TIME SERIES CLUSTERING

BY DYNAMICS TIME WARPING (DTW)

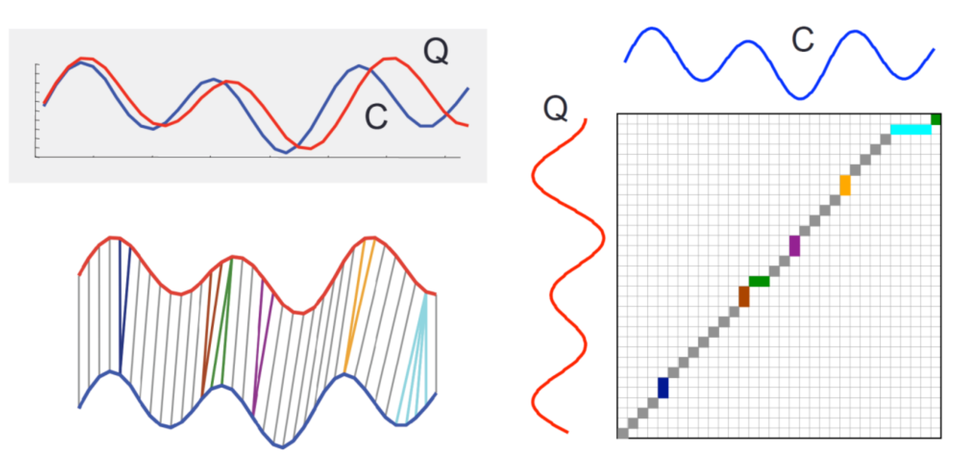
LIEN PHAM - APRIL 2022

Table of content

Table of content

What is dynamic time warping(DTW)

- DTW finds the minimum cost path between the complete matrix of pairwise distances between two time-series.
- This matrix of pairwise distances is referred to as the cost matrix.
- Low cost implies similarity, high cost implies dissimilarity. DTW finds a path through the cost matrix of minimum total cost. Each valid path through the cost matrix is called a “warping” path.



(a) Time series C and Q and the linked points in time by DTW.

(b) Minimum warping path W through the LCM of time series C and Q.

Why DTW?

Dynamic Time warping is a method of calculating distance that is more accurate than Euclidean distance

- It has an advantage over Euclidean if datapoints are shifted between each other and we want to look rather at its shape
- Additionally two time series don't have to be equal in length what is an assumption required by the Euclidean distance
- The Euclidean distance takes pairs of datapoints and compares them to each other.
- DTW calculates the smallest distance between all points - this enables a one-to-many match
- In literature dynamic time warping is often paired with k-medoids and hierarchical methods.

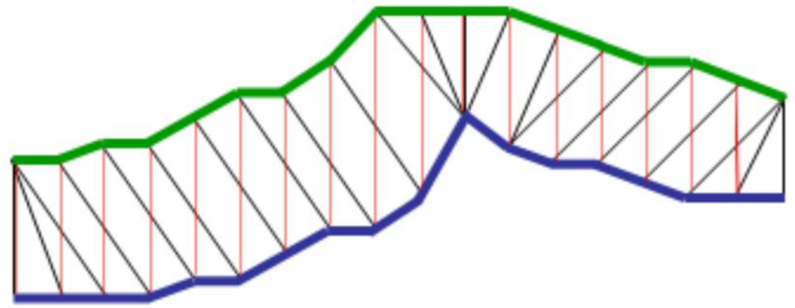
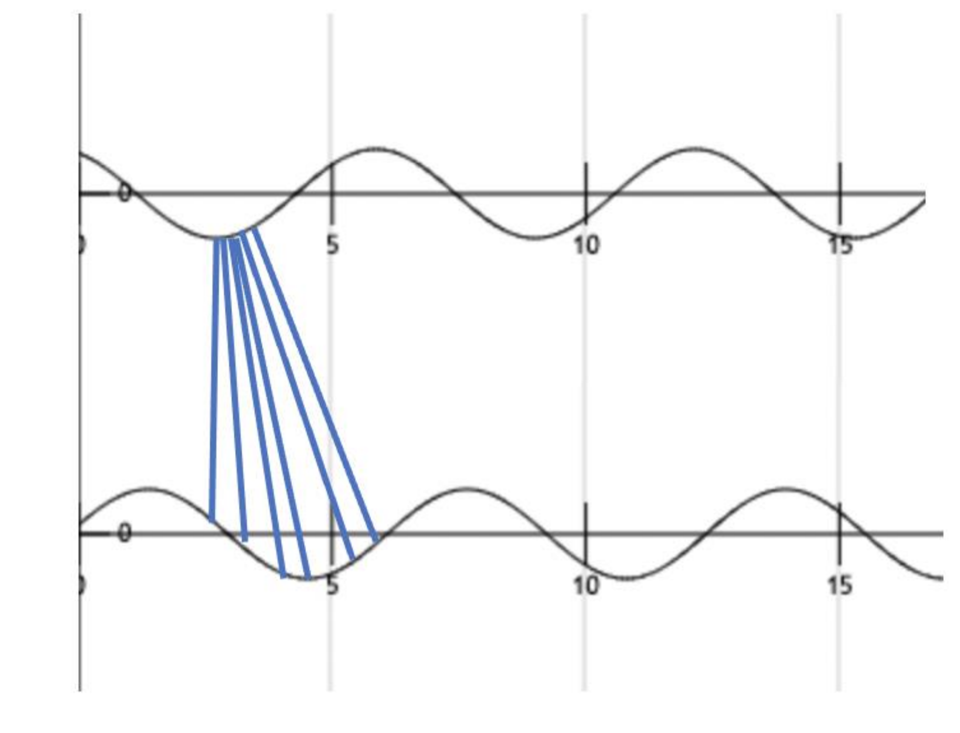


Fig. 1: Visual comparison of matched points based on DTW (black) and Euclidean (red) distance

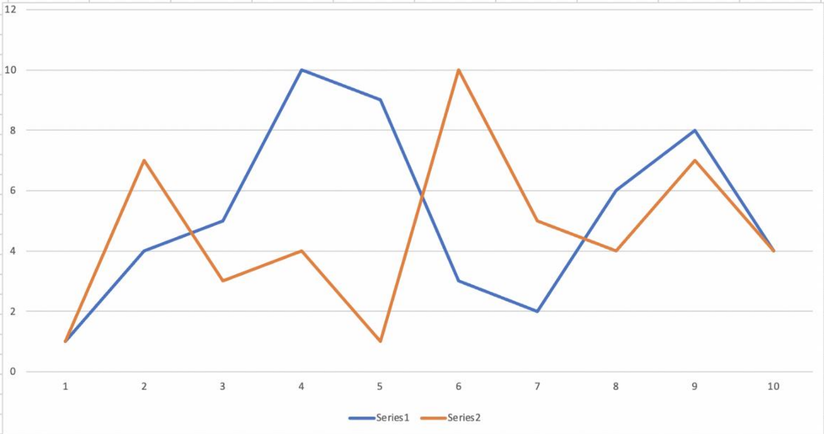
- DTW compares amplitude of first signal at time T with amplitude of second signal at time T+1 and T-1 or T+2 and T-2.
- This makes sure it does not give low similarity score for signals with similar shape and different phase



How it works? (1/8)

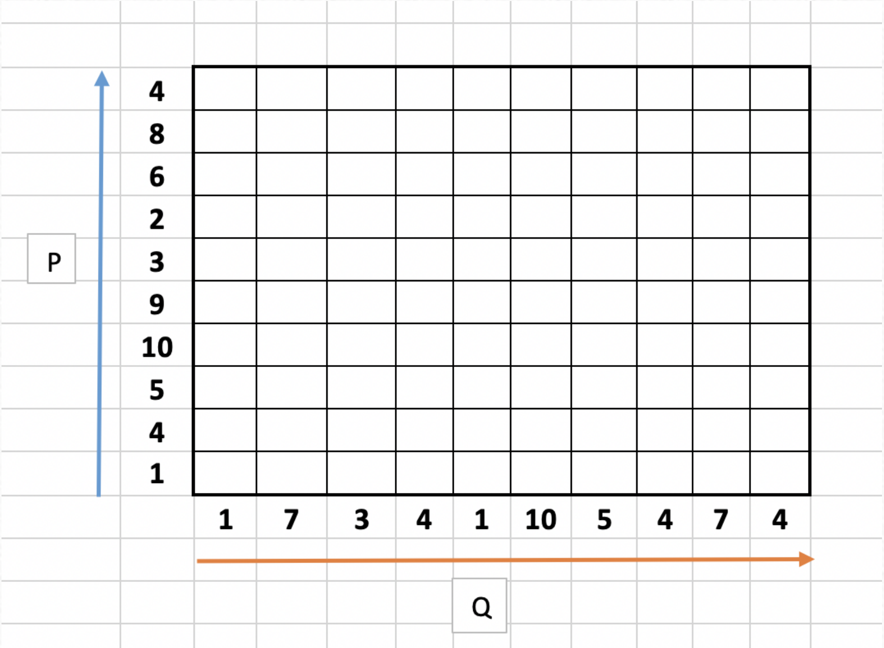
- Let us take two time series signals P and Q
- Series 1 (P) : 1,4,5,10,9,3,2,6,8,4
- Series 2 (Q): 1,7,3,4,1,10,5,4,7,4

Time	1	2	3	4	5	6	7	8	9	10
Series 1	1	4	5	10	9	3	2	6	8	4
Series 2	1	7	3	4	1	10	5	4	7	4



How it works? (2/8)

- ***Step 1 :*** **Empty Cost Matrix Creation**
- Create an empty cost matrix M with x and y labels as amplitudes of the two series to be compared.



How it works? (3/8)

- ***Step 2: Cost Calculation***
- Fill the cost matrix using the formula mentioned below starting from left and bottom corner.
- **M(i, j) = |P(i) — Q(j)| + min (M(i-1,j-1), M(i, j-1), M(i-1,j))**
- where
- M is the matrix
- i is the iterator for series P
- j is the iterator for series Q

P										
4	42	24	-	-	-	-	-	-	-	-
8	39	21	-	-	-	-	-	-	-	-
6	32	20	-	-	-	-	-	-	-	-
2	27	19	-	-	-	-	-	-	-	-
3	26	14	-	-	-	-	-	-	-	-
9	24	10	-	-	-	-	-	-	-	-
10	16	8	12	11	-	-	-	-	-	-
5	7	5	5	5	-	-	-	-	-	-
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

How it works? (4/8)

- Let us take few examples (11,3 and 8) to illustrate the calculation as highlighted in the below table.

P

4	42	24	-	-	-	-	-	-	-	-
8	39	21	-	-	-	-	-	-	-	-
6	32	20	-	-	-	-	-	-	-	-
2	27	19	-	-	-	-	-	-	-	-
3	26	14	-	-	-	-	-	-	-	-
9	24	10	-	-	-	-	-	-	-	-
10	16	8	12	**11**	-	-	-	-	-	-
5	7	5	5	5	-	-	-	-	-	-
4	**3**	3	4	4	7	13	14	14	17	17
1	0	6	**8**	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

for 11,
|10 –4| + min(5, 12, 5)
= 6 + 5
= 11
Similarly for 3,
|4 –1| + min(0)
= 3+ 0
= 3
and for 8,
|1 –3| + min(6)
= 2 + 6
= 8

P

4	42	24	-	-	-	-	-	-	-	-
8	39	21	-	-	-	-	-	-	-	-
6	32	20	-	-	-	-	-	-	-	-
2	27	19	-	-	-	-	-	-	-	-
3	26	14	-	-	-	-	-	-	-	-
9	24	10	-	-	-	-	-	-	-	-
10	16	8	12	**11**	-	-	-	-	-	-
5	7	5	5	5	-	-	-	-	-	-
4	**3**	3	4	4	7	13	14	14	17	17
1	0	6	**8**	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

How it works? (6/8)

- ***Step 3:* Warping Path Identification**
- Identify the warping path starting from top right corner of the matrix and traversing to bottom left. The traversal path is identified based on the neighbour with minimum value.
- In our example it starts with 15 and looks for minimum value i.e. 15 among its neighbors 18, 15 and 18.

P

4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	15	18
6	32	20	14	13	17	28	15	15	14	16
2	27	19	11	12	12	20	14	13	17	18
3	26	14	10	11	13	16	11	12	16	17
9	24	10	14	16	19	9	12	17	18	21
10	16	8	12	11	14	8	13	18	16	21
5	7	5	5	5	8	12	12	13	15	16
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

Min(18,15,18)

How it works? (7/8)

In our example it starts with 15 and looks for minimum value i.e. 15 among its neighbours 18, 15 and 18.

P										
4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	**15**	18
6	32	20	14	13	17	28	15	15	14	16
2	27	19	11	12	12	20	14	13	17	18
3	26	14	10	11	13	16	11	12	16	17
9	24	10	14	16	19	9	12	17	18	21
10	16	8	12	11	14	8	13	18	16	21
5	7	5	5	5	8	12	12	13	15	16
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36
Q	**1**	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Min(19,15,14)

- The next number in the warping traversal path is 14. This process continues till we reach the bottom or the left axis of the table

P										
4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	**15**	18
6	32	20	14	13	17	28	15	15	**14**	16
2	27	19	11	12	12	20	14	13	17	18
3	26	14	10	11	13	16	11	12	16	17
9	24	10	14	16	19	9	12	17	18	21
10	16	8	12	11	14	8	13	18	16	21
5	7	5	5	5	8	12	12	13	15	16
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36
Q	**1**	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Min(19,15,14)

How it works? (5/8)
Local cost matrix (LCM)

- The full table will look like this:

P										
4	42	24	20	17	20	25	19	18	18	15
8	39	21	19	17	20	19	18	19	15	18
6	32	20	14	13	17	28	15	15	14	16
2	27	19	11	12	12	20	14	13	17	18
3	26	14	10	11	13	16	11	12	16	17
9	24	10	14	16	19	9	12	17	18	21
10	16	8	12	11	14	8	13	18	16	21
5	7	5	5	5	8	12	12	13	15	16
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

How it works? (8/8)

- ***Step 4:* Final Distance Calculation**
- Time normalised distance , D
- where k is the length of the series d.
- k = 12 in our case.
- D = (15 + 15 + 14 + 13 + 11 + 9 + 8 + 8 + 4 + 4 + 3 + 0) /12

= 104/12 = 8.63

$$D = \frac{\sum_{i=1}^{k} d(i)}{\sum_{i=1}^{k} k}$$

The final path looks like this

4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	**15**	18
6	32	20	14	13	17	28	15	15	**14**	16
2	27	19	11	12	12	20	14	**13**	17	18
3	26	14	10	11	13	16	**11**	12	16	17
9	24	10	14	16	19	**9**	12	17	18	21
10	16	8	12	11	14	**8**	13	18	16	21
5	7	5	5	5	**8**	12	12	13	15	16
4	3	**3**	**4**	**4**	7	13	14	14	17	17
1	**0**	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

- The warping path is the best alignment between 2 time series

- When determining the DTW distance between two time series, first an (n × m) local cost matrix (LCM) is calculated, where element (i, j) contains the distance between xi and yj .
- This distance is usually defined as the quadratic difference: d(xi , yj) = (xi – yj) 2 . Next, a warping path W = w1, w2, …, wK is determined, where max(n, m) ≤ K ≤ m+n–1.
- This path traverses the LCM under three constraints:
  - 1. Boundary condition: The path must start and end in the diagonal corners of the LCM: w1 = (1, 1) and wK = (n, m).
  - 2. Continuity: Only adjacent elements in the matrix are allowed for steps in the path. This includes diagonal adjacent elements. So, if wq = (i, j), then wq+1 is either element (i + 1, j), (i, j + 1) or (i+ 1, j + 1) for q = 1, …, K –1 and i = 1, …, n–1 and j = 1, …, m–1.
  - 3. Monotonicity: Subsequent steps in the path must be monotonically spaced in time

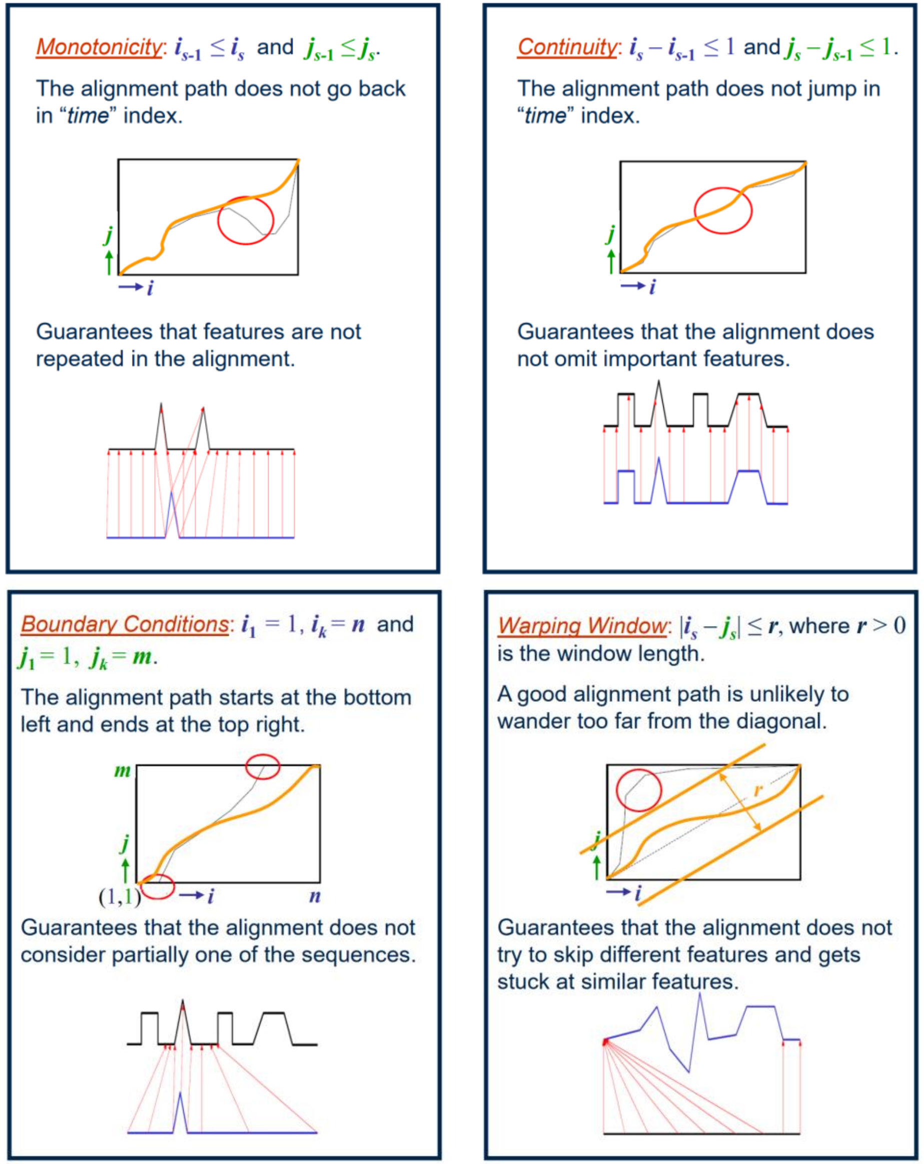
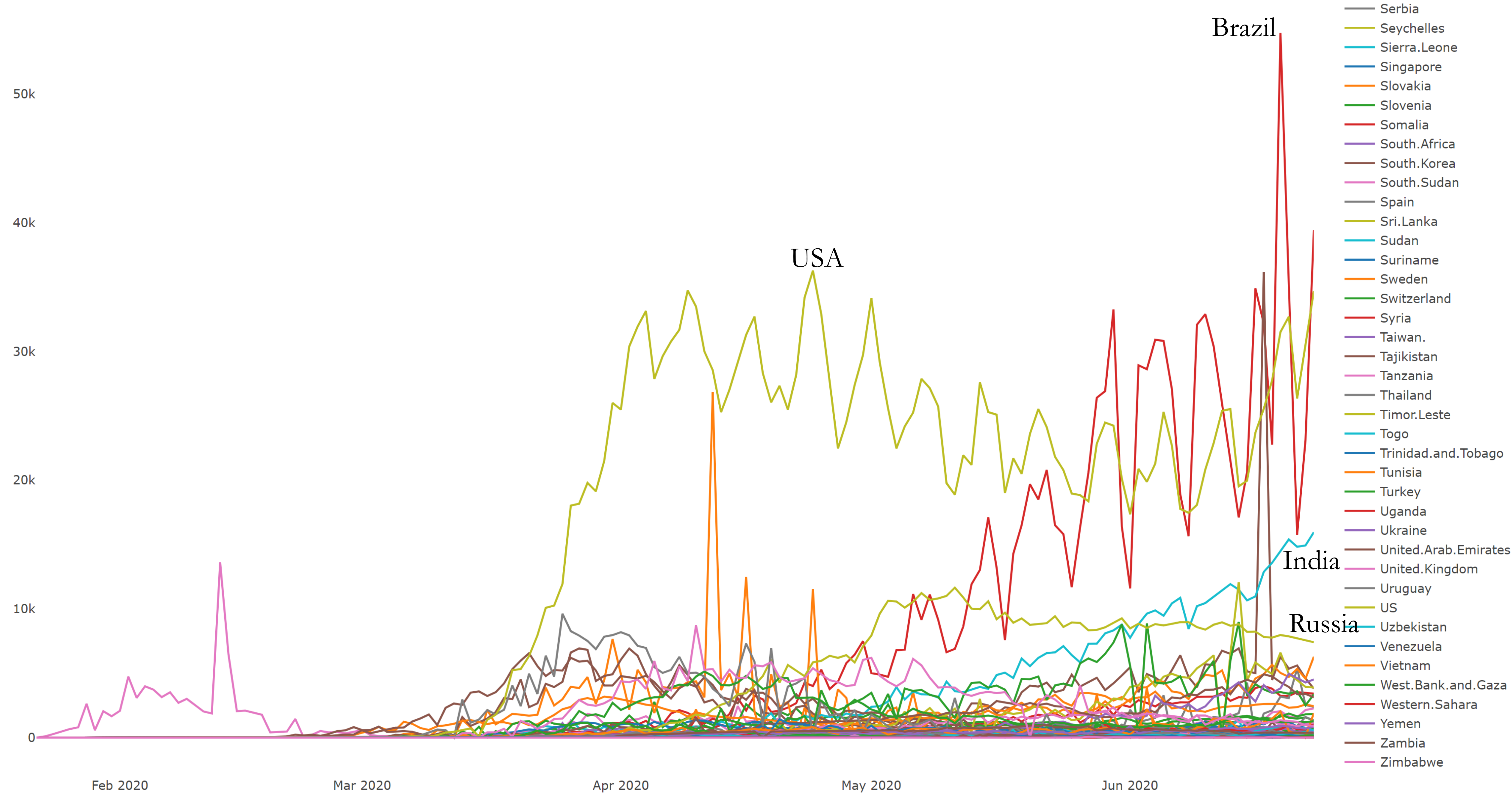


Table of content

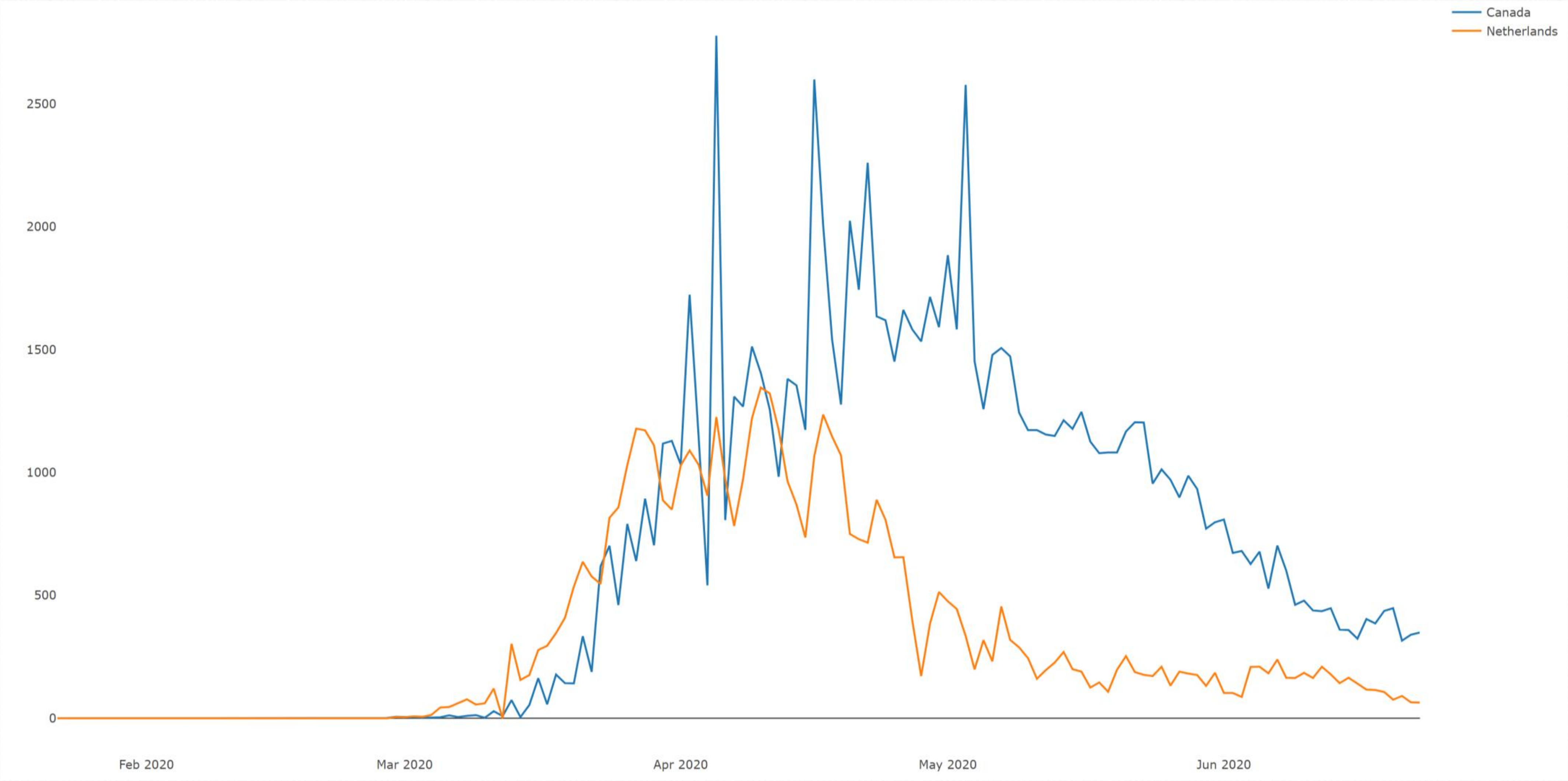
Descriptions:

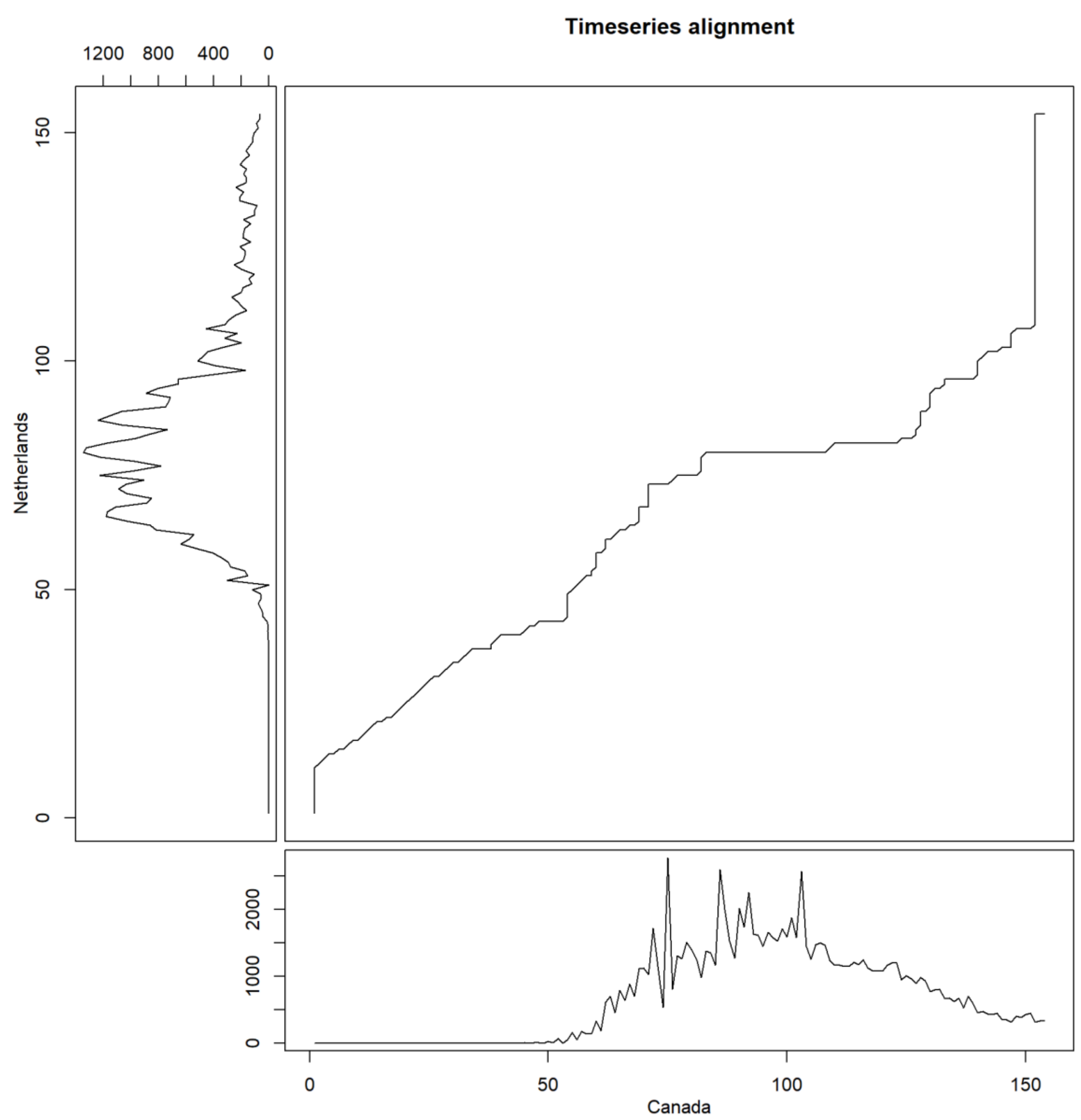
- The dataset consists of covid active cases from Jan 22 to June 23, 2020 of 187 countries
- Datapoint is daily measurement of new cases for each country (154 data points)

Date	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	Bahamas	Bahrain
2020-03-08	3	0	2	0	0	0	4	0	13	25	0	0	0
2020-03-09	0	2	1	0	0	0	0	0	15	27	0	0	10
2020-03-10	1	8	0	0	0	0	5	0	16	51	2	0	15
2020-03-11	2	2	0	0	0	0	2	0	21	64	0	0	85
2020-03-12	0	11	4	0	0	0	0	3	0	56	0	0	0
2020-03-13	0	10	2	0	0	1	12	4	72	202	4	0	0
2020-03-14	4	5	11	0	0	0	3	10	50	151	0	0	15
2020-03-15	5	4	11	0	0	0	11	8	47	205	8	0	4
2020-03-16	5	9	6	1	0	0	11	26	80	158	5	1	0
2020-03-17	1	4	6	37	0	0	12	26	75	314	0	0	14
2020-03-18	0	4	14	0	0	0	11	6	116	314	0	0	28
2020-03-19	0	5	13	14	0	0	18	31	113	367	16	2	22
2020-03-20	2	6	3	22	1	0	31	21	110	375	0	0	7
2020-03-21	0	6	49	13	1	0	30	24	280	426	9	1	20
2020-03-22	16	13	62	25	0	0	108	34	478	768	12	0	29
2020-03-23	0	15	29	20	1	2	35	41	133	892	7	0	43
2020-03-24	34	19	34	31	0	0	86	14	362	809	15	1	15
2020-03-25	10	23	38	24	0	0	0	16	320	305	6	0	27
2020-03-26	10	28	65	36	1	4	115	25	446	1321	29	4	39


Brazil
USA
India
Russia
50k
40k
30k
20k
10k
0
Feb 2020
Mar 2020
Apr 2020
May 2020
Jun 2020
Serbia
Seychelles
Sierra.Leone
Singapore
Slovakia
Slovenia
Somalia
South.Africa
South.Korea
South.Sudan
Spain
Sri.Lanka
Sudan
Suriname
Sweden
Switzerland
Syria
Taiwan.
Tajikistan
Tanzania
Thailand
Timor.Leste
Togo
Trinidad.and.Tobago
Tunisia
Turkey
Uganda
Ukraine
United.Arab.Emirates
United.Kingdom
Uruguay
US
Uzbekistan
Venezuela
Vietnam
West.Bank.and.Gaza
Western.Sahara
Yemen
Zambia
Zimbabwe

We will illustrate how to calculate the warping path of 2 countries, Canada and Netherland



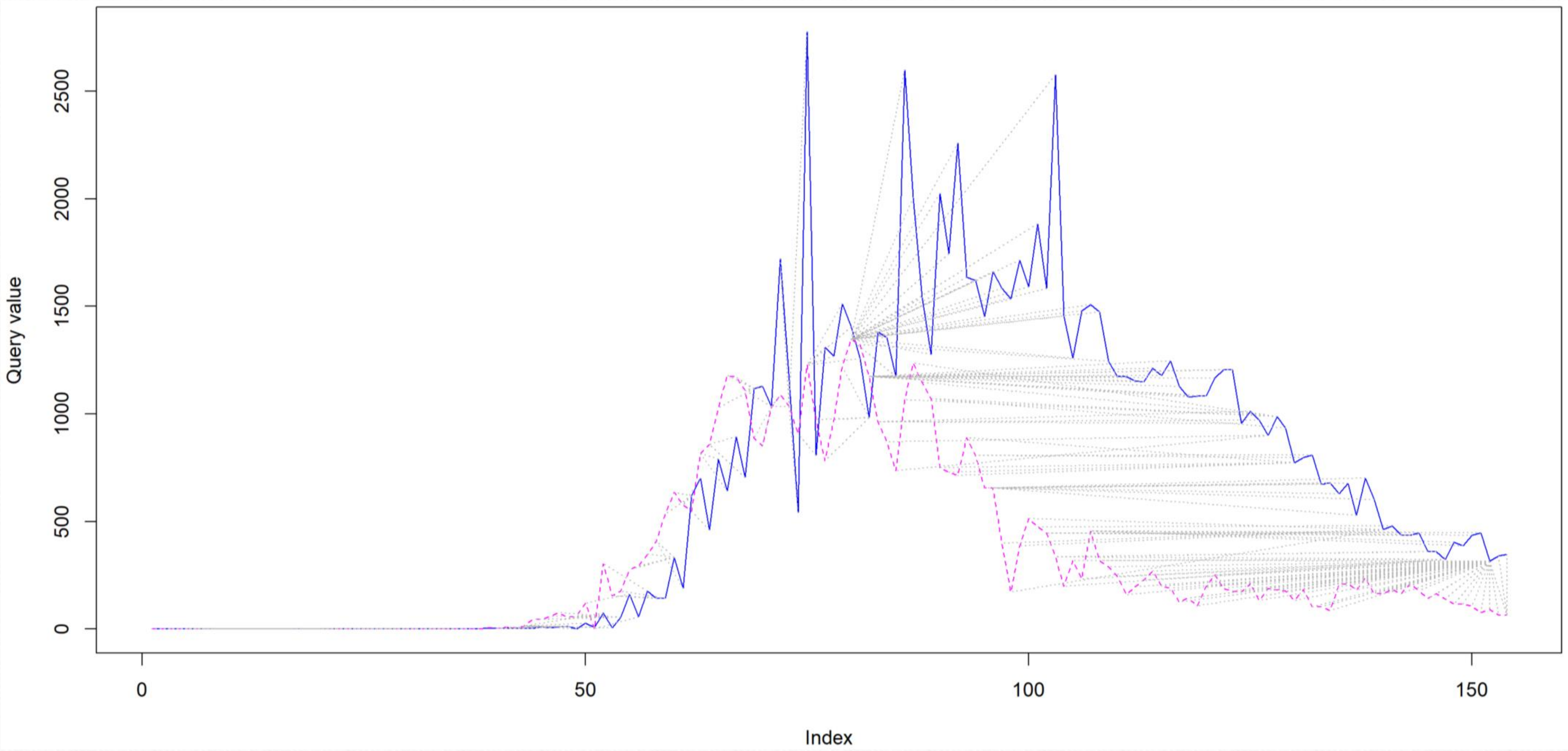


- The same plot can be made using type="threeway" that additionally shows the plots on x ad y axis.
- The dtw functions has to also have keep=TRUE defined

plot(dtw(df_canada_netherlands$Canada, df_canada_netherlands$Netherlands, keep=TRUE),
xlab="Canada", ylab="Netherlands", xaxp = c(0,10,10), yaxp = c(0,10,10), type="threeway")

Dtw step patterns

One can also plot the connections as below using type=“twoway”.



- DTW by default uses symmetric2.
- In tabs there are calculations performed for symmetric1 and symmetric2 step patterns.
- Because an array has it's point [1,1] in the upper left corner all matrices are rotated 90 degrees to the right compared to the results from dtw. Traditionally the 'beginning' of the dtw matrix should be in the lower left corner

```
> dtw(df_canada_netherlands$Canada, df_canada_netherlands$Netherlands)$stepPattern
Step pattern recursion:
g[i,j] = min(
     g[i-1,j-1] + 2 * d[i  ,j  ] ,
     g[i  ,j-1] +     d[i  ,j  ] ,
     g[i-1,j  ] +     d[i  ,j  ] ,
  )

 Normalization hint: N+M
> dtw(df_canada_netherlands$Canada, df_canada_netherlands$Netherlands, step.pattern = symmetric1)$stepPattern
Step pattern recursion:
g[i,j] = min(
     g[i-1,j-1] +     d[i  ,j  ] ,
     g[i  ,j-1] +     d[i  ,j  ] ,
     g[i-1,j  ] +     d[i  ,j  ] ,
  )

 Normalization hint: NA
```

Alignment plot
Creating a zeroes matrix

```
dtw_matrix = matrix(rep(c(0),13225), nrow=115, ncol=115,
byrow = TRUE)
dtw_matrix
```

- The matrix is calculated according to the step pattern formula.
- Additionally d(0,j)=d(i,0)=d(0,0)= That is why it is not included in calculations - will never be accounted for by min() function. I will be using the default euclidean distance between points

First element:

```
a1 = df_canada_netherlands$Canada
length(a1)
a2 = df_canada_netherlands$Netherlands

dtw_matrix[1,1] = sqrt(a1[1]^2 + a2[1]^2)
dtw_matrix

for (i in 2:115){
 dtw_matrix[i,1] = sqrt((a1[i] - a2[1])^2) + dtw_matrix[i-1,1]
}
dtw_matrix
```

First column:

```
for (i in 2:115){
 dtw_matrix[i,1] = sqrt((a1[i] - a2[1])^2) + dtw_matrix[i-1,1]
}
dtw_matrix
```

First row:

```
for (j in 2:115){
 dtw_matrix[1,j] = sqrt((a1[1] - a2[j])^2) + dtw_matrix[1,j-1]
}
dtw_matrix
```

The rest of the matrix:

```
for (i in 2:115){
 for (j in 2:115){
  dtw_matrix[i,j] = sqrt((a1[i] - a2[j])^2) + min(dtw_matrix[i,j-1],
dtw_matrix[i-1,j], dtw_matrix[i-1,j-1] + sqrt((a1[i] - a2[j])^2))
 }
}
dtw_matrix
```

Dtw_matrix (illustrated)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
9	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
10	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
15	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
16	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
17	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
18	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
19	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
20	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
21	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
22	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
23	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
24	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
25	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
26	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
27	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
28	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
29	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
30	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
31	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
32	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
33	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
34	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
35	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
36	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
37	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
38	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
39	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
40	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
41	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
42	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
43	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
44	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37
45	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
46	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54
47	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
48	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77
49	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79
50	108	108	108	108	108	108	108	108	108	108	108	108	108	108	108	108	108	108
51	117	117	117	117	117	117	117	117	117	117	117	117	117	117	117	117	117	117
52	191	191	191	191	191	191	191	191	191	191	191	191	191	191	191	191	191	191
53	196	196	196	196	196	196	196	196	196	196	196	196	196	196	196	196	196	196
54	250	250	250	250	250	250	250	250	250	250	250	250	250	250	250	250	250	250
55	413	413	413	413	413	413	413	413	413	413	413	413	413	413	413	413	413	413
56	470	470	470	470	470	470	470	470	470	470	470	470	470	470	470	470	470	470
57	648	648	648	648	648	648	648	648	648	648	648	648	648	648	648	648	648	648
58	791	791	791	791	791	791	791	791	791	791	791	791	791	791	791	791	791	791
59	933	933	933	933	933	933	933	933	933	933	933	933	933	933	933	933	933	933
60	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267	1267
61	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456	1456
62	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075	2075
63	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777	2777
64	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238	3238

Warping path

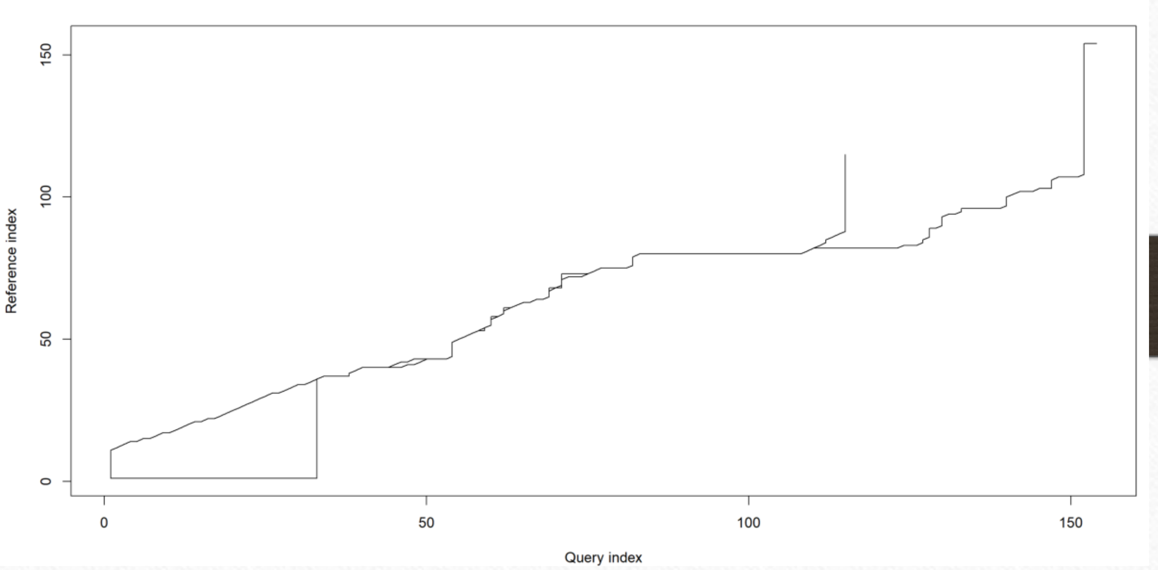


Table of content

For Internal criterion. A good clustering will produce high quality clusters in which:

- the intra-class (that is, intra-cluster) similarity is high
- the inter-class similarity is low
- The measured quality of a clustering depends on both the document representation and the similarity measure used
- Internal criterion is used when we don’t have a ground of truth or expert knowledge.
    - Silhouette coefficient
    - Dunn index

There is clValid package in R measures internal quality of clusters based on 3 criteria:

- Connectivity
- Average Silhouette width: measures of compactness and separation of the clusters.
    - Observations with a large Si (almost 1) are very well clustered.
    - A small Si (around 0) means that the observation lies between two clusters.
    - Observations with a negative Si are probably placed in the wrong cluster.
- Dunn index: measures of compactness and separation of the clusters

The connectivity should be **minimized**, while both the Dunn index and the silhouette width should be **maximized**

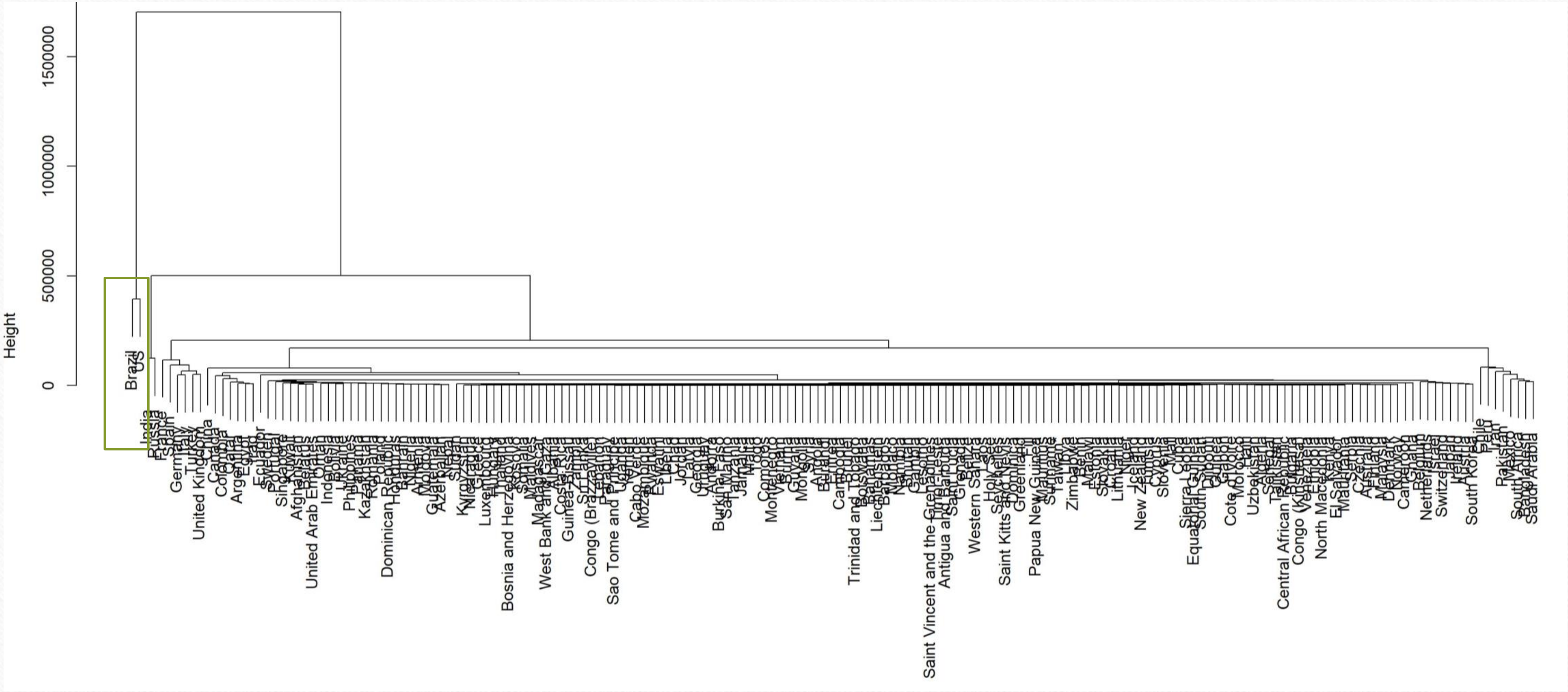
```
internal = clValid(df_covid_new.T1, nClust = 2:5, clMethods = clmethods, validation = "internal")
summary(internal)
#Clustering Methods:
#  hierarchical kmeans pam

#Cluster sizes:
#  2 3 4 5

#Validation Measures:
#  2       3       4       5

#hierarchical Connectivity   2.9290  5.8579  9.7159 12.6448
#Dunn            1.0743  1.4232  0.6891  0.7203
#Silhouette      0.9541  0.9396  0.8821  0.8471
#kmeans        Connectivity   4.8579  5.8579 15.3139 26.3238
#Dunn            0.5902  1.4232  0.1396  0.1227
#Silhouette      0.9512  0.9396  0.8464  0.8198
#pam           Connectivity   2.9290 15.0246 17.9536 20.9988
#Dunn            1.0743  0.0534  0.1359  0.1359
#Silhouette      0.9541  0.8326  0.8179  0.8335

#Optimal Scores:

#              Score  Method       Clusters
#Connectivity 2.9290 hierarchical 2
#Dunn         1.4232 hierarchical 3
#Silhouette   0.9541 hierarchical 2

#Hierarchical method, k=2
```

Comment:

- According to connectivity and Silhouette score, hierarchical method with k =2 is optimal
- According to Dunn score, , hierarchical method with k =3 is optimal

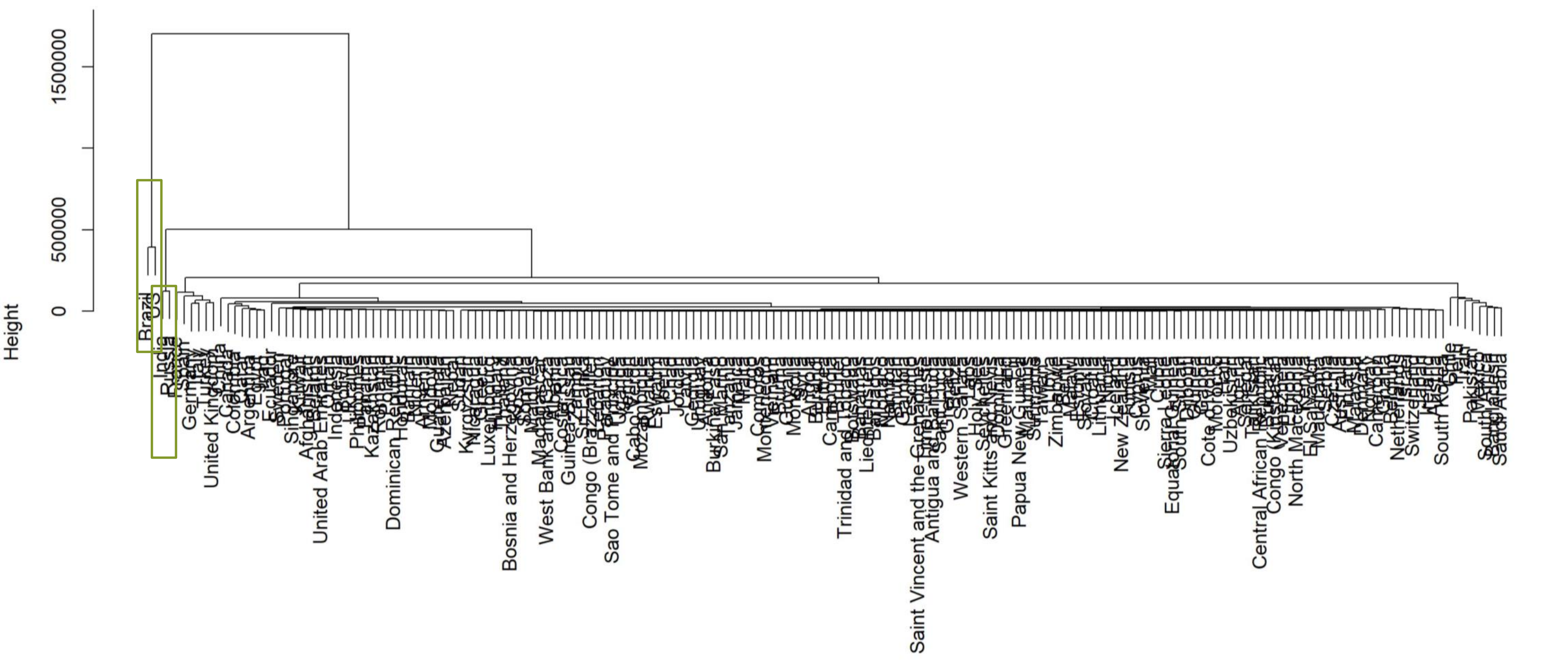
Result of Hierarchal clustering with k =2

- **US, Brazil : group 2**
- **The rest in group 1**



Result of Hierarchal clustering with k =2

- **US, Brazil : group 2**
- **India, Russia: group 3**
- **The rest in group 1**



Conclusion

- Dynamic time warping (DTW) has been used as a technique to calculate more robust distance for time series data, since it allows elastic shifting of sequence in order to detect similar shapes with different phases.
- Furthermore, it can be used to measure similarity between sequences of different lengths

- However, DTW has its own limitations; it is quite expensive to compute and it does not obey the triangular inequality,showing resistance to indexing.
- In order to overcome these limitations, several studies have defined computationally cheap lower bounds Lb(q→, s→) for DTW distance and indices that are used to prune fast sequences that may not be included in the query results