especially for the willingness. I would also like to thank to my family for their support throughout the bachelor's degree.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on April 25, 2021 .....

Abstrakt

Tato práce se zabývá analýzou národních časových řad denního počtu nově nakažených virem COVID-19. Data jsou převzatá ze Světové zdravotnické organizace. Ve fázi předzpracování dat jsou národní řady přeškálovány na počet obyvatel v dané zemi. Datům je snížena dimenzi s pomocí metody Piecewise aggregate approximation a jsou odstraněny všechny složky časových řad s vyjímkou trendu. V práci jsou definovány čtyři metody porovnání časových řad jako Dynamic Time Warping (DTW), Edit Distance With Real Penalty (ERP), Longest Common Subsequence Similarity ($LCSS$) a Diskrétní Fréchetova vzdálenost. V další fázi je na předzpracovaná data aplikován algoritmus aglomerativního hierarchického shlukování s použitím průměrné párové vzdálenosti a využitím předchozích metrik. V poslední fázi jsou zvoleny výsledné počety shluků pro všechny metriky s využitím dendrogramu. V závěru práce se nachází vykreslené shluky, které jsou diskutovány spolu s vlastnostmi použitých metod měření vzdálenosti.

Klíčová slova časová řada, COVID-19, měření podobnosti/vzdálenosti časových řad, DTW, ERP, LCSS, Fréchetova vzdálenost, hierarchické shlukování, Piecewise aggregate approximation

Abstract

This thesis analyses the national time series of newly infected people by COVID-19. The data are taken from the World Health Organization. In the preprocessing phase are the national time series scaled to respect the size of the population. The dimension is reduced using the Piecewise aggregate approximation and just the trend component of the time series is taken into account. In the thesis, there are defined four measures of time series (dis)similarity like Dynamic Time Warping (DTW), Edit Distance With Real Penalty (ERP), Longest Common Subsequence Similarity ($LCSS$), and Discrete Fréchet distance. In the following phase, the preprocessed data are clustered using the agglomerative hierarchical clustering algorithm with the use of the average linkage that exploits the defined measures. In the last phase, the resulting count of clusters is chosen for each metric using the dendrogram. In the conclusion of this thesis, there are the resulting plots, which are further discussed together with the properties of the distance measures.

Keywords time series, COVID-19, (Dis)similarity measure of time series, DTW, ERP, LCSS, Fréchet distance, hierarchical clustering, Piecewise aggregate approximation

List of abbreviations

WHO	World Health Organization
DTW	Dynamic Time Warping
ERP	Edit Distance With Real Penalty
LCSS	Longest Common Subsequence Similarity
DFD	Discrete Fréchet Distance
PAA	Piecewise aggregate approximation

Chapter 1

Introduction

Since the year 2019, we face a new type of coronavirus called COVID-19 (COrona VIrus Disease 2019). COVID-19 is very infectious, so it has quickly influenced our lives. World Health Organization declared the outbreak a pandemic in March 2020. The first case registered was in December 2019 in Wuhan, China.

The virus spreads through aerosols or tiny liquid droplets in the air. As of 16 March, there are over 120 million confirmed cases and about 2.6 million deaths, which causes the pandemic to be one of the deadliest of all time. The majority have mild symptoms, but there are groups with severe symptoms, especially older people. Symptoms onset in one to fourteen days after infection, but some individuals have no symptoms at all. Some of those that do not have noticeable symptoms yet can still be infectious. The transmission of COVID-19 differs from nation to nation because of different conditions for the virus.

Governments introduced measures to mitigate the outbreak, but they vary between states a lot. These measures include wearing a face mask, social distancing, movement restriction, quarantine, and so on.

There is a large amount of national data concerning this topic like daily new cases, deaths, recoveries, tested people, vaccine utilization, etc.

This thesis aims to compare the national time series using some (dis)similarity measures and then discuss the similarities and differences of the results and eventually the appropriateness of the measures. To achieve the task, we use time series clustering to compare the time series and then we do a comparison on the resulting clusters and/or a tree called dendrogram.

The aim is to do just the technical analysis and not to interpret the results from the perspective of geography, sociology, and so on… The interpretation requires additional knowledge.

This work can help governments to compare other states and adapt their measures. Results could also be used as a basis for future research on this topic.

The structure of this thesis is divided into the following 3 chapters: Theory and methods, Application to COVID-19 time series, and Discussion and conclusion. In the first one, we define all distance measures and some other methods. In the Application to COVID time series, we apply the defined methods to the time series, and in the last chapter, we discuss the results.

In the rest of this chapter, we present some existing work concerning this topic, and then we introduce the data source.

1.1 Related work

There is a lot of research concerning the time series data analysis of COVID-19. A big part of the research is dedicated to forecasting some variables like daily new cases, deaths, filled hospital beds, etc. [1],[2],[3],[4],[5] Some try to find some correlation with other variables such as air temperature, humidity, air pollution, etc. [6],[7],[8],[9] The majority devotes just to a specific region rather than to the whole world or a continent. Just a few of the most similar are discussed.

The first one [10] takes data from Johns Hopkins University [11] just for the USA. Every time series contains data from the day when the 5th case was confirmed until June 21st, 2020, and contains daily new cases and daily death counts. The time series are clustered using the Dynamic Time Warping distance measure, which can handle different time series lengths. A parametric approach is proposed that considers daily new cases and the daily death toll together in one distance measure. The hierarchical clustering algorithm created clusters using the single linkage method. Nine clusters were obtained using the Calinski-Harabasz criterion. 2 big clusters had 18 and 14 countries, and 3 clusters had just a single country that had unique behavior. Different behavior was observed in the eastern and western parts of the USA. A representative from each cluster was selected, and a prediction was made using the Logistic, Gompertz, and SIR mathematical models.

The second one [12] took data of United States of America, Spain, Italy, Germany, United Kingdom, France, and Iran. The data source was the World Health Organization and contained daily new cases and death counts. Time series for each country recorded data from 22 February 2020 up to 18 April 2020. First of all, they studied the correlation between confirmed cases, total deaths, and population size. They found that there is a strong positive correlation. Based on previous observation, the data were rescaled to the population size of the USA. The fuzzy clustering was applied to the preprocessed data. The clustering results indicated that Italy and Spain's outbreak was very similar and different from other countries.

The third one [13] used data of 191 countries from the Our World In Data [14]. For each country, a time series was formed containing daily new cases of 100 days starting from the day when the 10th case was confirmed. Data are smoothed using a moving average. The paper used the Pearson correlation coefficient transformed to the correlation distance that forms a metric space. A distance matrix containing the distance for each pair of time series was created. They used graph theory to cluster the data. The problem was represented as a complete graph where the time series were nodes, and each edge from node a to node b has a cost assigned according to the distance measure. Then the Kruskal's algorithm was applied to get the minimal spanning tree (MST) of the graph. Based on the MST, a new distance metric was defined called "the subdominant ultrametric distance". This metric was used to cluster the data using the hierarchical algorithm. According to the stopping rules for the clustering algorithm, four clusters were obtained. Cluster 1 had 104 members and tended to continue to grow. One of the reasons stated was a higher level of poverty. Cluster 2 had 43 countries and was the

first to reach the peak of daily infections and quickly entered into a decline phase. This cluster was composed of small countries and islands, and thus the authors highlight the importance of geography as a key factor. Cluster 3 had 43 countries, and after a steep increase, it started to decline. This cluster also had the highest number of deaths and the oldest age group, and thus the authors underline the importance of protecting people in risk factor groups. The last cluster was formed just by Mongolia and a special time series that was the average of all countries.

The fourth one [15] used data of daily new cases in the USA from the Johns Hopkins University [11]. The data were from March 22, 2020 to July 25, 2020. First of all, the time series were smoothed using the 7–day moving average, then they represented the data as a matrix X of dimension $n \times m$, where n was the number of observations (49 in this case) and m was the number of days (126 in this case). Then the matrix was represented as $X \approx WH$ where W is a matrix of coefficients in rows with dimension $n \times r$ and H is a matrix of basis vectors in columns $r \times m$, r was selected using cross validation as 12. The "weighted nonnegative factorization" algorithm was used to get the matrices W and H. Based on the previous procedure, the k-means algorithm was used and the coefficient matrix W was passed as an input. The hyperparameter k was selected using the elbow method as 7. Arizona and Louisiana were singletons in their clusters and there were two big clusters with 13 and 21 states. Compared to the [10] article, the structure of the clusters seems quite the same, as there are two big clusters and a few singletons. On the other hand, the clusters' content is fairly different, which may be because this work did not take daily death counts into account compared to the other one. They also applied previous procedures to each week starting from March 22, 2020, separately to determine if there is a significant change in the clusters' structure over weeks. They found a structural change in the period before and after May 30. They concluded that this change of structure could be due to business reopening in most states.

1.2 Data

Data are taken from the World Health Organization [16] on 14 April 2021. There are multiple data sets: *daily new cases and deaths, latest reported counts, and vaccination.* The *daily new cases and deaths* data set is used for analysis, and the *latest reported counts* data set is used to get the population size for each country. These data sets contain data for countries from all over the world. The structure of the data is in tables 1.1 and 1.2. The reported dates of cases represent case detection as opposed to symptom onset. As they say, the data may not always be accurate: [16]

> "*Case detection, definitions, testing strategies, reporting practice and lag times (e.g., time to case notification, and time to reporting of deaths) differ between countries, territories and areas. These factors, amongst others, influence the counts presented with variable under or overestimation of true case and death counts, and variable delays to reflecting these data at a global level. Due to differences in reporting methods, cut-off times, retrospective data consolidation, and reporting delays, the number of new cases may not always reflect daily totals published by individual countries, territories, or areas. Cases and deaths reported from international conveyances, included in global totals but not reflected*

Table 1.1 Structure of the WHO *daily new cases and deaths* data set [16]

Field name	Description
Date_reported	Date of reporting
Country_code	ISO Alpha-2 country code
Country	Name of country
WHO_region	WHO regional offices
New_cases	Count of new cases for this date
Cumulative_cases	Sum of all cases to this date
New_deaths	Count of new deaths for this date
Cumulative_deaths	Sum of all deaths to this date

Table 1.2 Structure of the WHO *latest reported counts* data set [16]

Field name	Description
Name	Country, territory, area
Cases - cumulative total	Total cases reported to current date
Cases - cumulative total per 100000 population	The previous one scaled
...	

in epidemiological curves as not associated with a country or region."

Counts include both domestic deaths and deaths of foreigners that died in that country. In this work, all of these distortions are assumed to be insignificant, so they are ignored for simplicity. Additionally, the tests do not have sensitivity and specificity equal to 100%, so we will also do not consider it.

There are some cases when a country has reported daily new cases and deaths due to COVID-19, and then after some time, the data was invalidated. For example, when a country reports some newly infected people and then finds out that the tests were wrong. As a result of the updates, there are some records with negative daily new cases and deaths.

Chapter 2

Theory and methods

In this chapter, we start with defining a time series. Then we define the used measures of (dis)similarity. We also introduce the data clustering and a clustering algorithm. Finally, we show a dimensionality reduction technique for time series and the average time series of unequal lengths.

2.1 Time series

Time series is a set of observations in time, more formally [17]:

▶ **Definition 2.1.** *Let $(\Omega, \mathcal{F}, P)$ be a probability space and T a set of indices interpreted as time. Time series is a set $\{X_t, t \in T\}$, where X_t is a random variable from $(\Omega, \mathcal{F}, P)$ for every t.*

In this thesis, we will use just $X_t \in N_0$ for every t from $T \subseteq \mathbb{Z}$.

Time series has the following properties [17]:

Trend component: defined as a long-term evolution of the mean value

Seasonal component: periodically repeating pattern in the time series with a relatively regular period

Cyclic component: fluctuations that do not have a fixed period

Remainder component: noise that the previous components can not describe

This definition is quite vague because the cyclic or even seasonal components can be confused with the trend in a shorter time section.

Time series can be decomposed in the previous components and expressed with the following models:

Additive model: $Y_t = T_t + S_t + C_t + R_t$

Multiplicative model: $Y_t = T_t \cdot S_t \cdot C_t \cdot R_t$

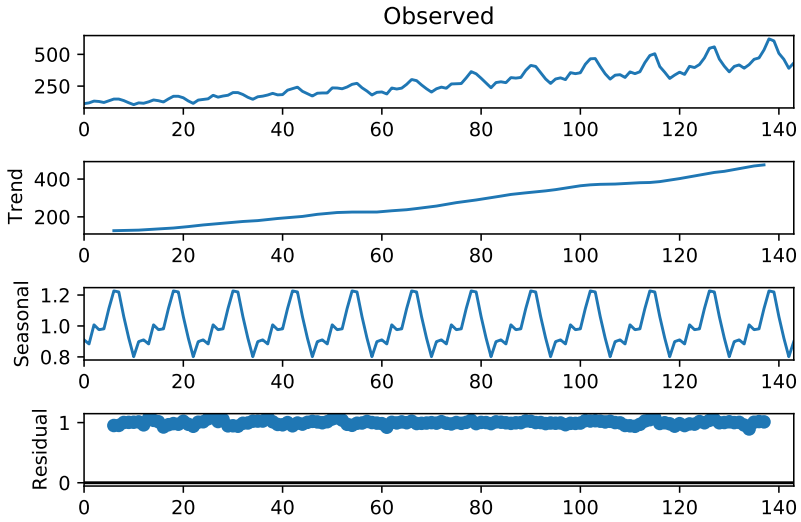


Figure 2.1 Components of the multiplicative model of a time series

Where Y_t, T_t, S_t, C_t and R_t are the values of the time series, the trend component, the seasonal component, the cyclic component, and the remainder component in the time t respectively. The trend and cyclic components are often merged [17],[18]. In Figure 2.1 is a time series decomposition of the multiplicative model using moving averages. The trend and cyclic components are merged into one.

One way to decompose the time series is to use the moving average method [18], but there is an assumption that the seasonality period has a constant length. In this method, the trend and cycle components are merged into trend–cycle component. We define the moving average in a little different way than in [18] because it is handy in another section. A moving average MA^m of size m for a time series $X = x_1, x_2, \ldots, x_n$ of length n is defined as:

$$MA_i^m = \frac{1}{m} \sum_{j=i}^{i+m-1} x_j.$$

The moving average is defined just if the time series has a length of at least m and just for $1 \leq i \leq n-m+1$. The moving average is shifted to the left compared to the original definition, but it is not a problem for us. The shift can be seen in Figure 2.2 where is a MA^5 of a time series.

Using the moving average, we can get the trend–cycle component. At first, we need to determine the size m of the period of the seasonality component. We will call *seasonality elements* the elements that form the period. We can estimate the m, for example, from a chart. When we get the m then we compute the MA^m of the time series. We know that there are m seasonality elements, and the MA_i^m contains each exactly once, so we will presume that the seasonality and remainder variances are averaged out. The blue curve in Figure 2.2 is the sinus function with distortions generated by a uniform distribution in the range from -0.5 to 0.5. We can see that the black curve has a shape close to the sinus function with shortened ends.

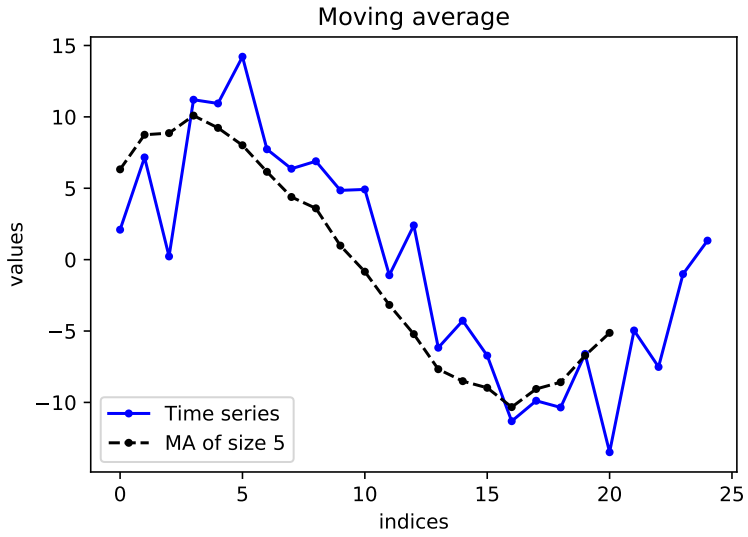


Figure 2.2 Moving average of size $m = 5$

2.2 (Dis)similarity measure

To compare two time series, we need a function that takes 2 time series as arguments and outputs a number that can be interpreted as a measure of similarity or dissimilarity. We will define these 2 types of (dis)similarity measures. The first one that measures an actual similarity returns for two alike time series a higher number, we will call it a similarity measure. On the other hand, the second one that measures dissimilarity returns for two alike time series a lower number, we will call it dissimilarity measure (sometimes it is called a distance measure).

The distance measure usually has some specific properties. When a distance measure holds the following three axioms, then we call it a metric.

▶ **Definition 2.2** (Metric)**.** *A metric on a set $\mathcal{X}$ is a function d: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and satisfies these axioms for every $x, y, z \in \mathcal{X}$ [19]:*

Positive definiteness: *$d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$*

Symmetry: *$d(x, y) = d(y, x)$*

Triangular inequality: *$d(x, y) + d(y, z) \geq d(x, z)$*

Tuple $(d, \mathcal{X})$ forms a metric space.

For example, the Euclidean distance and the Euclidean space form a metric space. When the triangular inequality does not hold, then the function is called a semi-metric [20].

This is a general definition of a metric space. In this thesis, our set $\mathcal{X}$ will be all time series of finite length.

We will use 4 different (dis)similarity measures which we will discuss in this section. The measures must accept 2 time series of possibly different lengths, so we need an elastic measure rather than a locked step measure.

A locked step measure for two time series X and Y always compares ith element of X with ith element of Y. On the other side, an elastic measure does not have this fixed step limitation (but could have some others). This is specific for our domain because the date of the first confirmed case (the first record in a time series) varies between countries, so the time series lengths differ from country to country. We also want to be able to compute it in a reasonable time.

The selected distance measure depends on the kind of similarity we are interested in. It always depends on the domain. It could be any function that satisfies the investigators purpose. The selected measures are one of the most used to compare time series. We do not want to select the "best" one, but we want to observe how they behave on COVID-19 time series and compare the results.

For each measure, first we define it, then we provide a recursive formula. We show that the recursive formula calculates the measure. Finally, based on the formula, we introduce a dynamic programming algorithm.

In the future sections, let X and Y be some time series $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$ of lengths n and m respectively.

2.2.1 Dynamic time warping

The Dynamic Time Warping (DTW) is an elastic distance measure that has been used initially to compare speech data [21]. One feature is it can cope with different time scalings of time series. When we say "Hello" slowly and then fast or when we have the same trajectory of a fast and slow moving objects, this measure finds both time series similar. It is a dissimilarity measure, therefore the lower number, the more alike the time series are. The DTW does not fulfill the triangular inequality, so it is not a metric.

A warping path W is a mapping between points in time series X and Y, let (i, j) denote mapping between points x_i and y_i. We can denote a warping path between 2 time series as $W = w_1, w_2, \ldots, w_l$, where $w_k = (i, j)$ is the kth element of W and l is the length of W. The DTW path has some constraints [22]:

Boundary condition: $w_1 = (1, 1)$ and $w_l = (n, m)$. It means that the first points of both time series are mapped together as well as the last points.

Continuity: If $w_k = (i, j)$ then $w_{k+1} = (i', j')$, $i' - i \leq 1$ and $j' - j \leq 1$ for $1 \leq k < l$. This means we can shift at most by one index to positive numbers (in both series) in the subsequent mapping.

Monotonicity: If $w_k = (i, j)$ then $w_{k+1} = (i', j')$, $i' - i \geq 0$ and $j' - j \geq 0$ for $1 \leq k < l$. In other words, we can just increase the index in the subsequent mapping.

▶ **Definition 2.3.** *The DTW distance between time series X and Y is*

$$DTW(X, Y) = \min \left\{ \sum_{(i,j) \in W} d(x_i, y_j) \middle| W \text{ is a } DTW \text{ warping path} \right\}.$$

In Figure 2.3 we can see two time series (red and blue) and the optimal DTW path. A mapping (i, j) is denoted with a black dashed line.

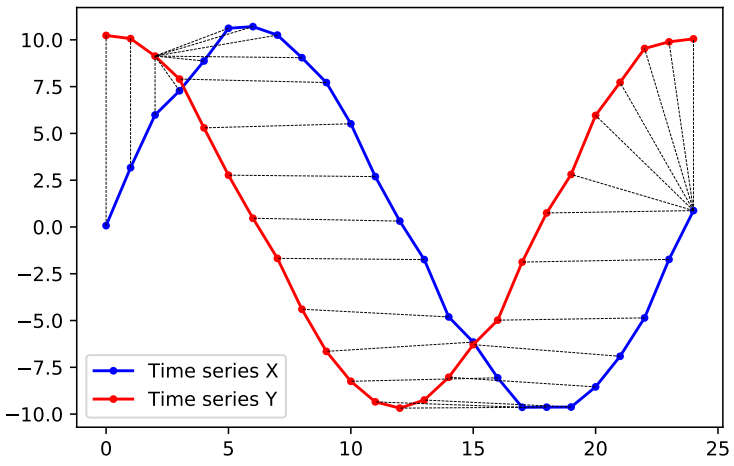


Figure 2.3 A *DTW* path between 2 time series

To calculate the distance, we introduce a new matrix D of size $n \times m$ and every cell $D_{ij} = d(x_i, y_j)$ and represents also a mapping (i, j), we will call it a *DTW* matrix for some X and Y. We show that W corresponds to a special path in the matrix D, and based on this, we calculate the distance.

In the following lemma, we call a "path through a matrix" a sequence of elements of a matrix. An element in ith row and jth column is denoted as (i, j).

▶ **Lemma 2.4.** *Let $P = p_1, p_2, \ldots, p_l$ be a path of length l through the DTW matrix D called D-path. D-path fulfills $p_1 = (1, 1)$, $p_l = (n, m)$, and if $p_k = (i, j)$ then $p_{k+1} \in \{(i+1, j), (i, j+1), (i+1, j+1)\}$ for every $1 \leq k < l$ (shift constraint). Let $Q \subseteq \{(i, j) | 1 \leq i \leq n \wedge 1 \leq j \leq m\}$. Q is a D-path $\iff$ Q is a DTW path.*

Proof. We need to show that: Q is a D-path $\implies$ Q is a *DTW* path $\wedge$ Q is a *DTW* path $\implies$ Q is a D-path. Both D-path and *DTW* path are subsets of

$$\{(i, j) | 1 \leq i \leq n \wedge 1 \leq j \leq m\}.$$

Therefore, it is sufficient just to show that the constrains are equivalent.

- The boundary conditions are the same by definition.
- Shift constraint implies continuity. If $p_k = (i, j)$ then $(i', j') = p_{k+1} \in \{(i+1, j), (i, j+1), (i+1, j+1)\}$ (shift constraint), then $i' - i \leq 1$ and $j' - j \leq 1$ for $1 \leq k < l$, thus the continuity holds.
- Shift constraint implies monotonicity. If $p_k = (i, j)$ then $(i', j') = p_{k+1} \in \{(i+1, j), (i, j+1), (i+1, j+1)\}$ (shift constraint), then $i' - i \geq 0$ and $j' - j \geq 0$ for $1 \leq k < l$, so the monotonicity holds.
- Continuity and monotonicity implies the shift constraint. If $w_{k+1} = (i, j)$ then $w_k = (i', j')$, and $0 \leq i' - i \leq 1$ and $0 \leq j' - j \leq 1$ for $l > k \geq 1$ (Continuity

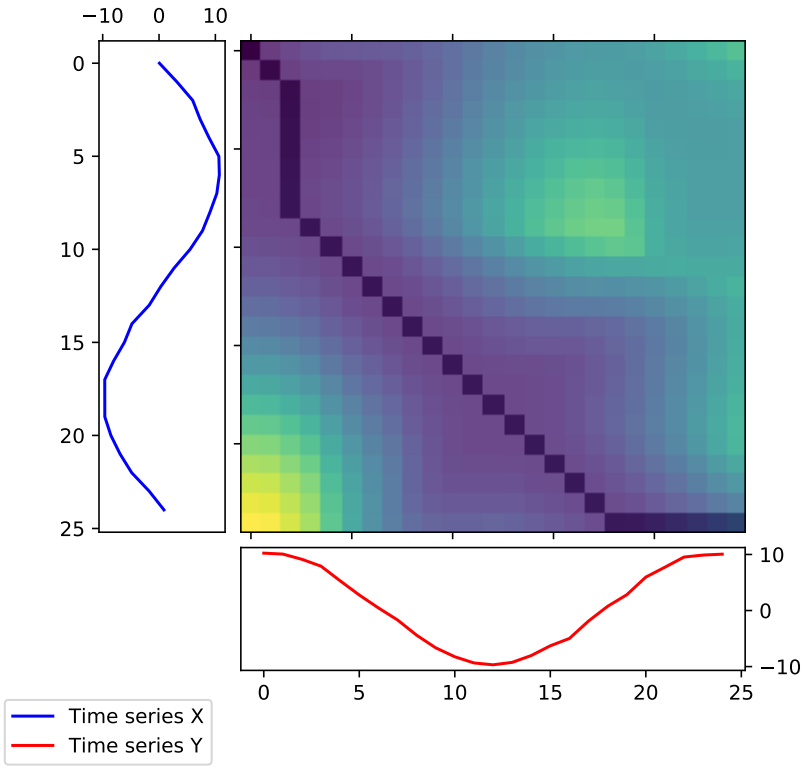


Figure 2.4 A D-path of two time series

and monotonicity). This implies that the available options are $i^{'} \in \{i, i+1\}$ and $j^{'} \in \{j, j+1\}$, therefore $(i^{'}, j^{'}) \in \{(i+1, j), (i, j+1), (i+1, j+1)\}$. Thus, the shift constraint holds.

In the Figure 2.4 is a D-path (the black line) of two time series, we can see that it starts in $(0, 0)$ and ends in the opposite corner. The shift condition is also satisfied. The color of (i, j) indicates the sum of the minimal $D-path$ to the (i, j). The lower number, the darker color is on (i, j). We can see that the D-path goes through the darkest area. The 2.4 Lemma showed that we can calculate the minimal DTW path using a D-path because they are equivalent and also the costs of the mappings (i, j) are the same. We mean the value of D-path as the sum of all elements on the path through the matrix M. The determination of the minimal D-path is straightforward. The only possible moves to (i, j) are from $(i-1, j)$, $(i, j-1)$, $(i-1, j-1)$. To figure out the optimal (i, j) it is sufficient to get the minimum of the predecessors. When one of the time series has a

length equal to 0, and the other is nonempty, we define the distance as infinity [23].

$$DTW(X,Y) = \begin{cases} 0 & \text{if } m = n = 0, \\ \infty & \text{if } m = 0 \text{ or } n = 0, \\ d(x_1, y_1) + \min \begin{cases} DTW(X, tail(Y)), \\ DTW(tail(X), Y), \\ DTW(tail(X), tail(Y)) \end{cases} & \text{otherwise,} \end{cases}$$

where $tail(X)$ is a function that takes a time series and returns the same time series without the first element. The $DTW(X, tail(Y))$, etc., can be envisioned as possible predecessors.

We show that this is the minimal distance denoted as s. For contradiction, we assume that there is another $DTW(X,Y) = z < s$. Without loss of generality, we suppose that the predecessor of the last element (i,j) is $(i,j-1)$ in z. If we remove the last element from z we get $z - d(x_1, y_1) < DTW(X, tail(Y))$ and it is a contradiction. The arguments are the same for other predecessors. It implies that s is the minimal one.

Algorithm 1: *DTW*

Input : Time series $X = x_1, x_2, \ldots, x_n$, $X = y_1, y_2, \ldots, y_m$
Output: Dynamic time warping distance between X and Y
M = matrix $(n+1) \times (m+1)$ full of infinities;
M[1,1] = 0;
if *n = 0 and m = 0* **then**
  return 0;
end
if *n = 0 or m = 0* **then**
  return ∞;
end
for $i \leftarrow 2$ **to** $n+1$ **do**
  for $j \leftarrow 2$ **to** $m+1$ **do**
    $M[i,j] = min\{M[i-1,j], M[i,j-1], M[i-1,j-1) + d(x_{i-1}, x_{j-1}\}$;
  end
end
return M[n+1, m+1];

The infinities in the first row and the first column guarantee that the values will not be used in the *min*.

2.2.2 Longest common subsequence similarity

The longest common subsequence similarity ($LCSS$) is widely used on strings to measure string similarity. The $LCSS$ is a similarity measure, so the higher number, the more alike are the time series. The basic idea is to find time series subsequences that are similar. This allows (unlike the DTW) some points not to be mapped to any other (e.g., outliers), so it is more robust to an outlier element in a time series. Every point is mapped at most once to another point, so the upper bound of the similarity value is the length of the shorter time series. At first, we define the longest common subsequence (LCS) for strings.

▶ **Definition 2.5** (*LCS* for strings)**.** *A subsequence S_G of a string $G = g_1, g_2, \ldots, g_n$ of length n is a sequence $g_{p_1}, g_{p_2}, \ldots, g_{p_l}$ of length l, where every $p_i \in \{1, \ldots, n\}$ and $p_1 < p_2 < \ldots < p_l$. In other words, a subsequence S_G of G can be obtained by deleting some elements in G and preserving the order of the others. Let $A = a_1, a_2, \ldots, a_n$ and $B = b_1, b_2, \ldots, b_m$ be strings of lengths n and m respectively and let Σ be an alphabet, $a_i, b_j \in \Sigma$ for every $1 \leq i \leq n, 1 \leq j \leq m$. The LCSS of strings A and B is the longest subsequence S_A of A that is also subsequence of B [24].*

For example, let $A =$ "$BCDAACD$" and $B =$ "$ACDBAC$", then common subsequences are "BC", "AAC", "DAC", "$CDAC$", *etc.* The longest common subsequence is "$CDAC$".

Although time series, in general, can not be imagined as strings (discrete data indexed by $\mathbb{N}$) we can apply the *LCSS* to the time series. Although it would work (even for $\mathbb{R}$), it is handy to introduce a threshold ϵ and consider two values x_1, x_2 similar when $|x_1 - x_2| \leq \epsilon$. To define the *LCSS* for time series, we first introduce a subsequence of time series and an ϵ-match.

▶ **Definition 2.6.** *Subsequence of a time series $X = x_1, x_2, \ldots, x_n$ with length of n is $S_X = x_{p_1}, x_{p_2}, \ldots, x_{p_l}$ of length l where every $p_i \in \{1, \ldots, n\}$ and $p_1 < p_2 < \ldots < p_l$.*

▶ **Definition 2.7.** *Time series $X = x_1, x_2, ..., x_n$ and $Y = y_1, y_2, \ldots, y_n$ of the same length n have a ϵ-match if and only if $|x_i - y_i| \leq \epsilon$ for every $1 \leq i \leq n$.*

▶ **Definition 2.8.** *The LCSS similarity between time series X and Y is*

$$LCSS(X, Y) = |LCS(X, Y)|,$$

where $LCS(X, Y) = s_{p_1}, s_{p_2}, \ldots, s_{p_l}$ is a subsequence of X of length l that has a ϵ-match with a subsequence of Y of the same length l (we will call it CS) that is also the longest one. Function $|\cdot| : \Sigma^* \to \mathbb{N}_0$ returns length of given string. Σ^* is a set of all finite string on alphabet Σ.

We need to find the longest one for a given threshold ϵ, first, we need a lemma to introduce the algorithm.

▶ **Lemma 2.9.** *Let $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$ be time series of lengths n, m respectively and $S = s_1, s_2, \ldots, s_l$ be a $LCS(X, Y)$. [25]*

1. *if $|x_1 - y_1| \leq \epsilon$ then $s_1 = x_1$,*
2. *if $|x_1 - y_1| > \epsilon$ then $|S| = \max\{LCSS(tail(X), Y), LCSS(X, tail(Y))\}$.*

Proof. 1. Assume for the purpose of contradiction that $s_1 \neq x_1$ then we can prepend x_1 to S and get a longer subsequence. It is not possible since S is a *LCS*.

2. If we add entries to X or Y at certain positions and retain the order then the *LCSS* cannot decrease, therefore

$$|S| \geq \max\{LCSS(tail(X), Y), LCSS(X, tail(Y))\}.$$

There is no mapping between x_1 and y_1 thus, the only possibilities are mappings $X \leftrightarrow tail(Y)$ and $tail(X) \leftrightarrow Y$, so

$$|S| \leq \max\{LCSS(tail(X), Y), LCSS(X, tail(Y))\}.$$

◀

For 2 time series where at least one of them has length 0, then the $LCSS$ is 0. If we have a time series X and Y of length n, m respectively and we have calculated $LCSS(tail(X), Y)$, $LCSS(X, tail(Y))$ and $LCSS(tail(X), tail(Y))$, then we can calculate $LCSS(X, Y)$.

1. if $|x_1 - y_1| \leq \epsilon$ then $LCSS(X, Y) = LCSS(tail(X), tail(Y)) + 1$. We have constructed a CS of X and Y with length $LCSS(tail(X), tail(Y)) + 1$ by prepending x_1 to the $LCS(X, Y)$. We must show that it is a $LCSS(X, Y)$. For the purpose of contradiction, we assume that there is another CS Z of X and Y with length $|Z| > LCSS(tail(X), tail(Y)) + 1$. When we remove x_1 from Z (it is there, according to Lemma 1), then we get a CS of tail(X) and tail(Y) that is longer than the $LCSS(tail(X), tail(Y))$: $LCSS(tail(X), tail(Y)) < |Z| - 1$. This is not possible.
2. if $|x_1 - y_1| > \epsilon$ then $LCSS(X, Y) = \max\{LCSS(X, tail(Y)), LCSS(tail(X), Y)\}$. Consequence of the Lemma 2.

Therefore, the recursive formula is:

$$LCSS(X, Y) = \begin{cases} 0 & \text{if } n = 0 \text{ or } m = 0, \\ LCSS(tail(X), tail(Y)) + 1 & |x_1 - y_1| \leq \epsilon, \\ \max\{LCSS(X, tail(Y)), LCSS(tail(X), Y)\} & |x_1 - y_1| > \epsilon. \end{cases}$$

Algorithm 2: *LCSS*

Input : Time series $X = x_1, x_2, \ldots, x_n$, $X = y_1, y_2, \ldots, y_m$
Output: Longest common subsequence of X and Y
M = matrix $(n + 1) \times (m + 1)$ full of zeros;
M[1,1] = 0;
if *n = 0 or m = 0* **then**
  return 0;
end
for $i \leftarrow 2$ **to** $n + 1$ **do**
  for $j \leftarrow 2$ **to** $m + 1$ **do**
    if $|x_1 - y_1| \leq \epsilon$ **then**
      $M[i, j] = M[i - 1, j - 1] + 1$;
    else
      $M[i, j] = max\{M[i - 1, j], M[i, j - 1]\}$;
    end
  end
end
return M[n+1, m+1];

In the future sections, we will use an algorithm that requires a distance measure. Therefore, the $LCSS$ can not be used at least with the current definition. We will define the longest common subsequence distance $LCSD$ with the $LCSS$ definition for X and Y as follows:

$$LCSD = 1 - \frac{LCSS(X,Y)}{\min\{n,m\}}.$$

The triangular inequality does not hold for the $LCSD$, thus it is not a metric. The equivalence in the positive definiteness axiom is also not satisfied due the presence of the ϵ. We can have two time series of length n and the $LCSD$ equal to zero, but the time series are not the same. The last symmetry axiom holds. We can see that for two time series with high longest common subsequence similarity, the longest common subsequence distance is low and vice versa.

2.2.3 Edit distance with real penalty

The Edit distance with a real penalty (ERP) was originally defined in this paper [26] as a combination of multiple distance measures to get the best of each. The ERP is a dissimilarity measure and the first one presented fulfills the positive definiteness, symmetry, and triangular inequality, thus it is a metric. Like the $LCSS$, the ERP is also based on a string dissimilarity measure called the Edit distance (ED), sometimes called the Levenshtein distance. To define the ERP, we first define the ED.

▶ **Definition 2.10** (Edit distance for strings)**.** *Let $A = a_1, a_2, \ldots, a_n, B = b_1, b_2, \ldots, b_m$ be strings of lengths n and m respectively and let Σ be an alphabet, $a_i, b_j \in \Sigma$ for every $1 \leq i \leq n, 1 \leq j \leq m$. Edit distance between A and B is the minimal number of insert, delete, and change symbol operations needed to make string B from string A. We denote it as $ED(A,B)$.*

For example, the edit distance between strings $KITTEN$ and $SITTING$ is 3. The operations are change, change, and insert:

$$KITTEN \to \mathbf{S}ITTEN \to SITT\mathbf{I}N \to SITTIN\mathbf{G}.$$

The operations have equal cost, namely, 1. This feature fits to strings because we are usually not interested in how similar are the symbols a and b. When working with time series it is different, we normally are interested in how similar are the time series entries x and y. The ERP takes this into account in contrast to the string ED.

▶ **Definition 2.11.** *Let X and Y be time series and an operation o is a function that takes a time series and returns the time series after the operation. Possible operations are insert, delete, and change of an entry. Let C be a function that takes an operation o and returns a cost. If the operation o is delete or insert of an element x, then $C(o) = |x|$, when the operation is change of x to y then $C(o) = |x - y|$. The ERP between X and Y is:*

$$ERP(X,Y) = \min\left\{\sum_{o \in O} C(o) \middle| O \text{ is a sequence of operations that transforms } X \text{ to } Y\right\}.$$

At first, we create a recursive function and show that it computes the *ERP*, and based on this recurrence, we introduce the dynamic programming algorithm that computes the *ERP*.

▶ Observation 2.12. Operations are applied to an entry independently of others. There is always a smallest *ERP* that applies at most one operation to an entry [27].

Based on this observation, we can apply the operations from left to right, thus there is always the last operation on the left, that is, one of insert, delete, or change.

If we have 2 time series and at least one of them has length 0, then the *ERP* is the sum of all entries in the other one (inserting, resp., deleting all elements). Suppose time series X, Y, and we have calculated the $ERP(X, tail(Y))$, $ERP(tail(X), Y)$, $ERP(tail(X), tail(Y))$. The recursive function is:

$$ERP(X,Y) = \begin{cases} \sum_{i=0}^{n} |x_i| & \text{if } m = 0, \\ \sum_{i=0}^{m} |y_i| & \text{if } n = 0, \\ \min \begin{cases} ERP(tail(X), Y) + |x_1|, \\ ERP(X, tail(Y)) + |y_1|, \\ ERP(tail(X), tail(Y)) + |x_1 - y_1| \end{cases} & \text{otherwise.} \end{cases}$$

We need to show that it is the minimal one (so it is the *ERP*), denote our result as s. We assume that there is another $ERP(X,Y) = z < s$. In z there is also the last operation, at first, we assume it is a delete. When we remove the delete operation, we get $z - |x_1| < ERP(X, tail(Y))$ and it is not possible. For other operations, the argumentation is the same. Thus $ERP(X,Y) = s$.

Algorithm 3: *ERP*

Input : Time series $X = x_1, x_2, \ldots, x_n$, $X = y_1, y_2, \ldots, y_m$
Output: Edit distance with real penalty between X and Y
M = matrix $(n+1) \times (m+1)$ full of infinities;
M[1,1] = 0;
M[1,2] = 0;
M[2,1] = 0;
if *n = 0* **then**
  return $\sum_{i=1}^{m} |y_i|$;
end
if *m = 0* **then**
  return $\sum_{i=1}^{n} |x_i|$;
end
for $i \leftarrow 2$ **to** $n+1$ **do**
  for $j \leftarrow 2$ **to** $m+1$ **do**
    $M[i,j] =$
     $min\{M[i-1,j] + |x_{i-1}|, M[i,j-1] + |y_{j-1}|, M[i-1,j-1] + |x_{i-1} - y_{j-1}|\}$;
  end
end
return M[n+1, m+1];

The infinities in the first row and the first column guarantee that the values will not be used in the *min*.

2.2.4 Discrete Fréchet distance

The Discrete Fréchet distance (*DFD*) is a distance measure based on the (nondiscrete) Fréchet distance. It is a metric, so it satisfies all of the 3 axioms. The intuitive definition of the distance is that a man is walking a dog on a leash. The dog and the man have their own curves. Both can go only forward, and their speed may vary, but neither can go backward. The distance is the length of the shortest leash necessary for the man to walk the dog from the starting points to the end points of the curves [28]. The discrete variant considers just a finite set of points on the curve. Thus the formulation is different: How to walk a frog if you are a frog yourself [29].

▶ **Definition 2.13.** *[28] Let X and Y be time series of lengths n resp. m and let $M \subseteq \{(i,j)|x_i \in X, y_j \in Y\}$. M fulfills:*

1. *If $(i,j) \in M$ then $(p,q) \notin M$ for $i < p \wedge q < j$ or for $i > p \wedge q > j$.*
2. *For every $i, 1 \leq i \leq n$ (resp. $j, 1 \leq j \leq m$) there is a $(i,z) \in M$ (resp. $(z,j) \in M$).*

The Discrete Fréchet distance is [28]:

$$DFD(X,Y) = \min_{M} \max_{(i,j)\in M} |x_i - y_j|.$$

The first condition in Definition 2.13 ensures that the man, nor the dog, can not go backward. The second one says that both must go through all points in their path.

In Figure 2.5 is an illustration of one element (i,j) of the M set. There are two time series X and Y, a leash between indexes i and j is denoted with a black dashed line, but the length of the leash is just the absolute value of the difference of y coordinates, and this is denoted with the orange line.

We will create a recursive formula, and based on the formula, we introduce an algorithm. At first, the base case: we define the dissimilarity measure just for nonempty time series. Let $X = x_1$ be a time series of length 1 and Y of length m. The distance is then $\max_{y_i \in Y} |x_1 - y_i|$. The arguments for the induction step are similar to the *ERP*. Suppose two time series X and Y of lengths n and m respectively and we have calculated the $DFD(X,Y)$, so we know the optimal moves on the curves. The last move to the state when both are in the end points (i,j) can be just one of: $(i-1,j)$, $(i,j-1)$, and $(i-1,j-1)$. Therefore, if we assume we have calculated the $DFD(X, tail(Y))$, $DFD(tail(X), Y)$, $DFD(tail(X), tail(Y))$ we can calculate:

$$DFD(X,Y) = \begin{cases} \max\limits_{x_i \in X} |x_i - y_1| & \text{if } m = 1, \\ \max\limits_{y_i \in Y} |x_1 - y_i| & \text{if } n = 1, \\ \max\left\{ d(x_1,y_1), \min\left\{ \begin{array}{l} DFD(X, tail(Y)), \\ DFD(tail(X), Y), \\ DFD(tail(X), tail(Y))) \end{array} \right\} \right. & \text{otherwise.} \end{cases}$$

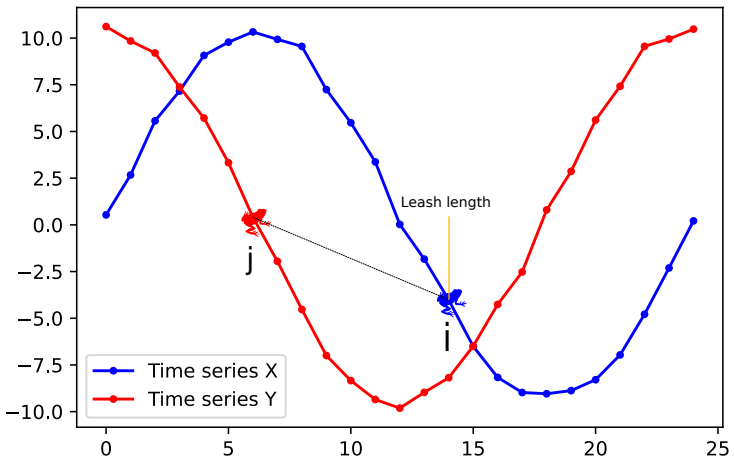


Figure 2.5 An illustration of a state when calculating the Fréchet distance

We have to show that it is the minimal one and therefore the DFD, we denote our result as s. We assume, for the purpose of contradiction, that there is a $z = DFD(X, Y) < s$. In z there is the last move, and if we remove the last move, then we get a lower distance than in the assumptions, and it is a contradiction. It implies that $s = DFD(X, Y)$. This recursive formula gives us a dynamic programming algorithm:

Algorithm 4: *DFD*

Input : Nonempty Time series $X = x_1, x_2, \ldots, x_n$, $X = y_1, y_2, \ldots, y_m$
Output: Discrete Fréchet distance between X and Y
M = matrix $(n + 1) \times (m + 1)$ full of infinities;
M[1,1] = 0;
for $i \leftarrow 2$ **to** $n + 1$ **do**
  for $j \leftarrow 2$ **to** $m + 1$ **do**
    $M[i, j] = \max\{d(x_i, y_i), min\{M[i - 1, j], M[i, j - 1], M[i - 1, j - 1]\}\}$;
  end
end
return M[n+1, m+1];

The infinities in the first row and the first column guarantee that the values will not be used in the min.

All of these algorithms try all combinations, and some of them are not interesting for us. For example, a mapping $(1, i), 1 \leq m$ and $(i, m), 1 < i \leq n$ in the DTW when mapping time series X and Y of lengths n and m respectively. We can use this observation to speed up the computation by limiting some of the mappings. The most known options are the Sakoe-Chiba band and Itakura parallelogram [30] shown in Figure 2.6. There is a square matrix of size 64 (thus for two time series of length 64) representing the dynamic programming matrix when calculating the measures. The black area in the figures represents the options that the algorithms try. They introduce a new hyperparameter r that prunes some of the options. These options are primarily used

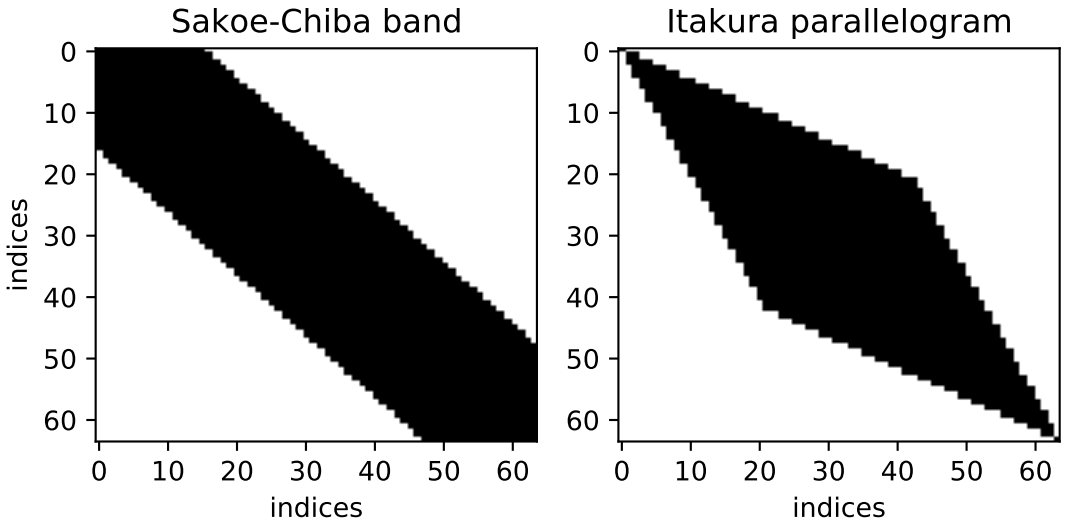


Figure 2.6 Constrains of measures

when calculating the DTW but can also be used in calculating other distance measures. There is a trade-off between the optimal result and speed, and it is not always easy to choose the right r. We decide not to use these constraints because when we lower the dimension of the time series with the PAA (introduced in another section in this chapter), then the speed is not a major issue for us. We mention it here just for completeness.

2.3 Time series clustering

The data clustering is one of the main machine learning disciplines. When we have a data set D of points in a space, then the data points are usually not uniformly distributed in the space of all possible values. There are some regions where there is a higher probability of point occurrence, this information is valuable for us because it reveals the inner structure of the data. Clustering of data works with this observation. The task is to split the data set D into k disjoint clusters, and an element is as much as possibly similar to elements in the same cluster and distinct to other elements in different clusters, more formally [19]:

$$C = C_1, C_2, \ldots, C_k \text{ is a set of mutually disjoint and is fulfilled that: } \bigcup_{i=1}^{k} C_i = D.$$

The data clustering is an unsupervised learning method where we do not know how to assess the output. We just want to get some information about the inner structure of the data distribution. To determine how similar are two points $x, y \in D$, we need a distance measure defined in Section 2.2. Clustering time series is just a special type of data clustering, we need just a corresponding distance measure. To summarize what is a clustering algorithm [19]:

Input: Data set D and a distance measure d, sometimes number of clusters k.

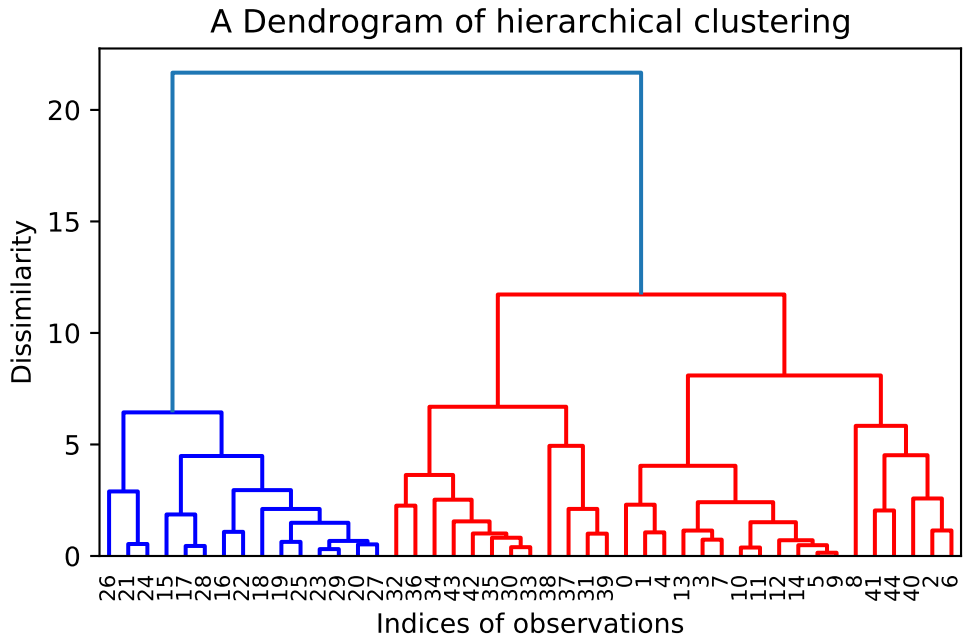


Figure 2.7 A dendrogram that shows a run of a hierarchical algorithm

Output: A label i for each $x \in C_i$, where $C = C_1, C_2, \ldots, C_k$ is a set of pairwise disjoint clusters that satisfies $\bigcup_{i=1}^{k} C_i = D$

We will use the agglomerative hierarchical clustering algorithm. The output of this algorithm is a clustering when we specify a stop condition like a number of clusters or a distance upper bound for merging. The process of merging clusters can be visualized using a graph called a dendrogram. It can be seen in Figure 2.7. The graph is a tree and each node represents a cluster. The leafs are the original points in the data set D, and each inner node represents a union of two clusters. The y axis shows a dissimilarity value. The height of a horizontal line represents the distance between two merged clusters. Some nonsingleton clusters that are under a certain threshold are colored. If a nonsingleton node k is directly under the threshold t, then all of its descendants are colored with the same color. All nodes above t have the same color, also including singleton clusters. In Figure 2.7 the threshold can be for example 15. Two vertical lines show which clusters are merged. There is a pseudo code of the algorithm [19]:

Algorithm 5: AgglomerativeHierarchicalCluster

Input : Data set D of length n, a distance measure d, and number of clusters $k > 0$

Output: A dendrogram

Insert each element i in D as a singleton cluster with 1 member into C as C_i;

for $i \leftarrow 1$ **to** $n - k$ **do**

  Find clusters $C_k, C_l \in C, k \neq l$ that are the closest by the distance measure d;

  Merge C_k, C_l into new cluster C_{n+i};

  Remove C_k, C_l from C;

  Insert C_{n+i} to C;

end

return Dendrogram from the run of the for loop and the clustering C;

We are searching for the closest clusters in the algorithm, but we have just defined some distances between cluster elements. There are multiple ways how we can define a distance between clusters [19].

Let A, B be two nonempty clusters of time series, and $D(A, B)$ is a distance between clusters A and B. There are some examples [19]:

Single linkage: The distance between the closest time series in different clusters. Generates clusters that look like long chains

$$D(A, B) = \min_{x \in A, y \in B} d(x, y).$$

We can see in Figure 2.8 on the left side some data points in 2 dimensional Euclidean space and on the right side a dendrogram of the agglomerative hierarchical clustering algorithm that uses the single linkage method. The points on the lines are equally spaced, therefore we can see in the dendrogram that the two clusters are formed at a constant distance of 2 and then merged at a distance 5.

Complete linkage: The distance between the most distant time series in different clusters. Tries to minimize the diameter at each step, therefore it generates compact clusters.

$$D(A, B) = \max_{x \in A, y \in B} d(x, y).$$

In Figure 2.9 we can see the same data points as in Figure 2.8. The dendrogram shows that the clusters are merged by 2, then by 4,...

Average linkage: A compromise of single and complete linkage

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y).$$

Centroid linkage: Distance between the centers of clusters

$$D(A, B) = d(\tilde{x}, \tilde{y}).$$

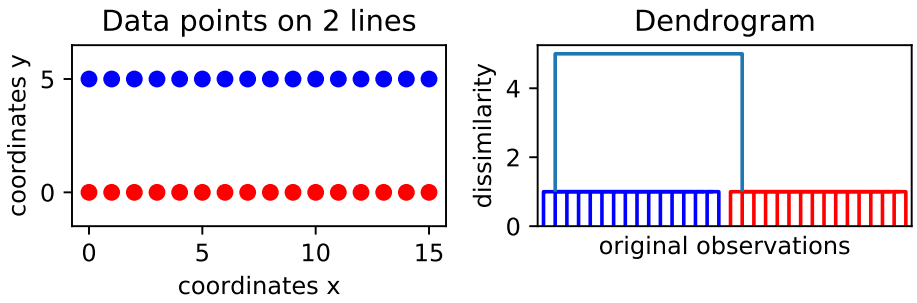


Figure 2.8 An illustration of the hierarchical clustering of data using the single linkage method

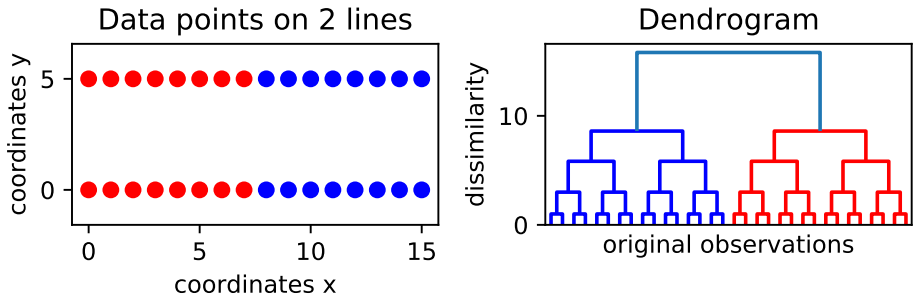


Figure 2.9 An illustration of the hierarchical clustering of data using the complete linkage method

Ward's method: Minimizes the sum of all cluster variances

$$D(A,B) = \sum_{x \in A \cup B} ||x - \tilde{x}_{A \cup B}|| - \sum_{x \in A} ||x - \tilde{x}_A|| - \sum_{x \in B} ||x - \tilde{x}_B||.$$

where $\tilde{x}$ is the center of a cluster X defined as:

$$\tilde{x} = \frac{1}{|X|} \sum_{x \in X} x.$$

When we envision a time series as a vector, then the cluster center makes sense, but just for time series of equal lengths. The sum of vectors with different lengths is ill defined. Different length of time series is our case, so we can not use the centroid linkage and Ward's method, at least with the current definitions.

Selecting the correct linkage is not clear, and it depends on what we want to get. We want the trend evolution of the time series in a cluster to be as much similar as possible (in a way defined by the time series dis/similarity measure). This requirement may not be satisfied for the single linkage as it can create long chains, the criterion is local. On the other hand, the complete linkage is global but very sensitive to outliers [31]. We will use the average linkage because it is something in between, all elements take part in the resulting distance.

We must define what kind of time series (dis)similarity measure the average linkage and thus the algorithm needs. The process just merges the closest clusters at each step. Every function that returns a number for two time series would return a clustering, but the result would be meaningless.

To get a meaningful result, the linkage needs a "distance" measure, therefore a function that for two similar elements returns a small value. On the other hand, for two different elements a large value. The distance measure has to be symmetric to calculate the distance just in one direction (e.g., for two elements x, y just $d(x, y)$ not $d(y, x)$). The triangular inequality and positive definiteness are not required by the linkage and algorithm, although the positive definiteness is quite natural for a distance measure.

2.4 Dimensionality reduction

Time series data have high dimensionality by nature because in one time series there are usually many entries. There are time series databases that are really big and they still grow. When working with big data sets, it is often possible to transform the data to something a lot smaller without losing too much information, it is called dimensionality reduction. For time series, there are plenty of techniques that reduce time series dimensionality like the Discrete wavelet transformation (DWT), Singular value decomposition (SVD), Discrete Fourier transformation (DFT), Adaptive Piecewise Constant Approximation (APCA), Piecewise Aggregate Approximation (PAA), etc. We will use a slightly different version of the PPA. We use the PPA because it is fast to compute, easy to implement, competitive with other approaches, and allows us to use the metrics defined in Section 2.2 compared to, for example, the APCA [32].

The PPA transforms a time series $X = x_1, x_2, \ldots, x_n$ of length n into another time series $\bar{X} = \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N$ of length N, where $1 \leq N \leq n$ with the following equation:

$$\bar{x}_i = \frac{1}{S} \sum_{j=(i-1)\cdot S+1}^{i\cdot S} x_j, \qquad \text{for } 1 \leq i \leq N,$$

where $S = \frac{n}{N}$ is the length of a single segment. In other words, the time series X is divided into N segments, and each consists of S elements, the resulting time series $\bar{X}$ consists of the means of the segments. This notation assumes that n is multiple of N, when it is not, we must take care of this problem.

We will use a slightly different version, instead of specifying the count of segments N, we specify a number of elements in one segment S. If n is not a multiply of S, then we create the last segment as the mean of the rest. For example, when we have $n = 36$ and $S = 8$, then we have 4 segments of 8 elements and 4 as a remainder. We create a $\bar{X} = \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4$, where $\bar{x}_1, \bar{x}_2, \bar{x}_3$ are the means of the complete segments and x_4 is the mean of the remainder with 4 elements. Therefore, the new formula is (just if n is a multiply of N):

$$\bar{x}_i = \frac{1}{S} \sum_{j=S\cdot(i-1)+1}^{i\cdot S} x_j, \qquad \text{for } 1 \leq i \leq N.$$

In figure 2.10 we can see an approximation with the PAA of a time series of length 25 and segment size $S = 3$.

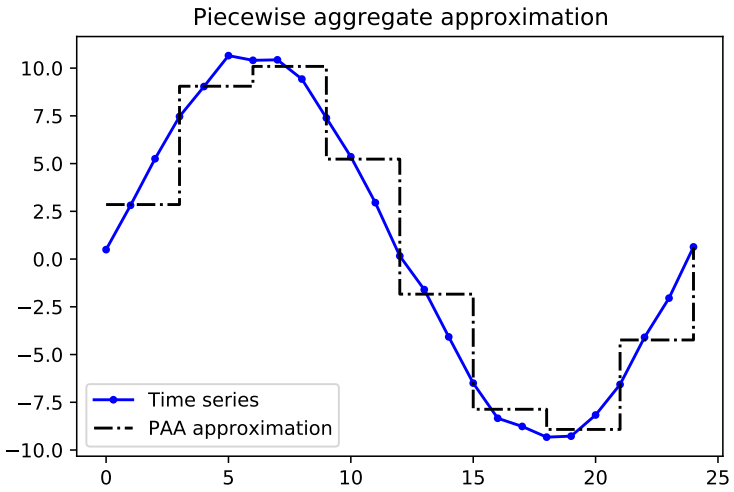


Figure 2.10 A time series of length 25 approximated with the PPA method with segment size of $S = 3$

We will call this modified version as the PAA in the future sections, and the PAA of X is noted as $\bar{X}$. This change of the PAA will help us to work with $\bar{X}$ of constant segment size. Suppose we have a time series X of length 50 and another one Y that has length 500, it is helpful because when we specify the number of segments $N = 10$, then X loses much more information than Y. On the other hand, the original definition of the PAA would help us with time series of different lengths.

2.5 Selecting the number of clusters

There is no automated way to select the number of clusters k for us, but some methods can give us a hint. There are cases where there are more choices on how to select a "good" k or there are data sets where the data does not make any cluster, so it is not always possible to select the "right" k.

The Silhouette score was tried but did not give much information. The score was increasing when we were decreasing the number of clusters and the growth was very smooth. The biggest number was for 2 clusters.

On the other hand, the output of the hierarchical clustering is a dendrogram, therefore we can get some hint of cluster count based on the dendrogram. We will use just the dendrogram to choose.

There are a lot of others like the Calinski-Harabasz criterion and Davies-Bouldin Index, etc. However, some of them require a centroid for a cluster, but it is ill defined in our space.

2.6 Average time series

When we have a cluster of some time series, we want to get a representative time series that shows the basic behavior of the cluster. We will use the average time series defined as follows:

▶ **Definition 2.14.** *Let $S = X^1, X^2, \ldots, X^l$ be some time series of lengths $n_1, n_2, \ldots, n_l$ respectively and let m be the length of the longest time series. Let C_{m-i+1} denote all the time series that has length at least i for all $i \geq 1$:*

$$C_{m-i+1} = \{X^j | X^j \in S, n_j \geq i\}.$$

We define the average time series $AVG(S) = a_1, a_2, \ldots, a_m$ of S as:

$$a_i = \frac{1}{|C_i|} \sum_{X^j \in C_i} X^j_{n_j - m + i},$$

for every $1 \leq i \leq m$.

We can envision the average time series as aligning the time series to the right and summing the columns and dividing by the count of rows, for example, for $S = S^1, S^2, S^3$:

$$\begin{aligned} S^1 &= & 1, 3, 3, 4 & & 1, 3, 3, 4 \\ S^2 &= & 5, 7, 7 & & 5, 7, 7 \\ S^3 &= & 7 & & 7 \end{aligned}$$

$$AVG(S) = \frac{1}{1}, \frac{3+5}{2}, \frac{3+7}{2}, \frac{4+7+7}{3} = 1, 4, 5, 6.$$

Chapter 3

Application to COVID time series

In this section, we will apply all the previous methods to the COVID-19 time series. We start with data preprocessing.

3.1 Data preprocessing

In this thesis, we will only consider the daily new infected people. Thus, we are not interested in deaths and cumulative counts. The *New_cases* and *Data_reported* columns in Table 1.1 together form a time series.

As said in Section 1.2, the data may be updated retrospectively and are completed at the end of the day. The data we use are taken on 14 April 2021. For simplicity and to get rid of the incomplete data, we consider just the records before Sunday, 11 April, 2021. The last day is included.

We are not interested in how long did it take for a country from the first COVID-19 report to the first infected person. We are interested in the diverse spread of the disease in time. Therefore, we do not want to have a time series that starts with any leading zeros. In the first step, we delete all records that have cumulative cases equal to zero. It means that we delete records before the first confirmed case. This also deletes countries with zero reported COVID-19 infections. The deleted countries are American Samoa, Cook Islands, Democratic Peoples Republic of Korea, Kiribati, Micronesia (Federated States of), Nauru, Niue, Palau, Pitcairn Islands, Tokelau, Tonga, Turkmenistan, Tuvalu. It contains small states where there are no infected people and countries that do not report any cases.

The records with negative new case counts described in Section 1.2 are considered insignificant and replaced with zeros with 1 exception. Puerto Rico had 37567 total cases and removed 32952 of them on 8 November 2020. We will not take this country into account as it has inaccurate data.

Subsequently, we delete all countries that have less than 100 000 inhabitants. Due to their small population, the countries with less than 100,000 inhabitants are not considered in the analyses for their insufficient statistical conclusiveness.

There is a record for a state called “Other”. We will not consider it. The resulting data set contains 189 countries out of 236.

We need the data to have the same scale to be comparable. For example, when we have a daily infected person count in the USA and the Czech Republic, then the number in the USA is much higher, but it does not mean that the pandemic is much worse than in the Czech Republic. In the USA, there are just many more people, so there are more chances for the virus.

Generally speaking, we want to say how good or bad is a "state of the outbreak" at a given time for a country compared to another. What is a "state of the outbreak" and if it is good or bad depends a lot on an investigator's purpose and interest. We will suppose that our "state of the outbreak" is the count of infected people divided by the population size of a country in a given time. We represent the comparison, how it is good or bad, by a difference of these numbers. There are many other options that could be interesting like dividing by population density or surface area, daily deaths with some scaling, and some completely different.

At a certain time, there are some people that are infected and some that are not. We do not have this information, we know just the daily positive tests. We do not consider the reliability of COVID-19 tests, thus we suppose the tests are totally reliable. If we were to test the whole population, then we would get the real infected people count, but the tests are conducted just on a subset of the country population since there are not enough tests. If we test randomly selected individuals, we can get a good estimate after dividing by the number of tests performed. The tests are not conducted on a randomized sample, the tests are conducted on people that come to test themselves. These individuals are more likely to have the virus than those uniformly randomly selected. Therefore, this estimate has a bias after the division. It is hard to estimate how big is the bias, so we will not consider the bias and divide just by the population size.

We scale the data, so we have daily new cases per 100 000 people. We get the size of the population from the *latest reported counts* data set and the columns *Cases - cumulative total per 100000 population* and *Cases - cumulative total* denoted as S_{scaled} and S respectively, by the following formula:

$$\text{population size} = \frac{S \cdot 100\ 000}{S_{scaled}}.$$

Finally, the piecewise aggregate approximation is applied to the time series to get a time series of lower dimension and to extract the trend. We have to select the segment size S. For the majority of the data, there is a weekly period. It makes sense since people are not tested uniformly during the week. The Figure 3.1 shows time series of the last 28 days before Sunday, 11 April 2021, in the USA and Czech Republic, the data are scaled. When comparing the time series, we are not interested in the other components, we are just interested in the evolution of the trend. Therefore, we can remove the other parts. We know that the period of seasonality is 7. The PAA is similar to the moving averages in the way that it takes an average of neighbor elements. When we use the MA^7 on a time series X, and then we take every element MA_i^m, where $i-1$ is a multiply of 7, we get the PAA approximation with segnemst size $S = 7$. Based on this observation, we can view the PAA as the trend–cycle component of the time series X of a lower dimension.

The COVID-19 time series contains a lot of distortion. For example, a significant increase in conducted tests in one day may likely result in an increase of the number of positive tests on that day. It could be caused by an area testing. This increase is an

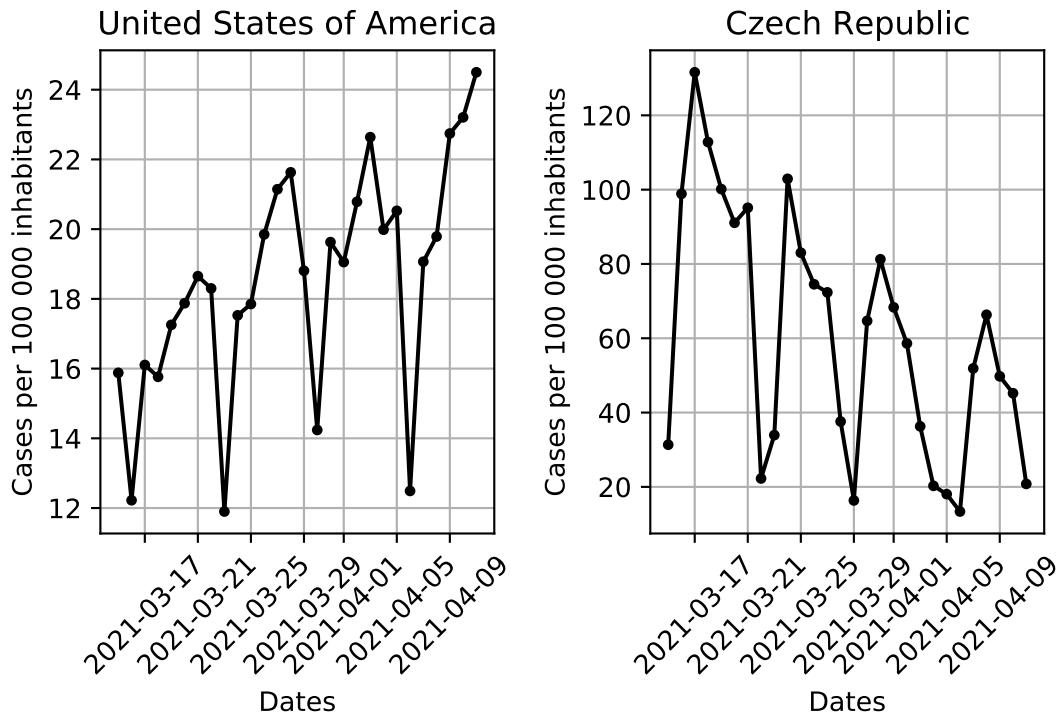


Figure 3.1 Seasonality of the USA and Czech republic

outlier entry in a time series since it is very distant to neighbors. The PAA can partly help with these outlier values since it takes the average of a range, thus the outlier is not as significant.

3.2 Choosing the number of clusters

In the following sections, we will choose the number of clusters for each distance measure. We will use just a dendrogram to select the number of groups since it shows the process of the agglomerative algorithm. The Silhouette score was tried but with a poor information value. Any other techniques are not used since the output is not significantly better than the hand tuned and the choice of the number of clusters depends also on the purpose and interest.

The dendrogram coloring is adjusted to the resulting number of clusters. The x axis contains the original observations, but the names are omitted for the sake of better readability. The detailed clusters (with the names) are included in the Appendix. The time series curves of a single cluster are in a final cluster plot.

In the data set, there are many outlier time series that have a very distinct trend compared to others. We will not delete them, but we will increase the number of clusters to get a more verbose result. The outliers will consume 1 cluster since the distance to others is significant. If we did not enlarge the cluster count, we would end up with one big group, and the rest would be singleton clusters that contain the outlier. The

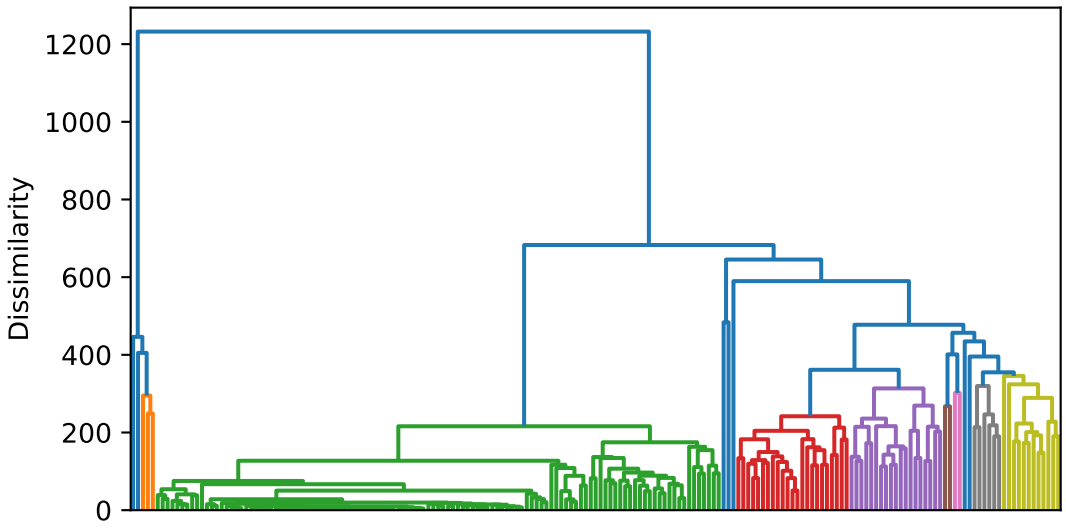


Figure 3.2 The *DTW* dendrogram

resulting plot of the clustering is in the following chapter.

The $\epsilon = 20$ for the *LCSD* was hand-tuned to differentiate the elements well.

3.2.1 *DTW*

The dendrogram for the *DTW* is in Figure 3.2. When we start from the bottom, we can see that some clusters on the left side are merged at shallow values compared to the others. There are just a few other clusters that are formed at the same dissimilarity. When we go a little higher, we can see around the dissimilarity of 200 that the large cluster is merged with a medium size cluster. On the right side, there are relatively many clusters that are linked around the dissimilarity of 150, and then the joins are quite uniformly distributed over the values up to 700. At the level of 700, two significant clusters are merged and at the level of around 1200 is the last merge of a small group and the rest.

We have to find a value of dissimilarity that cuts the dendrogram, and we get the resulting clusters. The green clusters are very similar to each other but very different from the others on the right. Therefore, it would make sense to place them into another group and not divide them. Making this observation on the clusters to the right of the green one is not so straightforward because the joins are distributed uniformly. It would be good to have the red and purple sets alone, so the cut value should be below the join. It seems that the best hand tuned value is 356, which yields 15 clusters that are in Figure 4.5.

3.2.2 *ERP*

The situation in Figure 3.3 is quite the same as in the previous section, but the orange groups are too joined at rather uniformly distributed values over the range. Therefore,

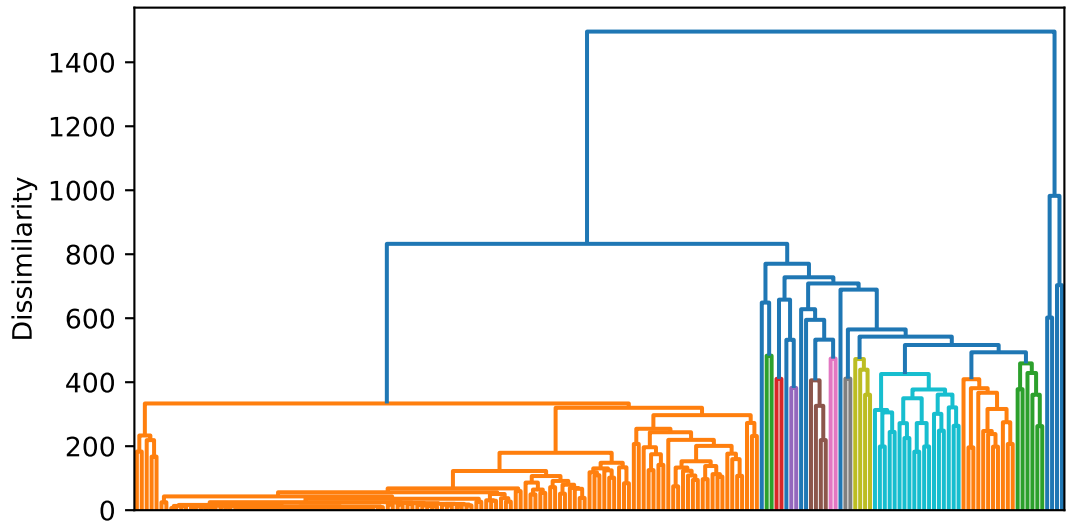


Figure 3.3 The *ERP* dendrogram

selecting the number of clusters is not clear. It would be good to separate the orange cluster to get a more verbose result, but that would yield too many clusters with a lot of singletons. We want the sets merged around the value 800 to be separated because the value is relatively large. However, there are multiple choices on how to select the cut below this upper bound. We choose to keep the blue, orange (on the right), and green clusters separated, so we will select the cut as 493. We get 20 sets that are plotted in Figure 4.6

3.2.3 *DFD*

On the left side of the Figure 3.4 majority of the time series are merged at values close to 0. Most of the values on the right are linked somewhere around 15. The merges are getting sparse very fast as the dissimilarity is increasing. We decide to take the smallest value where most of the clusters are merged, and the joins are relatively rare. We select the value 27 that yields 18 groups that are in Figure 3.4.

3.2.4 *LCSD*

The dendrogram for the *LCSD* is in Figure 3.5. The biggest yellowish group is very homogeneous compared to the others. On the left side, it is common that a big group is merged with a small one, which means the clusters have unique behavior. We want the yellowish and blue colored clusters alone, otherwise the merged cluster is too rough. It is also good to keep the blue clusters linked together because we would otherwise get another small group of elements which looks similar to another cluster. We do not want to have the result too verbose. Based on the requirement, we have just a small interval of values we will pick, for example, 0.06. It yields 28 clusters. The resulting clusters are in Figure 4.8

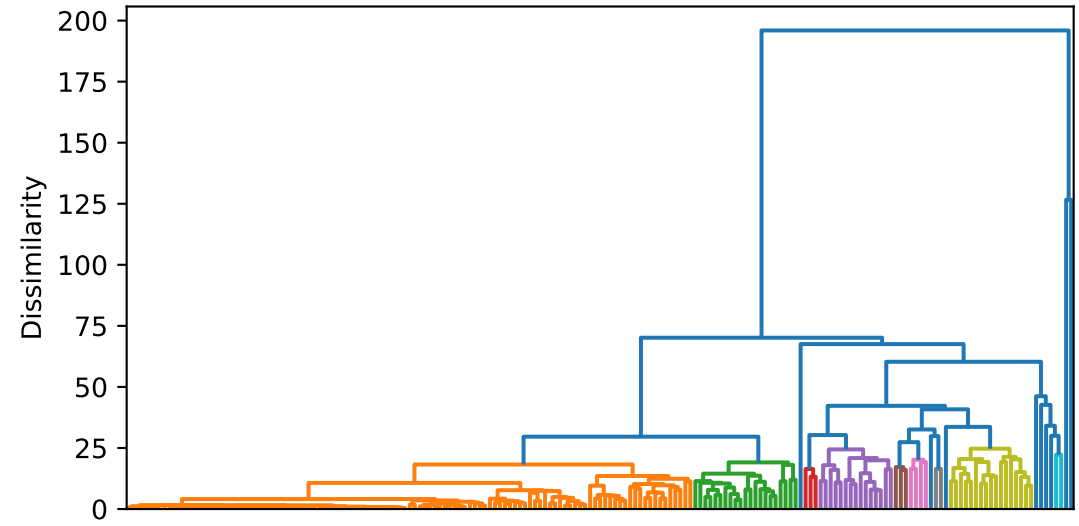


Figure 3.4 The *DFD* dendrogram

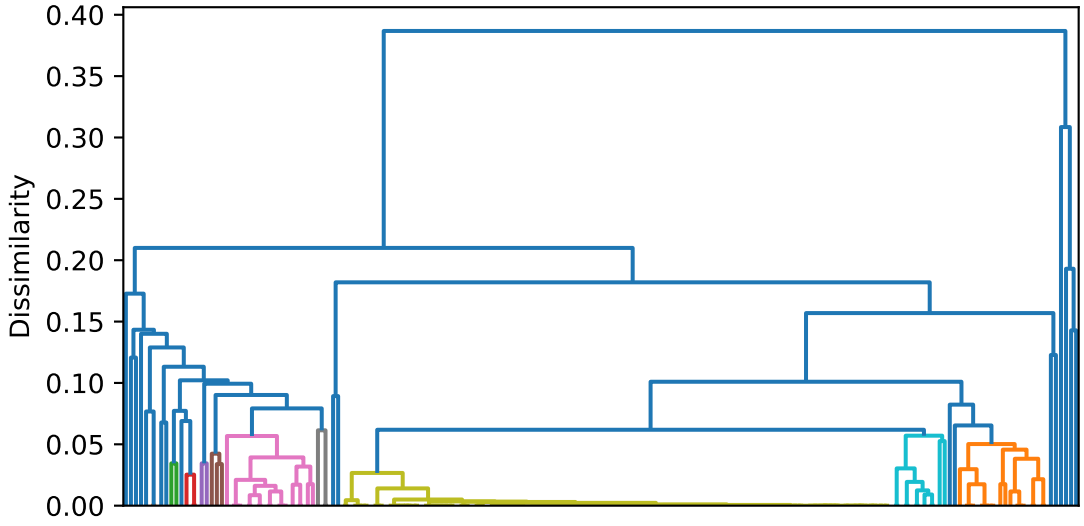


Figure 3.5 The *LCSD* dendrogram

Chapter 4

Discussion and conclusion

This chapter first describes and then discusses and concludes the results of the previous chapter.

4.1 Results

In the following 4 sections, we describe and comment the most significant clusters. There are plots just for these described clusters, plots for the rest are at the end of this chapter with the cluster tables. The clusters in the plots that contain the overall clusters are numbered from 1, starting left to right and then from top to bottom. The cluster numbers can be found in the left column of the tables.

4.1.1 *DTW*

We can see the result in Figure 4.1. The biggest green cluster has 115 time series. Most of the elements in the cluster are linked with a dissimilarity close to zero as we can see in Figure 3.2, so the cluster is very homogeneous. The average line for the group is very close to zero, but there are a lot of time series with some deviations. The outbreak for this cluster was very moderate with some peaks. There are for example China, Finland, Norway, India, Japan and Russian Federation.

The next biggest cluster is the red one with 23 countries, some of them are Canada, Germany, and Greece. The outbreak was a little worse than for the green cluster but still quite mild. There is a tiny wavelet in the average time series with two peaks and the second one seems still increasing. The peaks could indicate the first and second wave. There are also quite a lot of deviations.

The purple cluster has 19 elements like Austria, France, Italy, Netherlands, Poland and Slovakia. The black time series has a much more significant wavelet than the red cluster. The start of the outbreak seems slow but then there is a steep increase that looks like exponential. After the first peak there is a decline of infected people and after some time there is a second wave that is not as steep as the last one. The wave looks decreasing at the end.

The subsequent cluster is the one with yellowish color and 12 elements. There are a lot of deviations that are distinct from the average time series. We can also see from

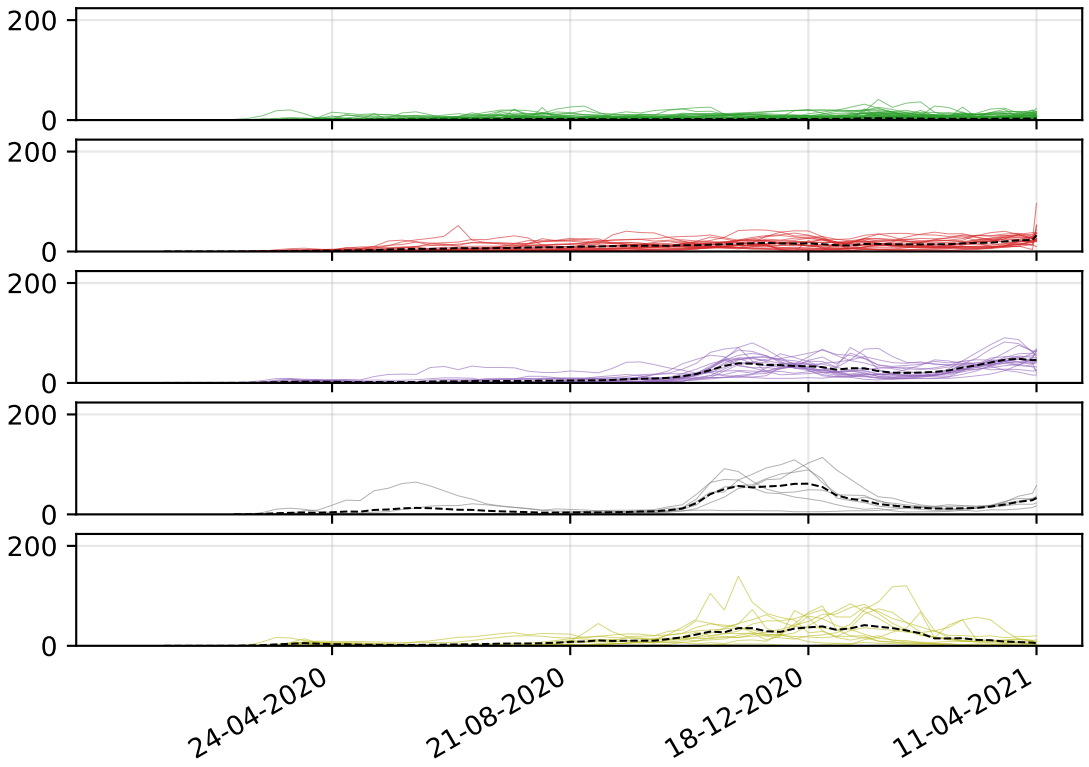


Figure 4.1 The five biggest clusters of the *DTW*

the dendrogram that the clusters are formed at higher values on average compared to the previous groups. The black time series has a slow increase to the highest value, that is constant for relatively long. Then there is a decline in daily infected people. There are two countries that have peaks that are relatively very distant from the average, both peak values are over 100. The peak earlier in time is French Polynesia and the other one is Portugal. This cluster contains for example Denmark, Spain, United Kingdom and United States.

The last cluster that has more than 3 elements is gray. This group is made of Armenia, Switzerland, Georgia, Croatia, Lithuania and Qatar. We can see a small wave close to the beginning of the average time series. Before and after the wave, there are almost no infected people until the second wave that is a lot bigger than the first one. There is a steep decline at the end of the second wave and the values come again close to zero. The end starts to increase.

The other clusters contain at most 3 time series. The clusters have a small number of elements since they have a unique behaviour. For these clusters, it is common to have some relatively high peaks compared to the more numerous groups. The peaks may be caused by a fast spread of the virus but also by a rise of tested people.

The clustered country names can be seen in Table 4.1

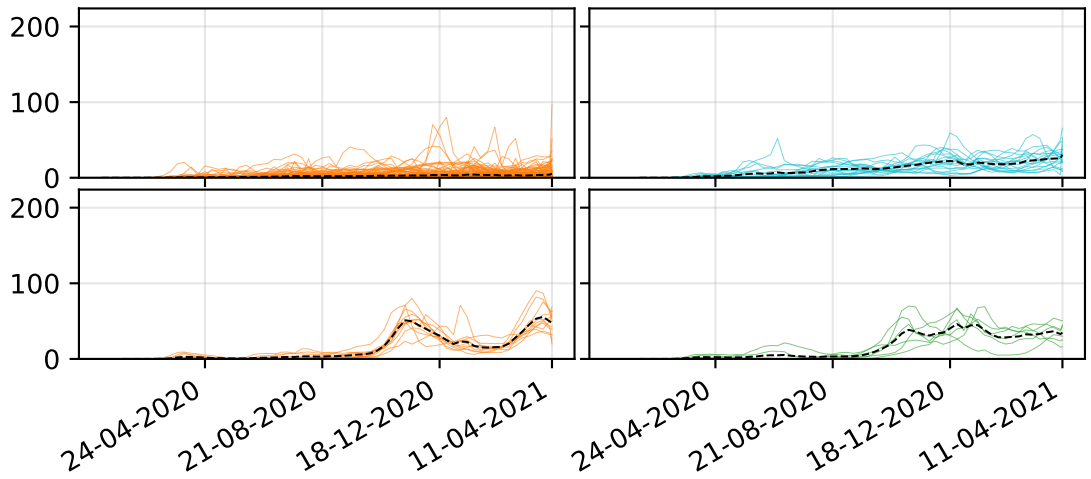


Figure 4.2 The four biggest clusters of the *ERP*

4.1.2 *ERP*

In the dendrogram in Figure 3.3 on the right side, we can see that it is common to merge a big cluster with a small one at relatively large values. This shows that the individual clusters are very different from the rest.

The orange group in Figure 4.2 has daily infected people close to zero on average, but there is a big amount of significant deviations. The set contains 127 members, that is, roughly two thirds. The last high peak is Martinique. Then there are two very similar close to the date 18-12-2020, the left one is Belize and on the right is Jersey. The most significant states are for example Canada, China, Norway, Russia and Greece.

The second biggest cluster has 18 time series, some of them are, for example, Germany and Denmark. We can see that there is a small wavelet with two peaks, the second wave seems to be still increasing. The differences from the black line look smaller than for the orange group. The average time series for these two clusters is moderate.

The orange cluster is very similar to the blue cluster since it has a wavelet, but the wavelet's peaks are higher. The difference is also that the orange cluster seems to decline at the end. The cluster has 11 elements like Austria, France, Italy and Poland.

The last green group with 6 time series has a steep growth, but then it remains constant for some time with some little divergences. There are for example Netherlands, Sweden, and Slovakia.

This metric deems two thirds of the data set relatively alike but a lot of time series from the rest as very distinct from each other. We have 24 clusters and 16 of them with less than 5 members. It is again common for small groups to have high peaks. The tops also often emerge earlier in the time series.

The cluster members are in Table 4.2

4.1.3 *DFD*

For the Discrete Fréchet distance, we have 18 clusters. Four of them are in Figure 4.3. The majority of the links in Figure 3.4 are below the dissimilarity of 20, then the merges

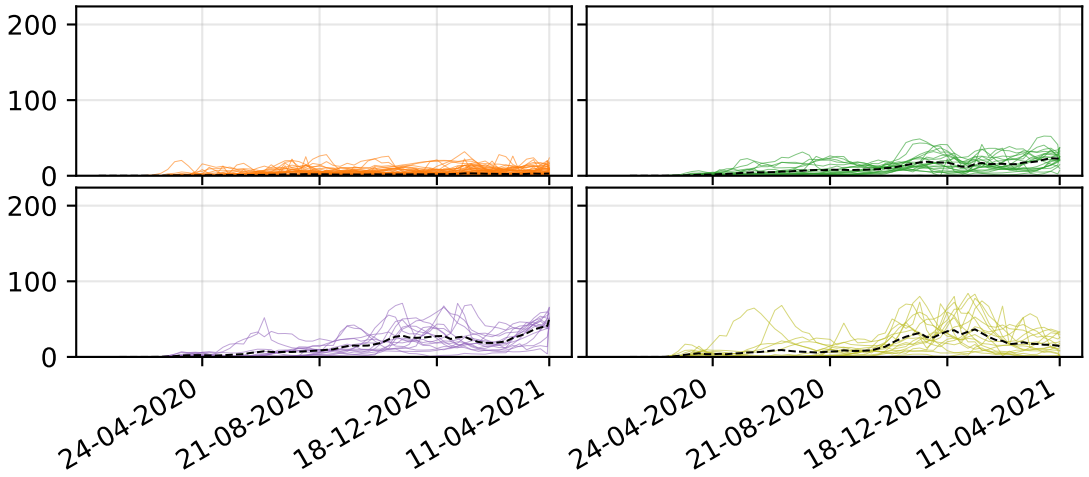


Figure 4.3 The four biggest clusters of the *DFD*

are relatively sparse.

We have again a big cluster that has the average time series very close to zero, but the deviations are quite small. The group has 113 members, that is, more than half. The most significant countries are, for example, China, Finland, and Russia.

The second most numerous group is the green one with 21 time series like Canada, Germany, and Greece. We can see that there is a wavelet and the second peak is still increasing. The tops are not so significant. This cluster is similar to the orange one because they have a common direct predecessor in the dendrogram.

When we look at the group with a yellowish color, we can see that there is a lot of divergence compared to the average time series. This may be because the time series have peaks at different times and they are misaligned. The time series in this cluster have a few steep tops, but the outbreak outside of the tops is mild. This set contains 17 members. There are for example Austria, Denmark, Spain, United Kingdom, Italy, Slovakia and United States.

The next is the purple cluster with 15 time series. The contained countries are for example France, Netherlands, and Sweden. The differences from the black time series are big, the reason may be the same as for the yellowish cluster. The elements in this group reach high peaks as well, but a little less than the yellowish ones.

There are 14 groups out of 20 with just a few elements. The members in one cluster tend to have the same number of tops with the same height and for the same time interval. The countries are in Table 4.3

4.1.4 *LCSD*

We can see the dendrogram in Figure 3.5 and some of the cluster plots in Figure 4.4.

As in all previous dendrograms, there is a big group that has the majority of the elements in the data set. This group contains for example Canada, China, Finland, Greece and Russia. The average time series has again little to no daily infections. All elements in the green set have values close to zero with some thin and low peaks.

There are two clusters with 18 members. We start with the orange one that has for

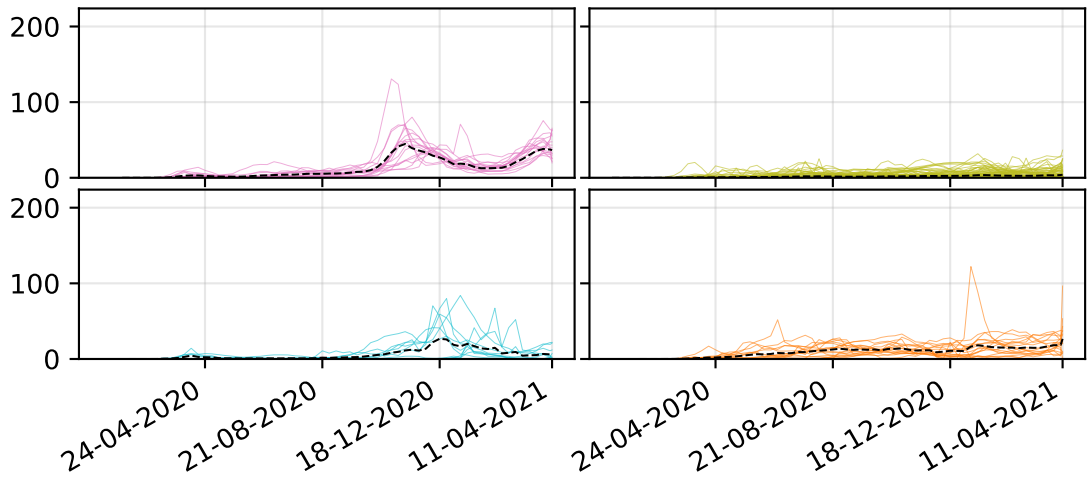


Figure 4.4 The four biggest clusters of the *LCSD*

example Austria, Belgium, France, Italy, and Poland. The average curve is still relatively close to zero and there are some waves, but the tops are insignificant. The time series are quite close to the black one with some exceptions. There are a few outlier entries that are relatively distant to the other values. The highest outlier peak is Ireland. The peak at the end is in Martinique time series.

The other cluster with the same number of elements is the pink one. The average time series starts slowly, but then the curve looks exponential. There is a linear decline after the first top and then another wave. The second peak seems to start to decrease. The cluster is again quite homogeneous but there are some outlier entries. The highest peak in this cluster is Belgium.

The blue set has 11 members. The peak of the average curve is quite thick but low. It looks that there is a linear increase and decrease. The blue curves are homogeneous until the top, but then there are some serious deviations. There are for example Germany, Denmark, and United Kingdom.

We have 28 clusters and 24 of them contain less than 4 elements. Some clusters have time series with an outlier entry. The clusters look very homogeneous. The named time series in each cluster are in Table 4.4.

4.2 Conclusion

All used distance measures have a large cluster that has the majority of the elements. Most of the subclusters in the big one are formed at low dissimilarity values in the dendrograms. The average curves in the plots are very close to zero with no wavelets. Differences between the black line and the time series are quite small, but the *ERP* distance measure is an exception. The states in this cluster had a quite moderate outbreak.

When we have a time series X with a trend, time series Y with the same trend but with a few significant but short time interval deviations and a last curve Z with the same trend but shifted on the y axis. The *ERP* distance seems to tend to find X more similar to Y than X to Z compared to the other distance measures. It is the reason

we have a lot of deviations in the large cluster using the *ERP*. There are time series that are close to zero with some significant peaks and some are just close to zero. This observation can be made with multiple the *ERP* clusters.

There are also 2 clusters with 2 waves on the average time series for each method. The clusters differ in the heights of the waves. For example, the red and purple one for the *DTW*, the blue and the less numerous orange one for the *ERP*. For the *DFD*, it is questionable since the are a lot of deviations and the peaks are not aligned in the same time. These clusters contain states with 2 waves of the outbreak and one cluster has those with a more severe epidemic.

The differences from the average time series for the *DFD* may be caused by the misalignment of the peaks. The distance takes just the y axis into account and thus is not sensitive to the position of the peaks in time. It means that if we have 2 time series with the same wave and out of the wave the values are constant but the waves are in different positions in time, the *DFD* is not sensitive to the positions of the wave. This can be observed in the more numerous clusters for the distance measure.

There are a few sporadic entries that are very distant to the other values and the average line in the *LCSD* measure. The longest common subsequence is able to skip a value that does not participate in the final subsequence. Therefore, the *LCSD* distance is not sensitive to an exceptional outlier entry. For comparison, the *DTW* and *ERP* distance measures always incorporate an entry in a way into the final distance.

For the measures, we have discussed the biggest cluster and the 2 others with 2 waves. There is one other more numerous cluster (the *DTW* has 2), but it does not seem that they have a common pattern or behaviour. We move to clusters with just few elements.

Majority of the clusters are those with just little elements. They are unique in the eyes of the distance measure. The outliers are mostly the ones with relatively high peaks and/or the peaks occurred in different time. For example, the last row in Figure 4.5. The *DFD* measure is less sensitive to the positions of peaks in time, so the majority of outliers are with high and/or thick peaks. There are some countries that are outliers in the perspective of all measures like Israel, Czechia, and Aruba.

The countries that are singletons often have a small population. When the outbreak hits, the peaks are very high due to the scaling with low number of people.

4.3 Further work

It would be interesting to do a similar comparison with the daily death counts time series. This work used almost all countries from the WHO data set that contained a lot of outliers, some specific area analysis could yield some more detailed information.

A better data would give more precise results. Consistent testing strategies and reporting practices over all countries with randomized population samples.

The outputs from this thesis can be further interpreted by someone that has some additional knowledge like a social geographer, epidemiologist, or by someone that knows the introduced measures.

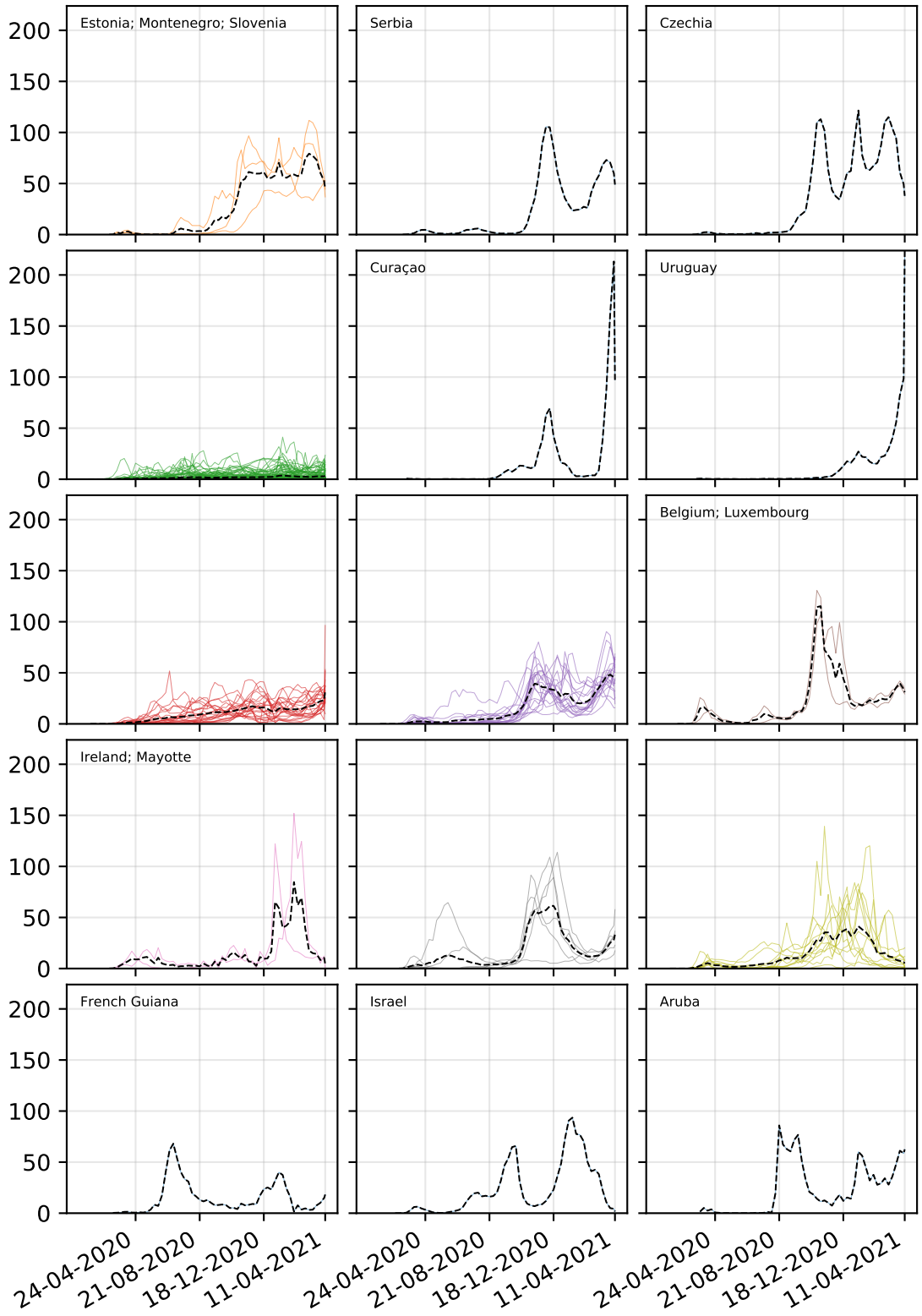


Figure 4.5 The *DTW* cluster plots

Table 4.1 Clusters for the *DTW*

4	Afghanistan; Angola; Australia; Barbados; Bangladesh; Burkina Faso; Burundi; Benin; Brunei Darussalam; Bolivia, Plurinational State of; Bahamas; Bhutan; Botswana; Belarus; Congo, The Democratic Republic of the; Central African Republic; Congo; Côte d'Ivoire; Cameroon; China; Cuba; Djibouti; Dominican Republic; Algeria; Ecuador; Egypt; Eritrea; Ethiopia; Finland; Fiji; Gabon; Grenada; Ghana; Gambia; Guinea; Equatorial Guinea; Guatemala; Guinea-Bissau; Guyana; Honduras; Haiti; Indonesia; India; Iraq; Iceland; Jamaica; Japan; Kenya; Kyrgyzstan; Cambodia; Comoros; Korea, Republic of; Kazakhstan; Lao People's Democratic Republic; Sri Lanka; Liberia; Lesotho; Libya; Morocco; Madagascar; Mali; Myanmar; Mongolia; Mauritania; Mauritius; Malawi; Mexico; Malaysia; Mozambique; New Caledonia; Niger; Nigeria; Nicaragua; Norway; Nepal; New Zealand; Papua New Guinea; Philippines; Pakistan; Réunion; Russian Federation; Rwanda; Saudi Arabia; Solomon Islands; Sudan; Singapore; Sierra Leone; Senegal; Somalia; Suriname; South Sudan; Sao Tome and Principe; El Salvador; Syrian Arab Republic; Eswatini; Chad; Togo; Thailand; Tajikistan; Timor-Leste; Tunisia; Trinidad and Tobago; Tanzania, United Republic of; Uganda; Uzbekistan; Saint Vincent and the Grenadines; Venezuela, Bolivarian Republic of; Virgin Islands, U.S.; Viet Nam; Vanuatu; Samoa; Yemen; South Africa; Zambia; Zimbabwe
7	United Arab Emirates; Albania; Argentina; Azerbaijan; Brazil; Canada; Chile; Colombia; Costa Rica; Cabo Verde; Germany; Guadeloupe; Greece; Iran, Islamic Republic of; Kuwait; Moldova, Republic of; Martinique; Maldives; Oman; Peru; Paraguay; Romania; Ukraine
8	Austria; Bosnia and Herzegovina; Bulgaria; Bahrain; Cyprus; France; Hungary; Italy; Jordan; Lebanon; Latvia; North Macedonia; Netherlands; Poland; Palestine, State of; Sweden; Slovakia; Turkey; Kosovo
12	Belize; Denmark; Spain; United Kingdom; Guam; Jersey; Saint Lucia; Malta; Panama; French Polynesia; Portugal; United States
11	Armenia; Switzerland; Georgia; Croatia; Lithuania; Qatar
1	Estonia; Montenegro; Slovenia
9	Belgium; Luxembourg
10	Ireland; Mayotte
2	Serbia
3	Czechia
5	Curaçao
6	Uruguay
13	French Guiana
14	Israel
15	Aruba

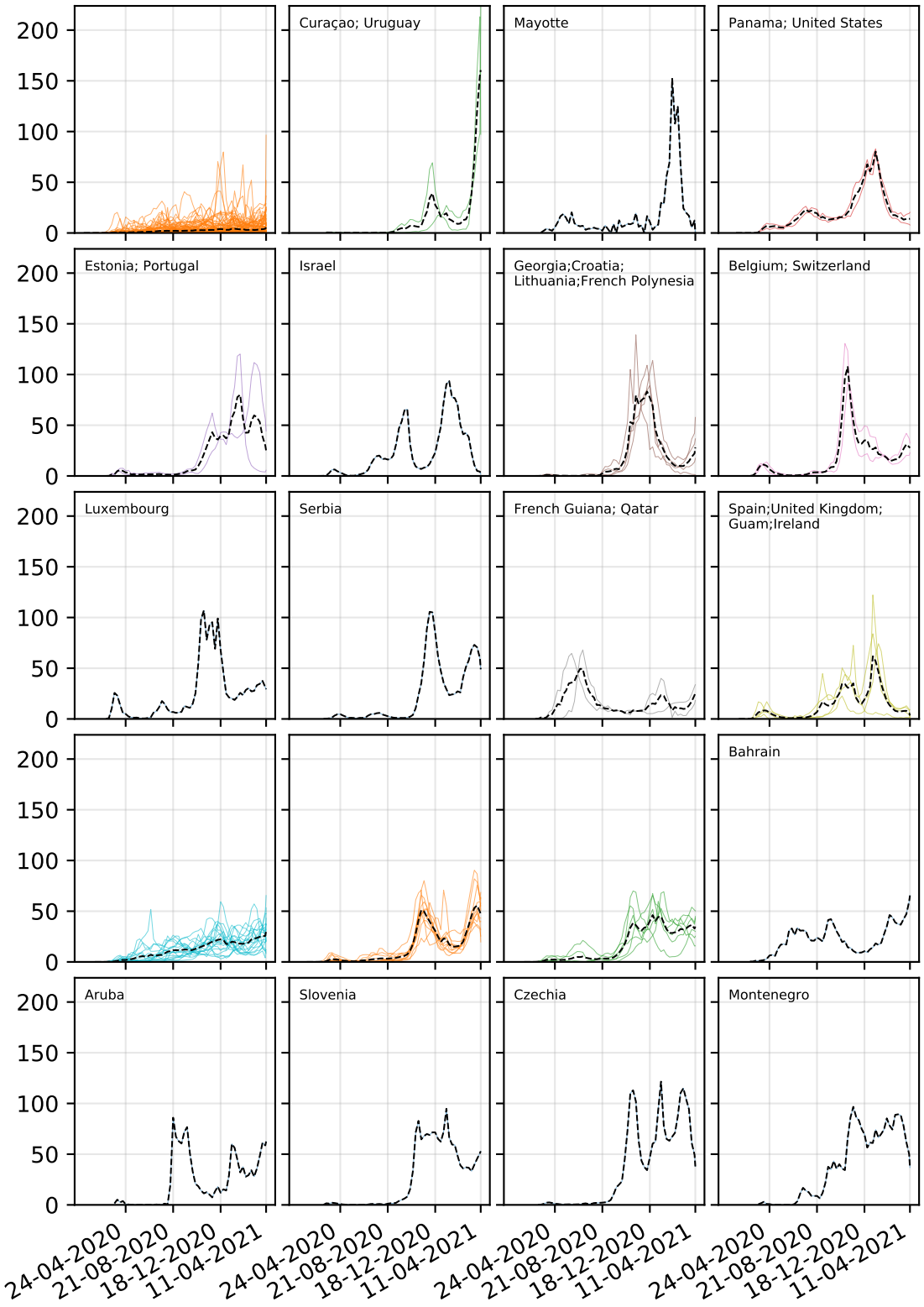


Figure 4.6 The *ERP* cluster plots

Table 4.2 Clusters for the *ERP*

1	Afghanistan; Angola; Australia; Azerbaijan; Barbados; Bangladesh; Burkina Faso; Burundi; Benin; Brunei Darussalam; Bolivia, Plurinational State of; Bahamas; Bhutan; Botswana; Belarus; Belize; Canada; Congo, The Democratic Republic of the; Central African Republic; Congo; Côte d'Ivoire; Cameroon; China; Cuba; Cabo Verde; Djibouti; Dominican Republic; Algeria; Ecuador; Egypt; Eritrea; Ethiopia; Finland; Fiji; Gabon; Grenada; Ghana; Gambia; Guinea; Guadeloupe; Equatorial Guinea; Greece; Guatemala; Guinea-Bissau; Guyana; Honduras; Haiti; Indonesia; India; Iraq; Iran, Islamic Republic of; Iceland; Jersey; Jamaica; Japan; Kenya; Kyrgyzstan; Cambodia; Comoros; Korea, Republic of; Kazakhstan; Lao People's Democratic Republic; Saint Lucia; Sri Lanka; Liberia; Lesotho; Libya; Morocco; Madagascar; Mali; Myanmar; Mongolia; Martinique; Mauritania; Mauritius; Malawi; Mexico; Malaysia; Mozambique; New Caledonia; Niger; Nigeria; Nicaragua; Norway; Nepal; New Zealand; Oman; Papua New Guinea; Philippines; Pakistan; Paraguay; Réunion; Russian Federation; Rwanda; Saudi Arabia; Solomon Islands; Sudan; Singapore; Sierra Leone; Senegal; Somalia; Suriname; South Sudan; Sao Tome and Principe; El Salvador; Syrian Arab Republic; Eswatini; Chad; Togo; Thailand; Tajikistan; Timor-Leste; Tunisia; Trinidad and Tobago; Tanzania, United Republic of; Uganda; Uzbekistan; Saint Vincent and the Grenadines; Venezuela, Bolivarian Republic of; Virgin Islands, U.S.; Viet Nam; Vanuatu; Samoa; Yemen; South Africa; Zambia; Zimbabwe
13	United Arab Emirates; Albania; Argentina; Brazil; Chile; Colombia; Costa Rica; Germany; Denmark; Kuwait; Moldova, Republic of; Malta; Maldives; Peru; Palestine, State of; Romania; Turkey; Ukraine
14	Austria; Bosnia and Herzegovina; Bulgaria; Cyprus; France; Hungary; Italy; Jordan; North Macedonia; Poland; Kosovo
15	Armenia; Lebanon; Latvia; Netherlands; Sweden; Slovakia
7	Georgia; Croatia; Lithuania; French Polynesia
12	Spain; United Kingdom; Guam; Ireland
2	Curaçao; Uruguay
4	Panama; United States
5	Estonia; Portugal
8	Belgium; Switzerland
11	French Guiana; Qatar
3	Mayotte
6	Israel
9	Luxembourg
10	Serbia
16	Bahrain
17	Aruba
18	Slovenia
19	Czechia
20	Montenegro

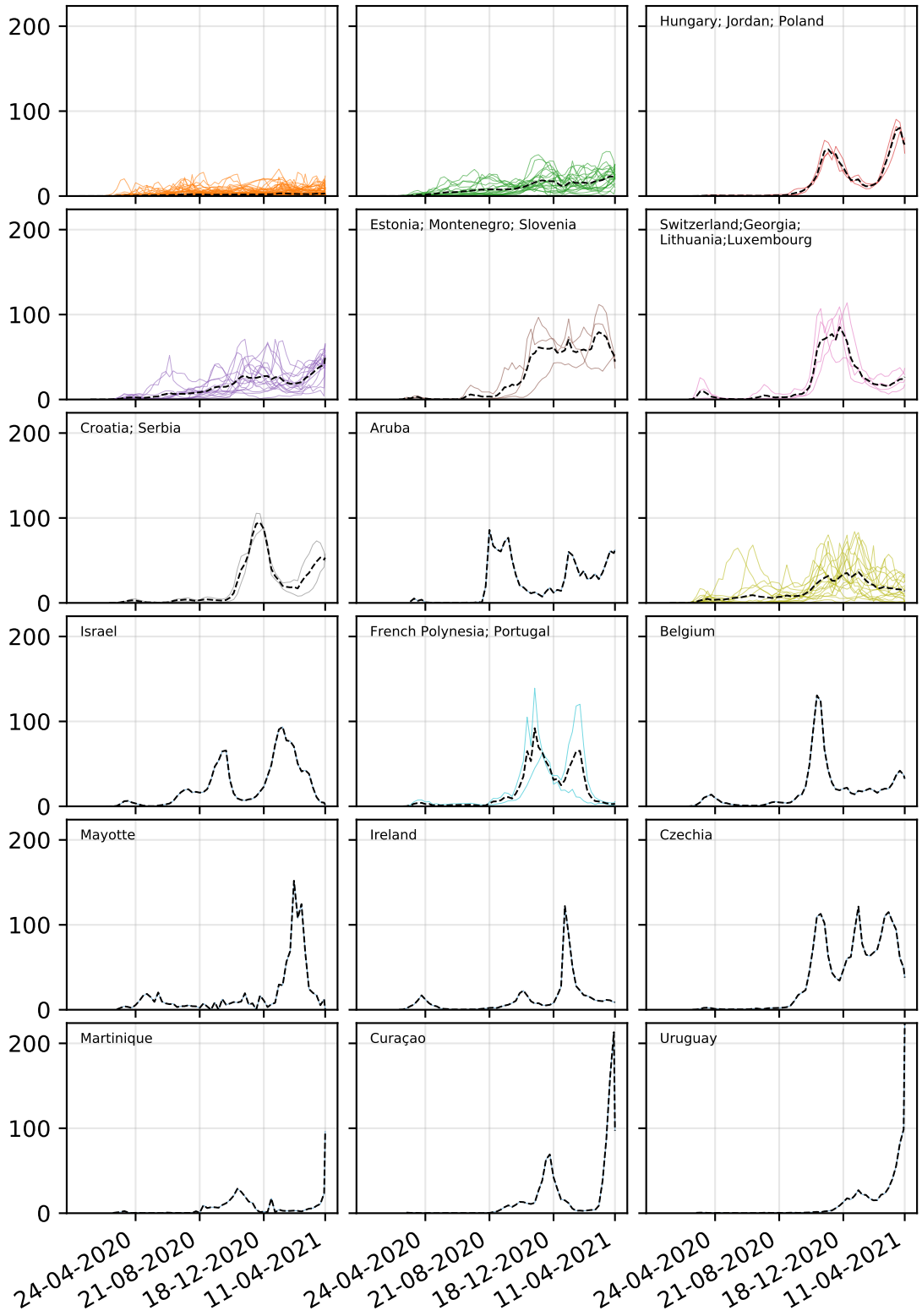


Figure 4.7 The *DFD* cluster plots

Table 4.3 Clusters for the *DFD*

1	Afghanistan; Angola; Australia; Barbados; Bangladesh; Burkina Faso; Burundi; Benin; Brunei Darussalam; Bolivia, Plurinational State of; Bahamas; Bhutan; Botswana; Belarus; Congo, The Democratic Republic of the; Central African Republic; Congo; Côte d'Ivoire; Cameroon; China; Cuba; Djibouti; Dominican Republic; Algeria; Ecuador; Egypt; Eritrea; Ethiopia; Finland; Fiji; Gabon; Grenada; Ghana; Gambia; Guinea; Equatorial Guinea; Guatemala; Guinea-Bissau; Guyana; Honduras; Haiti; Indonesia; India; Iraq; Iceland; Jamaica; Japan; Kenya; Kyrgyzstan; Cambodia; Comoros; Korea, Republic of; Kazakhstan; Lao People's Democratic Republic; Sri Lanka; Liberia; Lesotho; Libya; Morocco; Madagascar; Mali; Myanmar; Mongolia; Mauritania; Mauritius; Malawi; Mexico; Malaysia; Mozambique; New Caledonia; Niger; Nigeria; Nicaragua; Norway; Nepal; New Zealand; Papua New Guinea; Philippines; Pakistan; Réunion; Russian Federation; Rwanda; Saudi Arabia; Solomon Islands; Sudan; Singapore; Sierra Leone; Senegal; Somalia; Suriname; South Sudan; Sao Tome and Principe; El Salvador; Syrian Arab Republic; Eswatini; Chad; Togo; Thailand; Tajikistan; Timor-Leste; Trinidad and Tobago; Tanzania, United Republic of; Uganda; Uzbekistan; Venezuela, Bolivarian Republic of; Virgin Islands, U.S.; Viet Nam; Vanuatu; Samoa; Yemen; South Africa; Zambia; Zimbabwe
2	United Arab Emirates; Albania; Azerbaijan; Bulgaria; Brazil; Canada; Colombia; Cabo Verde; Germany; Greece; Iran, Islamic Republic of; Kuwait; Moldova, Republic of; Maldives; Oman; Peru; Paraguay; Romania; Tunisia; Ukraine; Saint Vincent and the Grenadines
9	Armenia; Austria; Belize; Denmark; Spain; United Kingdom; French Guiana; Guam; Italy; Jersey; Saint Lucia; Latvia; Malta; Panama; Qatar; Slovakia; United States
4	Argentina; Bosnia and Herzegovina; Bahrain; Chile; Costa Rica; Cyprus; France; Guadeloupe; Lebanon; North Macedonia; Netherlands; Palestine, State of; Sweden; Turkey; Kosovo
6	Switzerland; Georgia; Lithuania; Luxembourg
3	Hungary; Jordan; Poland
5	Estonia; Montenegro; Slovenia
7	Croatia; Serbia
11	French Polynesia; Portugal
8	Aruba
10	Israel
12	Belgium
13	Mayotte
14	Ireland
15	Czechia
16	Martinique
17	Curaçao
18	Uruguay

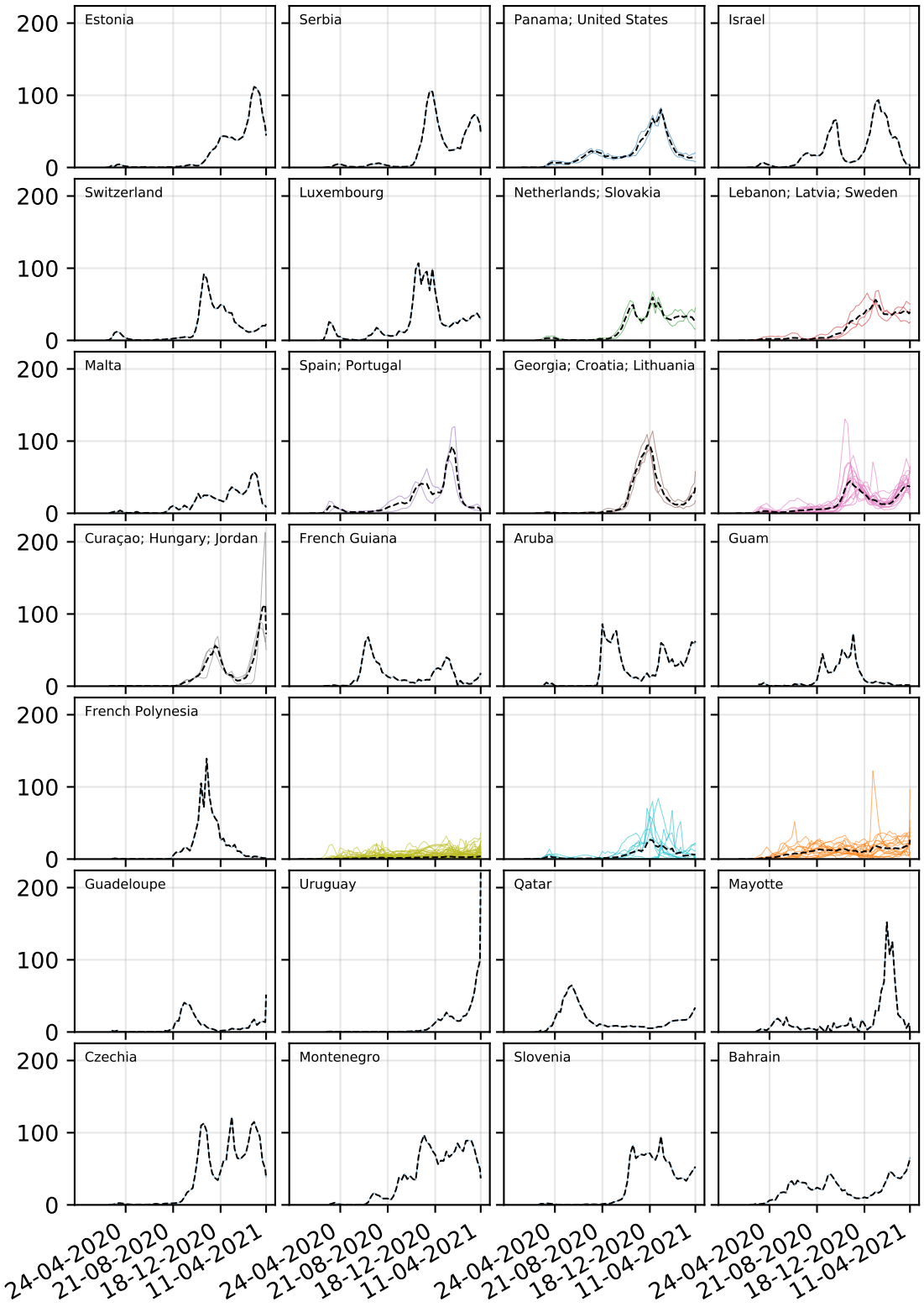


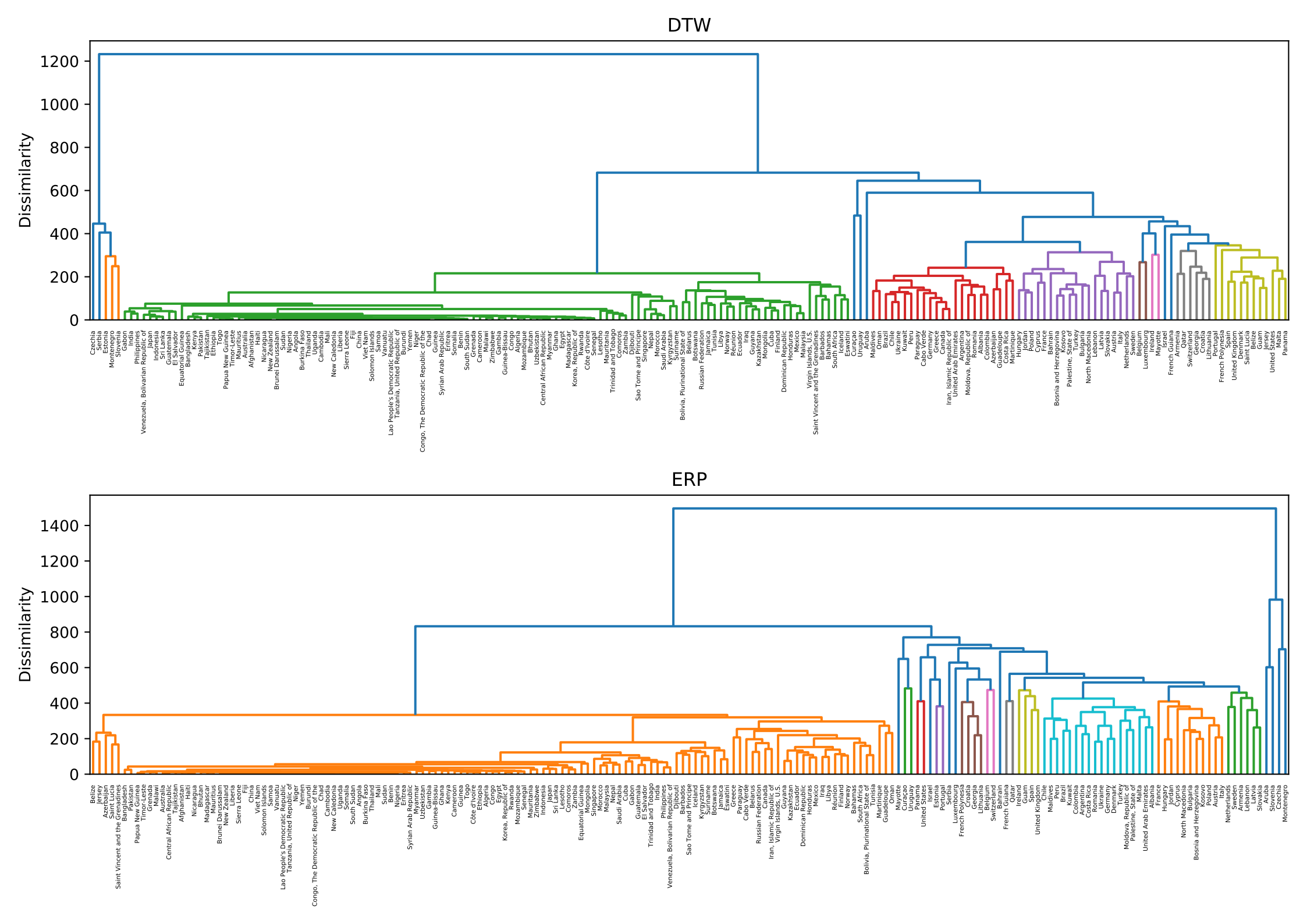
Figure 4.8 The *LCSD* cluster plots

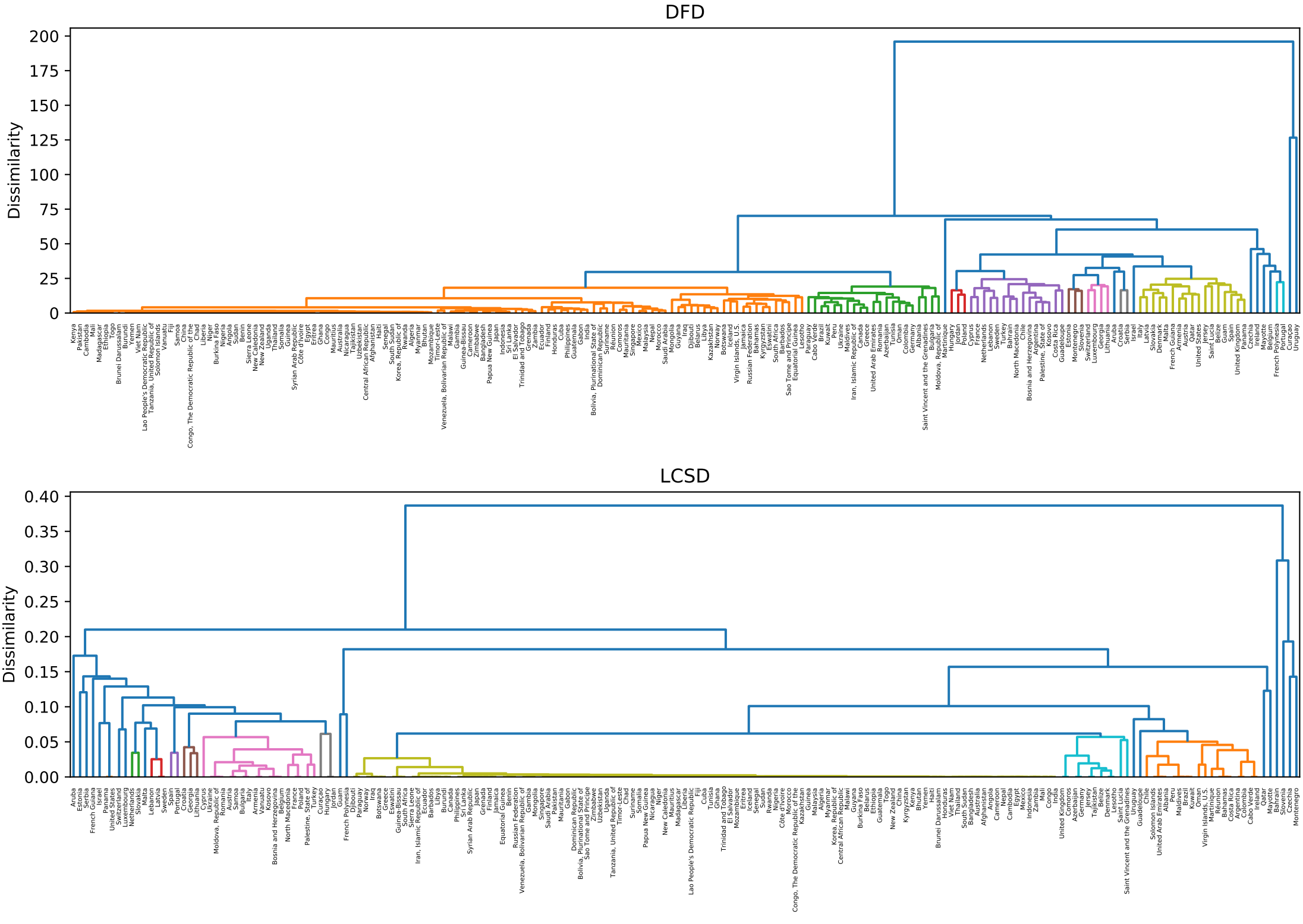
Table 4.4 Clusters for the *LCSD*

18	Afghanistan; Angola; Australia; Barbados; Bangladesh; Burkina Faso; Burundi; Benin; Brunei Darussalam; Bolivia, Plurinational State of; Bhutan; Botswana; Belarus; Canada; Congo, The Democratic Republic of the; Central African Republic; Congo; Côte d'Ivoire; Cameroon; China; Cuba; Djibouti; Dominican Republic; Algeria; Ecuador; Egypt; Eritrea; Ethiopia; Finland; Fiji; Gabon; Grenada; Ghana; Gambia; Guinea; Equatorial Guinea; Greece; Guatemala; Guinea-Bissau; Guyana; Honduras; Haiti; Indonesia; India; Iraq; Iran, Islamic Republic of; Iceland; Jamaica; Japan; Kenya; Kyrgyzstan; Cambodia; Korea, Republic of; Kazakhstan; Lao People's Democratic Republic; Sri Lanka; Liberia; Libya; Morocco; Madagascar; Mali; Myanmar; Mongolia; Mauritania; Mauritius; Malawi; Mexico; Malaysia; Mozambique; New Caledonia; Niger; Nigeria; Nicaragua; Norway; Nepal; New Zealand; Papua New Guinea; Philippines; Pakistan; Paraguay; Russian Federation; Rwanda; Saudi Arabia; Sudan; Singapore; Sierra Leone; Senegal; Somalia; Suriname; South Sudan; Sao Tome and Principe; El Salvador; Syrian Arab Republic; Eswatini; Chad; Togo; Thailand; Timor-Leste; Tunisia; Trinidad and Tobago; Tanzania, United Republic of; Uganda; Uzbekistan; Venezuela, Bolivarian Republic of; Viet Nam; Yemen; South Africa; Zambia; Zimbabwe
12	Armenia; Austria; Bosnia and Herzegovina; Belgium; Bulgaria; Cyprus; France; Italy; Moldova, Republic of; North Macedonia; Poland; Palestine, State of; Romania; Turkey; Ukraine; Vanuatu; Samoa; Kosovo
20	United Arab Emirates; Albania; Argentina; Brazil; Bahamas; Chile; Colombia; Costa Rica; Cabo Verde; Ireland; Kuwait; Martinique; Maldives; Oman; Peru; Réunion; Solomon Islands; Virgin Islands, U.S.
19	Azerbaijan; Belize; Germany; Denmark; United Kingdom; Jersey; Comoros; Saint Lucia; Lesotho; Tajikistan; Saint Vincent and the Grenadines
8	Lebanon; Latvia; Sweden
11	Georgia; Croatia; Lithuania
13	Curaçao; Hungary; Jordan
3	Panama; United States
7	Netherlands; Slovakia
10	Spain; Portugal
1	Estonia
2	Serbia
4	Israel
5	Switzerland
6	Luxembourg
9	Malta
14	French Guiana
15	Aruba
16	Guam
17	French Polynesia
21	Guadeloupe
22	Uruguay
23	Qatar
24	Mayotte
25	Czechia
26	Montenegro
27	Slovenia
28	Bahrain

Příloha A

Appendix


DTW
Dissimilarity
0
200
400
600
800
1000
1200
ERP
Dissimilarity
0
200
400
600
800
1000
1200
1400


DFD
Dissimilarity
200
175
150
125
100
75
50
25
0
Kenya
Pakistan
Cambodia
Mali
Malawi
Madagascar
Ethiopia
Togo
Brunei Darussalam
Burundi
Yemen
Viet Nam
Lao People's Democratic Republic
Tanzania, United Republic of
Solomon Islands
Vanuatu
Fiji
Samoa
China
Congo, The Democratic Republic of the
Chad
Liberia
Niger
Burkina Faso
Nigeria
Angola
Sudan
Benin
Sierra Leone
New Caledonia
New Zealand
Uganda
Thailand
Somalia
Guinea
Syrian Arab Republic
Côte d'Ivoire
Eritrea
Ghana
Congo
Mauritius
Australia
Nicaragua
Tajikistan
Uzbekistan
Central African Republic
Afghanistan
Haiti
Senegal
South Sudan
Korea, Republic of
Rwanda
Algeria
Myanmar
Ecuador
Mozambique
Timor-Leste
Venezuela, Bolivarian Republic of
Malawi
Guatemala
Guinea-Bissau
Cameroon
Zimbabwe
Bangladesh
Papua New Guinea
Japan
Indonesia
Sri Lanka
El Salvador
Trinidad and Tobago
Grenada
Zambia
Ecuador
Finland
Honduras
Cuba
Philippines
Guatemala
Gabon
India
Bolivia, Plurinational State of
Dominican Republic
Suriname
Réunion
Comoros
Mauritania
Singapore
Mexico
Malaysia
Nepal
Morocco
Saudi Arabia
Mongolia
Guyana
Iraq
Djibouti
Belarus
Libya
Kazakhstan
Norway
Botswana
Iceland
Virgin Islands, U.S.
Jamaica
Russian Federation
Bahamas
Kyrgyzstan
Eswatini
South Africa
Sao Tome and Principe
Equatorial Guinea
Lesotho
Paraguay
Cabo Verde
Brazil
Kuwait
Peru
Ukraine
Iran, Islamic Republic of
Maldives
Canada
Greece
United Arab Emirates
Romania
Azerbaijan
Tunisia
Oman
Colombia
Germany
Albania
Saint Vincent and the Grenadines
Belgium
Moldova, Republic of
Martinique
Hungary
Jordan
Poland
Cyprus
France
Netherlands
Lebanon
Sweden
Turkey
Bahrain
North Macedonia
Chile
Bosnia and Herzegovina
Argentina
Palestine, State of
Kosovo
Costa Rica
Guadeloupe
Estonia
Montenegro
Slovenia
Switzerland
Luxembourg
Georgia
Lithuania
Malaysia
Croatia
Serbia
Israel
Italy
Latvia
Slovakia
Denmark
Malta
French Guiana
Armenia
Austria
Qatar
United States
Jersey
Saint Lucia
Belize
Guam
Ireland
United Kingdom
Panama
Czechia
Iceland
Mayotte
Belgium
French Polynesia
Portugal
Curaçao
Uruguay
LCSD
Dissimilarity
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Aruba
Estonia
Serbia
French Guiana
Israel
Panama
United States
Switzerland
Luxembourg
Netherlands
Slovakia
Malta
Lebanon
Latvia
Sweden
Spain
Portugal
Croatia
Georgia
Lithuania
Cyprus
Ukraine
Moldova, Republic of
Romania
Austria
Samoa
Bulgaria
Italy
Armenia
Vanuatu
Kosovo
Bosnia and Herzegovina
Belgium
North Macedonia
France
Poland
Palestine, State of
Turkey
Curaçao
Hungary
Jordan
Fiji
French Polynesia
Djibouti
Paraguay
Norway
Iraq
Botswana
Greece
Eswatini
Guinea-Bissau
South Africa
Sierra Leone
Iran, Islamic Republic of
Ecuador
Barbados
Libya
Burundi
Canada
Philippines
Sri Lanka
Syrian Arab Republic
Japan
Grenada
Finland
Jamaica
Equatorial Guinea
Benin
Russian Federation
Venezuela, Bolivarian Republic of
Gambia
Mongolia
Singapore
Saudi Arabia
Pakistan
Mauritania
Gabon
Dominican Republic
Bolivia, Plurinational State of
Sao Tome and Principe
Zimbabwe
Uzbekistan
Morocco
Tanzania, United Republic of
Timor-Leste
Chad
Suriname
Somalia
Papua New Guinea
Nicaragua
Niger
New Caledonia
Mauritius
Madagascar
Liberia
Lao People's Democratic Republic
Fiji
Cuba
Tunisia
Ghana
Malawi
Trinidad and Tobago
El Salvador
Mozambique
Eritrea
Iceland
Seychelles
Sudan
Rwanda
Nigeria
Côte d'Ivoire
Morocco
Congo, The Democratic Republic of the
Kazakhstan
Guinea
Malaysia
Algeria
Myanmar
Korea, Republic of
Central African Republic
Malawi
Guyana
Burkina Faso
Belarus
Ethiopia
Guatemala
Togo
New Zealand
Kenya
Kyrgyzstan
Bhutan
Yemen
Haiti
Brunei Darussalam
Honduras
Viet Nam
Thailand
South Sudan
Bangladesh
Afghanistan
Australia
Angola
Cameroon
Nepal
Cambodia
Egypt
Mexico
Indonesia
Zambia
Mali
Congo
India
United Kingdom
Comoros
Azerbaijan
Germany
Jersey
Tajikistan
Belize
Denmark
Lesotho
Saint Lucia
Saint Vincent and the Grenadines
Uruguay
Guadeloupe
Chile
Solomon Islands
United Arab Emirates
Albania
Peru
Maldives
Brazil
Kuwait
Oman
Virgin Islands, U.S.
Réunion
Martinique
Bahamas
Costa Rica
Argentina
Colombia
Cabo Verde
Iceland
Qatar
Mayotte
Bahrain
Slovenia
Czechia
Montenegro

Bibliography

1. ZAWBAA, Hossam; EL-GENDY, Ahmed; SAEED, Haitham; OSAMA, Hasnaa; ALI, Ahmed M. A.; GOMAA, Dina; ABDELRAHMAN, Mona; HARB, Hadeer S.; MADNEY, Yasmin M.; ABDELRAHIM, Mohamed E. A. A study of the possible factors affecting COVID-19 spread, severity and mortality and the effect of social distancing on these factors: Machine learning forecasting model. *International Journal of Clinical Practice* [online]. 2021, e14116. Available from DOI: https://doi.org/10.1111/ijcp.14116.
2. TULSHYAN, Vatsal; SHARMA, Dolly; MITTAL, Mamta. An Eye on the Future of COVID-19: Prediction of Likely Positive Cases and Fatality in India over a 30-Day Horizon Using the Prophet Model. *Disaster Medicine and Public Health Preparedness* [online]. 2020, pp. 1–7. Available from DOI: 10.1017/dmp.2020.444.
3. FARCOMENI, Alessio; MARUOTTI, Antonello; DIVINO, Fabio; JONA-LASINIO, Giovanna; LOVISON, Gianfranco. An ensemble approach to short-term forecast of COVID-19 intensive care occupancy in Italian regions. *Biometrical Journal* [online]. 2021, vol. 63, no. 3, pp. 503–513. Available from DOI: https://doi.org/10.1002/bimj.202000189.
4. ZEROUAL, Abdelhafid; HARROU, Fouzi; DAIRI, Abdelkader; SUN, Ying. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons & Fractals* [online]. 2020, vol. 140, p. 110121. ISSN 0960-0779. Available from DOI: https://doi.org/10.1016/j.chaos.2020.110121.
5. CHIMMULA, Vinay Kumar Reddy; ZHANG, Lei. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons & Fractals* [online]. 2020, vol. 135, p. 109864. ISSN 0960-0779. Available from DOI: https://doi.org/10.1016/j.chaos.2020.109864.
6. WARD, Michael P.; XIAO, Shuang; ZHANG, Zhijie. The role of climate during the COVID-19 epidemic in New South Wales, Australia. *Transboundary and Emerging Diseases* [online]. 2020, vol. 67, no. 6, pp. 2313–2317. Available from DOI: https://doi.org/10.1111/tbed.13631.

7. SASIKUMAR, Keerthi; NATH, Debashis; NATH, Reshmita; CHEN, Wen. Impact of Extreme Hot Climate on COVID-19 Outbreak in India. *GeoHealth* [online]. 2020, vol. 4, no. 12, e2020GH000305. Available from DOI: https://doi.org/10.1029/2020GH000305. e2020GH000305 2020GH000305.

8. LI, He; XU, Xiao-Long; DAI, Da-Wei; HUANG, Zhen-Yu; MA, Zhuang; GUAN, Yan-Jun. Air pollution and temperature are associated with increased COVID-19 incidence: A time series study. *International Journal of Infectious Diseases* [online]. 2020, vol. 97, pp. 278–282. ISSN 1201-9712. Available from DOI: https://doi.org/10.1016/j.ijid.2020.05.076.

9. MERZON, Eugene; TWOROWSKI, Dmitry; GOROHOVSKI, Alessandro; VINKER, Shlomo; GOLAN COHEN, Avivit; GREEN, Ilan; FRENKEL-MORGENSTERN, Milana. Low plasma 25(OH) vitamin D level is associated with increased risk of COVID-19 infection: an Israeli population-based study. *The FEBS Journal* [online]. 2020, vol. 287, no. 17, pp. 3693–3702. Available from DOI: https://doi.org/10.1111/febs.15495.

10. ROJAS, Fernando; VALENZUELA, Olga; ROJAS, Ignacio. Estimation of COVID-19 dynamics in the different states of the United States using Time-Series Clustering. *medRxiv* [online]. 2020. Available from DOI: 10.1101/2020.06.29.20142364.

11. DONG, Ensheng; DU, Hongru; GARDNER, Lauren. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases* [online]. 2020, vol. 20, no. 5, pp. 533–534. Available from DOI: 10.1016/s1473-3099(20)30120-1.

12. MAHMOUDI, Mohammad Reza; BALEANU, Dumitru; MANSOR, Zulkefli; TUAN, Bui Anh; PHO, Kim-Hung. Fuzzy clustering method to compare the spread rate of Covid-19 in the high risks countries. *Chaos, Solitons & Fractals* [online]. 2020, vol. 140, p. 110230. ISSN 0960-0779. Available from DOI: https://doi.org/10.1016/j.chaos.2020.110230.

13. ALVAREZ, Emiliano; GABRIEL BRIDA, Juan; LIMAS, Erick. Comparisons of COVID-19 dynamics in the different countries of the World using Time-Series clustering. *medRxiv* [online]. 2020. Available from DOI: 10.1101/2020.08.18.20177261.

14. MAX ROSER Hannah Ritchie, Esteban Ortiz-Ospina; HASELL, Joe. Coronavirus Pandemic (COVID-19). *Our World in Data* [online]. 2020. Available also from: https://ourworldindata.org/coronavirus.

15. CHEN, Jianmin; YAN, Jun; ZHANG, Panpan. *Clustering US States by Time Series of COVID-19 New Case Counts with Non-negative Matrix Factorization* [online]. 2021. No. 3. Available from arXiv: 2011.14412v3 [stat.AP].

16. *WHO COVID-19 Dashboard* [online].
Geneva: World Health Organization, 2020 [visited on 2021-03-23].
Available from: https://covid19.who.int/info.

17. DEDECIUS, Kamil.
Téma 1: Úvod do problematiky, Markovské procesy [presentation]. In:
FIT ČVUT Courses [online].
Praha: ČVUT FIT v Praze, 2021 [visited on 2021-04-05].
Available from: https://courses.fit.cvut.cz/MI-SCR/lectures.

18. HYNDMAN, Rob. *Forecasting : principles and practice* [online]. 2nd ed.
Melbourne: OTexts, 2018. ISBN 9780987507112.
Available also from: https://otexts.com/fpp2/decomposition.html.
[Cited: 2021-3-25].

19. KLOUDA, Karel; PABLO MALDONADO LOPEZ, Juan; VAŠATA, Daniel.
Hierarchické shlukování a algoritmus k-means [presentation]. In:
FIT ČVUT Courses [online].
Praha: ČVUT FIT v Praze, 2021 [visited on 2021-04-01].
Available from: https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/files/BI-VZD-04-cs-lecture.pdf.

20. WILSON, Wallace Alvin. On Semi-Metric Spaces.
American Journal of Mathematics [online]. 1931, vol. 53, no. 2, pp. 361–373.
ISSN 00029327, ISSN 10806377.
Available also from: http://www.jstor.org/stable/2370790.

21. MEINARD, Müller. Dynamic Time Warping. In:
Information Retrieval for Music and Motion.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84.
ISBN 978-3-540-74048-3. Available from DOI: 10.1007/978-3-540-74048-3_4.

22. KEOGH, Eamonn; RATANAMAHATANA, Chotirat.
Exact indexing of dynamic time warping.
Knowledge and Information Systems [online]. 2005, vol. 7, pp. 358–386.
Available from DOI: 10.1007/s10115-004-0154-9.

23. CHEN, Lei; ÖZSU, M. Tamer; ORIA, Vincent.
Robust and Fast Similarity Search for Moving Object Trajectories. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*.
Baltimore, Maryland: Association for Computing Machinery, 2005, pp. 491–502.
SIGMOD '05. ISBN 1595930604. Available from DOI: 10.1145/1066157.1066213.

24. BERGROTH, L.; HAKONEN, H.; RAITA, T.
A survey of longest common subsequence algorithms. In:
Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000 [online]. 2000, pp. 39–48.
Available from DOI: 10.1109/SPIRE.2000.878178.

25. STEIN, Cliff. *Longest common subsequence* [online] [visited on 2021-03-28].
Available from:
http://www.columbia.edu/~cs2035/courses/csor4231.F11/lcs.pdf.

26. CHEN, Lei; NG, Raymond. On the marriage of lp-norms and edit distance. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* [online]. 2004, pp. 792–803.
Available also from: https://dl.acm.org/doi/10.5555/1316689.1316758.

27. MALÍK, Josef; SUCHÝ, Ondřej; TVRDÍK, Pavel; VALLA, Tomáš. Dynamické programování [presentation]. In: *FIT ČVUT Courses* [online]. Praha: ČVUT FIT v Praze, 2021 [visited on 2021-04-01].
Available from: https://courses.fit.cvut.cz/BI-AG1/media/lectures.

28. ARONOV, Boris; HAR-PELED, Sariel; KNAUER, Christian; WANG, Yusu; WENK, Carola. Fréchet Distance for Curves, Revisited.
In: AZAR, Yossi; ERLEBACH, Thomas (eds.). *Algorithms – ESA 2006* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 52–63.
ISBN 978-3-540-38876-0.

29. HAR-PELED, Sariel. *Geometric approximation algorithms* [online]. Providence, R.I: American Mathematical Society, 2011 [visited on 2021-04-05]. Mathematical surveys and monographs, no. v. 173. ISBN 9780821849118.
Available from: https://sarielhp.org/book/chapters/frechet.pdf.
Additional chapters, Fréchet distance.

30. GELER, Z.; KURBALIJA, V.; IVANOVIĆ, M.; RADOVANOVIĆ, M.; DAI, W. Dynamic Time Warping: Itakura vs Sakoe-Chiba. In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)* [online]. 2019, pp. 1–6.
Available from DOI: 10.1109/INISTA.2019.8778300.

31. MANNING, Christopher; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. *Single-link and complete-link clustering* [online]. 2009 [visited on 2021-04-14].
Available from:
https://nlp.stanford.edu/IR-book/html/htmledition/single-link-and-complete-link-clustering-1.html. publisher: Cambridge University Press.

32. KEOGH, Eamonn; CHAKRABARTI, Kaushik; PAZZANI, Michael; MEHROTRA, Sharad.
Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* [online]. 2001, vol. 3, no. 3, pp. 263–286.

Content of the enclosed media

- `readme.txt` ..... brief description of the media content
- `thesis.pdf` ..... thesis text in PDF format
- `data` ..... data used for analysis
- `src`
  - `impl`
    - `practical.ipynb` ..... jupyter notebook source code for analysis
    - `image_aux.ipynb` ..... jupyter notebook source code for other images
  - `thesis` ..... thesis source code in LaTeX format and bibliography file
    - `images` ..... images used in the thesis