КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ТАРАСА ШЕВЧЕНКА
МЕХАНІКО-МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ТЕОРІЇ ЙМОВІРНОСТЕЙ, СТАТИСТИКИ ТА
АКТУАРНОЇ МАТЕМАТИКИ

КУРСОВА РОБОТА

НА ТЕМУ:

Алгоритми динамічної трансформації часу у статистиці

Виконала:
студентка 3 курсу
механіко-математичного факультету
групи «Статистика»
Шкляр Ірина Володимирівна

Науковий керівник:
доктор фізико-математичних наук
професор кафедри теорії ймовірностей,
статистики та актуарної математики
Майборода Ростислав Євгенович

КИЇВ – 2023

Зміст

Вступ

Алгоритми динамічної трансформації часу широко використовуються в сучасному світі. Вони дозволяють знайти оптимальну відповідність між часовими рядами, та є дуже важливими в машинному навчанні, особливо в розпізнаванні мови. Вперше застосовані у випадку, де два мовних сигнали представляють одну й ту саму вихідну сказану фразу. Але з часом було знайдено застосування їх і в інших областях.

В своїй курсовій роботі я буду використовувати їх для аналізу даних covid-19 за 2020-2023 роки. Ці дані включають в собі інформацію про Німеччину, Австрію, Угорщину, Польщу та Румунію. Аналізувати я буду саме підтверджені випадки зараження ковідом. Для цього я буду використовувати мову програмування R та бібліотеку dtw.

1. Теоретична частина

Алгоритм динамічної трансформації тимчасової шкали (DTW-алгоритм, від англ. dynamic time warping) - алгоритм, що дозволяє знайти оптимальну відповідність між часовими рядами.

Часові ряди - широко поширений тип даних, що, фактично, зустрічається в будь-якій науковій галузі, і порівняння двох послідовностей є стандартним завданням. Для обчислення відхилення буває досить простого виміру відстані між компонентами двох послідовностей (евклідова відстань). Однак часто дві послідовності мають приблизно однакові загальні форми, але ці форми не вирівняні по осі X. Щоб визначити подібність між такими послідовностями, ми повинні «деформувати» вісь часу однієї (або обох) послідовності, щоб досягти кращого вирівнювання.

Алгоритм

Вимірювання відстані між двома тимчасовими рядами потрібно для того, щоб визначити їхню подібність і класифікацію. Таким ефективним вимірюванням є евклідова метрика. Для двох часових послідовностей це просто сума квадратів відстаней від кожної n-ої точки однієї послідовності до n-ої точки іншої.
Однак використання евклідової відстані має істотний недолік: якщо два часові ряди однакові, але один з них незначно зміщений у часі (вздовж осі часу), то евклідова метрика може вважати, що ряди відрізняються один від одного. DTW-алгоритм було введено для того, щоб подолати цей недолік і надати наочний вимір відстані між рядами, не звертаючи уваги як на глобальні, так і локальні зрушення на часовій шкалі.

Класичний алгоритм з формулами

Розглянемо два часові ряди – Q довжини n та C довжини m:

$Q = q_1, q_2, \ldots, q_i, \ldots, q_n;$ (1)

$C = c_1, c_2, \ldots, c_j, \ldots, c_m.$ (2)

На першому етапі ми будуємо матрицю d розмірності n x m (матриця відстаней), в якій елемент d з індексами i та j (у подальшому елемент з індексом позначаємо як q_i – q з індексом i) – відстань d(q_i,c_j) між двома точками: q_i та c_j. Зазвичай використовується евклідова відстань: d(q_i,c_j) = (q_i - c_j)^2 або замість квадрата модуль. Кожен елемент (i,j) матриці відповідає вирівнюванню між точками q_i та c_j.

На другому етапі будуємо матрицю трансформацій D, кожний елемент якої рахується із наступної наведеної формули:

$$D_{i\ j} = d_{i\ j} + min(D_{i-1\ j}, D_{i-1\ j-1}, D_{i\ j-1}).$$

Після заповнення матриці трансформації ми переходимо до заключного етапу, який полягає в тому, щоб побудувати деякий оптимальний шлях трансформації та dtw відстань.

Шлях трансформації W - це набір суміжних елементів матриці, який встановлює відповідність між Q і C. Він являє собою шлях, який мінімізує загальну відстань між Q і C. k-ий елемент шляху W визначається як $w_k = (i,j)_k, \quad d(w_k) = d(q_i, c_j) = (q_i - c_j)^2$. Тому:

$$W = w_1, w_2, \ldots, w_k, \ldots, w_K; \qquad max(m,n) \leqslant K < m+n,$$

де K – довжина шляху.

Але шлях трансформації має задовольняти наступні умови:

- Граничні умови: початок шляху w_1 = (1,1), а кінець – w_K = (n,m). Це обмеження гарантує, що шлях трансформації містить усі точки обох часових рядів.
- Неперервність: будь-які два суміжні елементи шляху W, $w_k = (w_i, w_j)$ и $w_{k+1} = (w_{i+1}, w_{j+1})$, задовольняють наступні нерівності:

$$w_i - w_{i+1} \leqslant 1,\ w_j - w_{j+1} \leqslant 1.$$

Це обмеження гарантує, що шлях трансформації пересувається на один крок за один раз. Тобто обидва індекси i та j можуть збільшитись тільки на 1 на кожному кроці шляху.

- Монотонність: будь-які два суміжні елементи шляху W, $w_k = (w_i, w_j)$ и $w_{k-1} = (w_{i-1}, w_{j-1})$, задовольняють наступні нерівності:

$$w_i - w_{i-1} \geqslant 0,\ w_j - w_{j-1} \geqslant 0\,.$$

Це обмеження гарантує, що шлях трансформації не повертатиметься назад до пройденої точки. Тобто обидва індекси і та j або залишаються незмінними, або збільшуються (але не зменшуються).

Хоча існує велика кількість шляхів трансформації, що задовольняють всі вищевказані умови, проте нас цікавлять тільки той шлях, який мінімізує dtw відстань.

dtw відстань між двома послідовностями розраховується на основі оптимального шляху трансформації за допомогою формули:

$$DTW(Q,C) = min\left\{\frac{\sum_{k=1}^{K} d(w_k)}{K}\right\}.$$

2. Практична частина

2.1. Підготовка до роботи з dtw

Дані про covid-19 для курсової роботи я взяла з першоджерела [1] у форматі csv. Обробивши початкові дані: перебравши і виділивши тільки потрібні нам країни в текстовому редакторі, я розпочала їх аналіз в R.

Спочатку я вирішила подивитися як дані будуть виглядати графічно: графік захворювань на ковід для 5 країн.
Спочатку я вирішила подивитися як дані для п’яти країн будуть відображені на одному графіку, за допомогою функції `plot()`:

```
> plot(data_dtw[,7])
```

де `data_dtw` – це наш набір даних, прочитаний RStudio за допомогою функції `read.csv`, `[,7]` – ми беремо тільки перші 7 стовпчиків в таблиці даних, де написані саме дані про захворюваність. Результат – на рисунку 1.

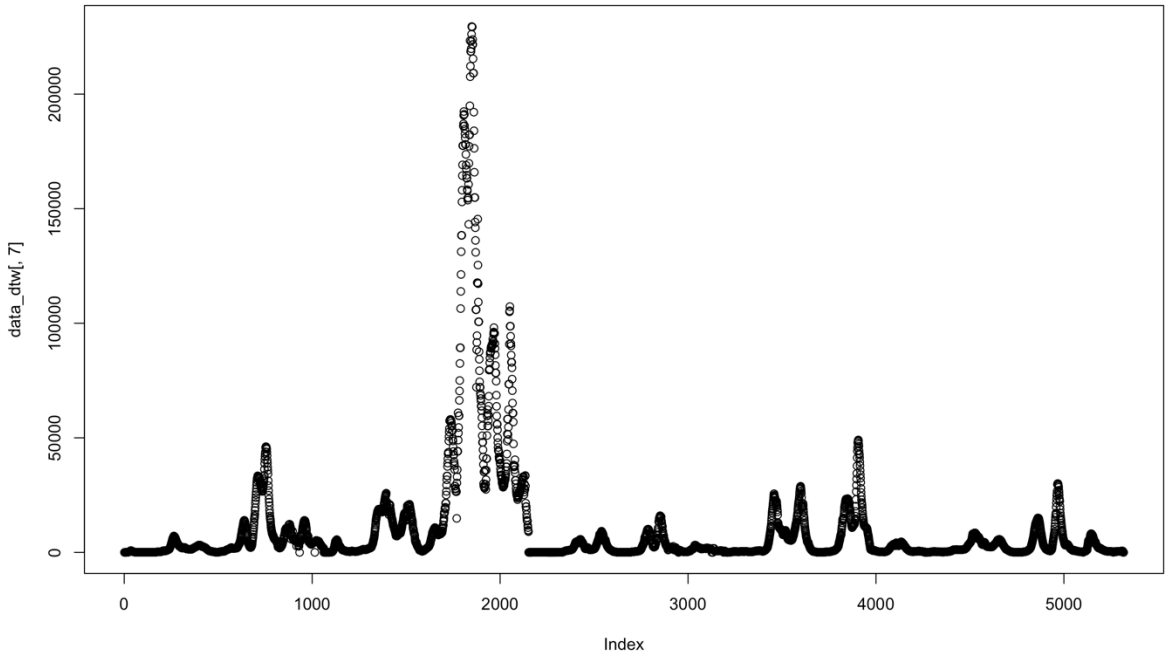


Рис.1

Наступним кроком, я зробила графіки п’яти країн на різних малюнках також за допомогою функції `plot()`:

```
> plot(data_dtw[1:1061,7])
```

Тут 1:1061 – номера рядків таблиці даних, по яким будувався графік, для кожної країни вони різні:
1:1061 – Австрія, рисунок 2,
1062:2151 – Німеччина, рисунок 3,
2152:3205 – Угорщина, рисунок 4,
3206:4258 – Польща, рисунок 5,
4259:5318 – Румунія, рисунок 6.

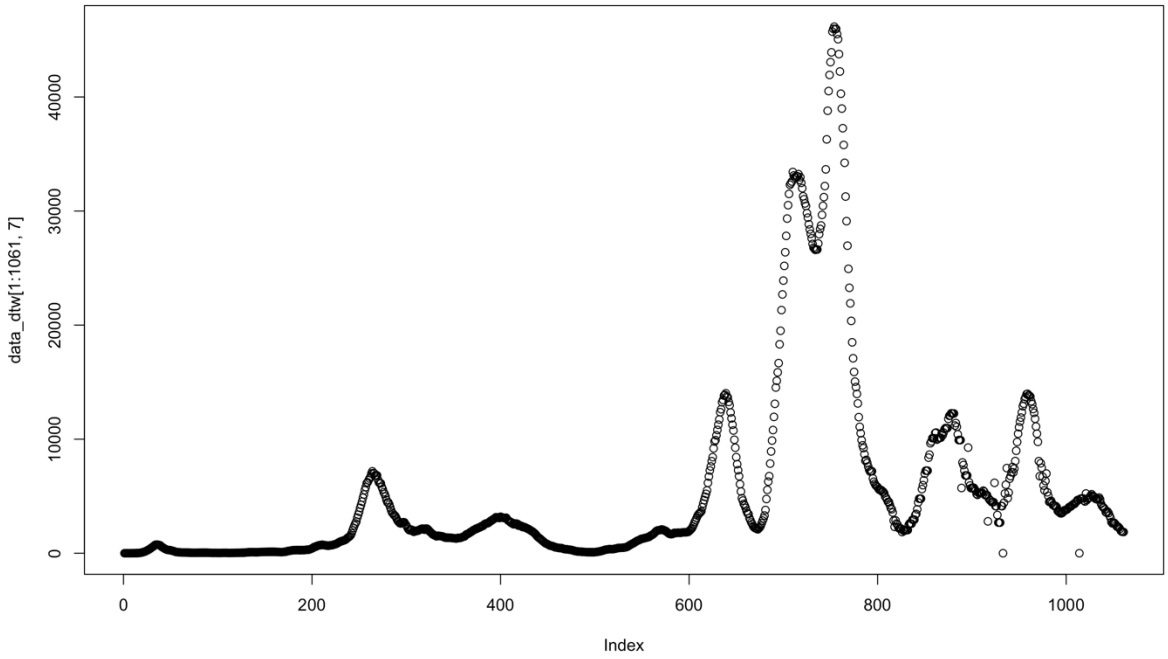


Рис.2

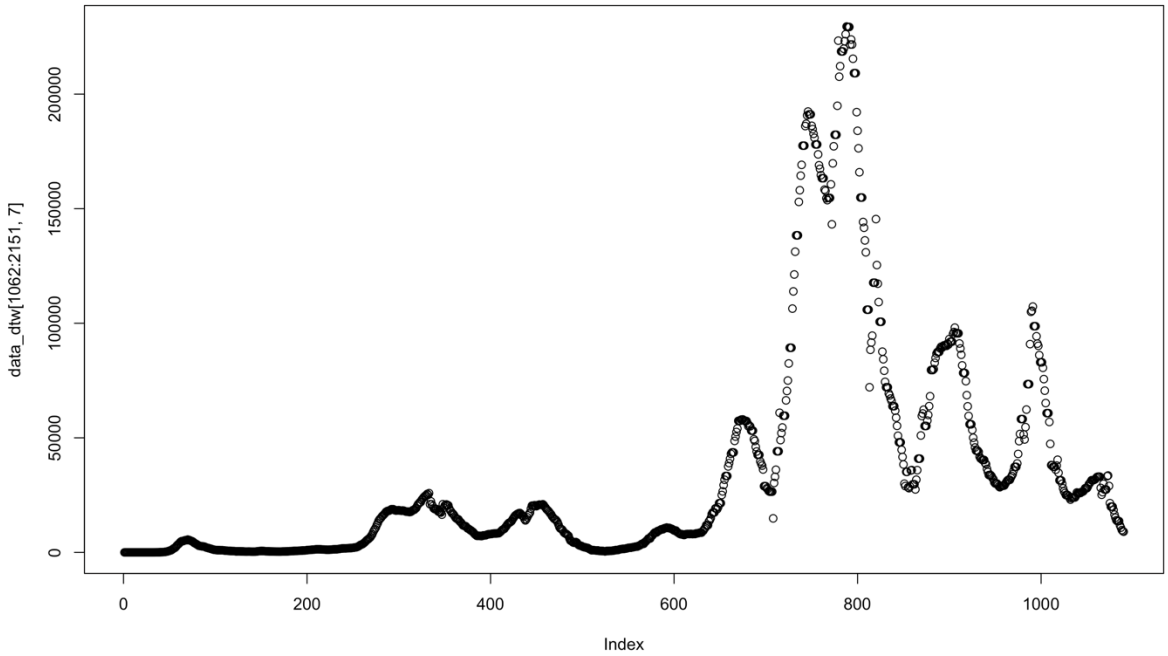

data_dtw[1062:2151, 7]
0
50000
100000
150000
200000
0
200
400
600
800
1000
Index

Рис.3

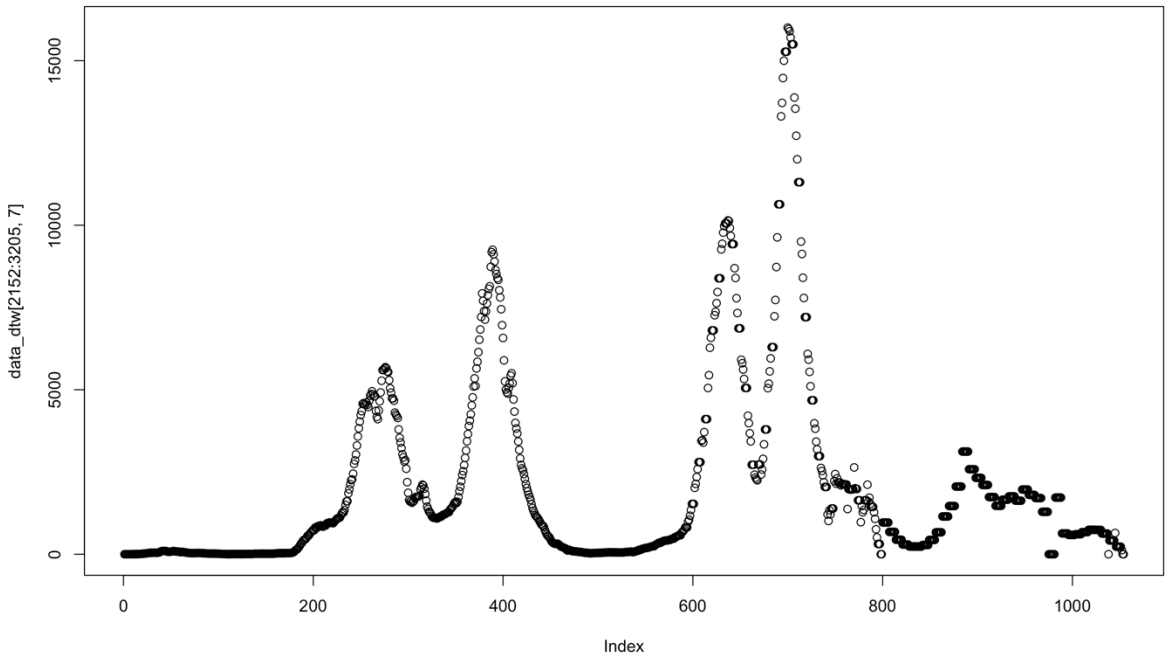

data_dtw[2152:3205, 7]
0
5000
10000
15000
0
200
400
600
800
1000
Index

Рис.4

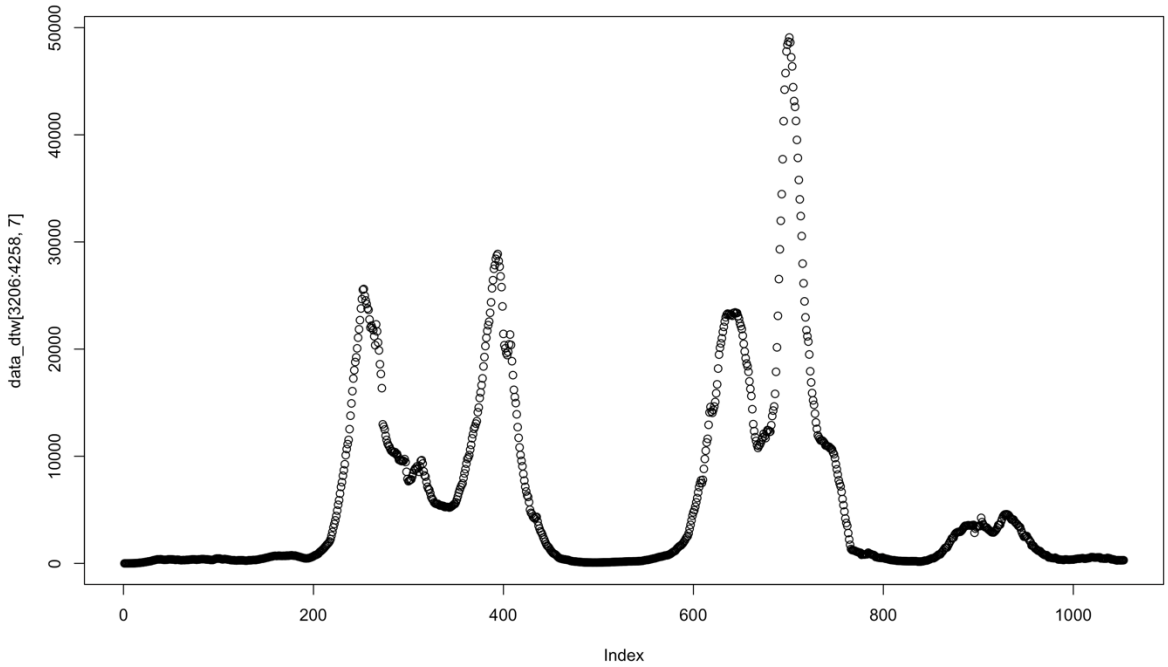

data_dtw[3206:4258, 7]
0
10000
20000
30000
40000
50000
0
200
400
600
800
1000
Index

Рис.5

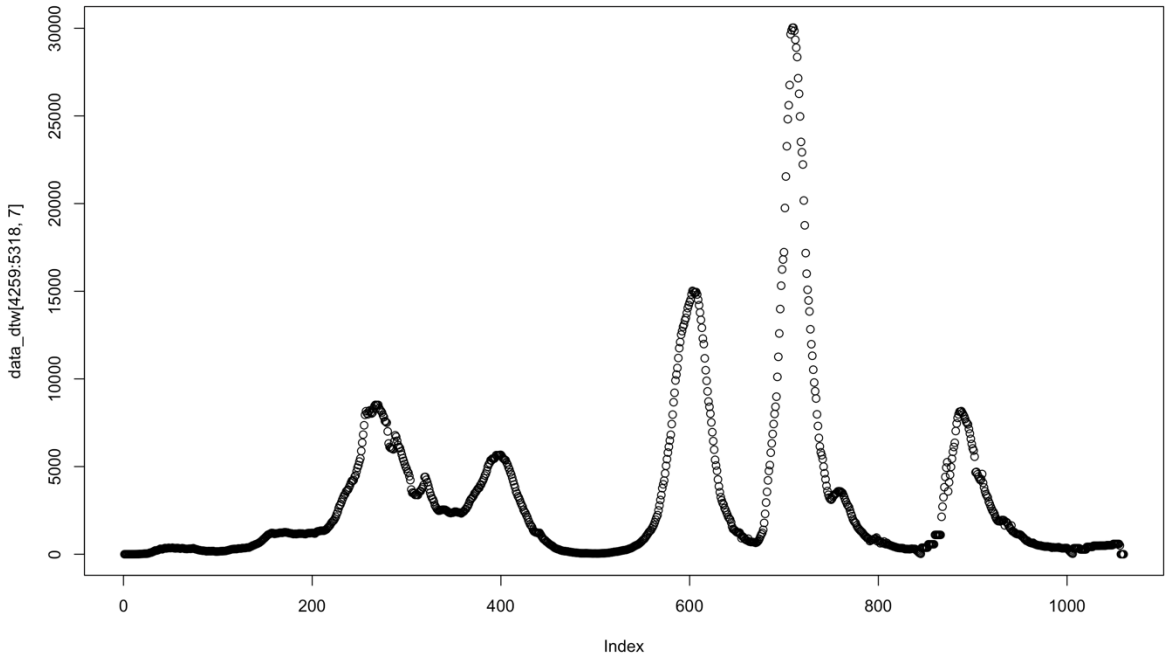

data_dtw[4259:5318, 7]
0
5000
10000
15000
20000
25000
30000
0
200
400
600
800
1000
Index

Рис.6

Далі наведені графіки для 5 країн, розташованих на одному рисунку за допомогою функцій `plot()` та `lines()`, рисунок 7:

```
> plot(data_dtw[1062:2151,7], col='purple')    – Німеччина
> lines(data_dtw[1:1061,7], col='blue')        – Австрія
> lines(data_dtw[2152:3205,7], col='pink')     – Угорщина
> lines(data_dtw[3206:4258,7], col='red')      – Польща
> lines(data_dtw[4259:5318,7], col='green')    – Румунія
```

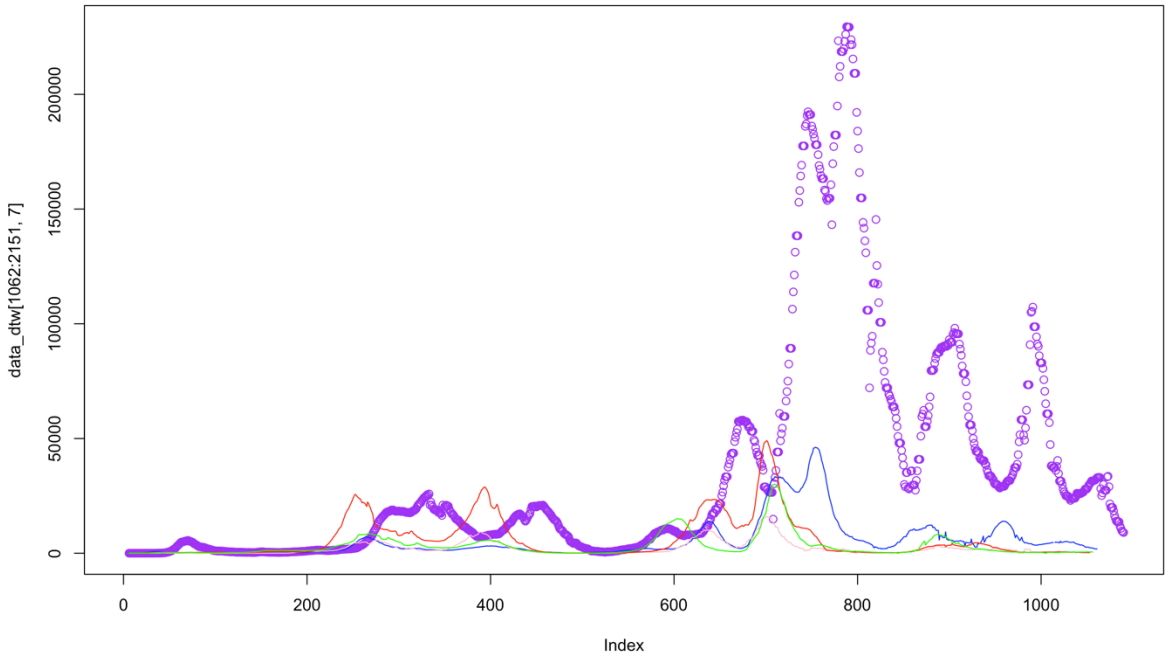


Рис.7

Бачимо, що лідером по зараженню covid-19 на піку захворюваності була Німеччина. Тому її я порівнювати з іншими країнами не буду, оскільки це очевидно.

Далі щоб проводити роботу з алгоритмами динамічної трансформації часу, потрібно в таблиці даних `data_dtw` замінити N/A на нулі (щоб був числовий формат). За замовчуванням, коли завантажуєш дані, в пустих клітинках – де значення нульові, стоять N/A, які не дозволяють фрейм даних перевести в часовий ряд. Тому я використала такий код, який заміняє n/a на 0:

```
> data_dtw[is.na(data_dtw)] <- 0
```

2.2. Робота з dtw

Щоб використовувати бібліотеку dtw (алгоритми динамічної трансформації часу), її потрібно завантажити та підключити в RStudio. Далі, щоб використовувати в роботі з даними функції dtw, необхідно перевести таблицю даних з формату csv у формат часових рядів.
Для цього я розділила фрейм даних на 5 частин, де в кожній знаходяться дані окремо про кожну з країн:

```
> dt1 <- data_dtw[1:1061,7]
> dt2 <- data_dtw[1062:2151,7]
> dt3 <- data_dtw[2152:3205,7]
> dt4 <- data_dtw[3206:4258,7]
> dt5 <- data_dtw[4259:5318,7]
```

Після цього я перевела ці дані у часові ряди (Time-Series) за допомогою функції `ts()`:

```
> timese1 <- ts(dt1)
> timese3 <- ts(dt3)
> timese4 <- ts(dt4)
> timese5 <- ts(dt5)
```

Тепер для кожної країни створено свій часовий ряд з якими надалі ми і будемо працювати та використовувати алгоритми динамічної трансформації часу.

Із бібліотеки dtw я використала функцію, яка так і називається: `dtw()`. Використаємо `keep = TRUE`, щоб ми могли зробити матрицю трансформації, `twoway` рисує порівняння точки в точку з відповідними лініями, `threeway` перевіряє часові ряди та їх криву трансформації.

Найперше подивимось криву трансформації для часових рядів Австрії та Угорщини, рисунок 8:

```
> t2 <- dtw(timese1, timese3, keep=TRUE)
> plot(t2, type="threeway")
```

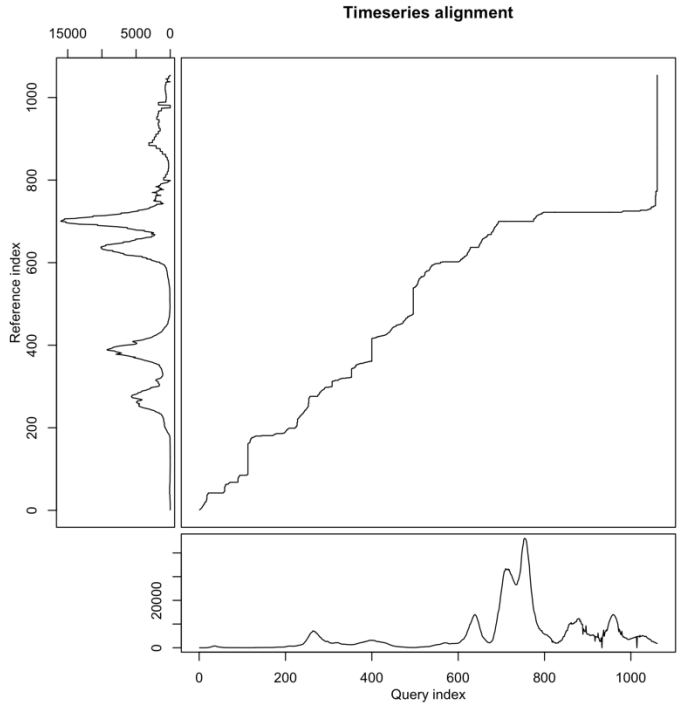


Рис.8

Далі подивимось графіки twoway точки в точку з відповідними лініями. `step=rabinerJuangStepPattern(6,"c")` означає, що ми взяли шлях Рабінера-Джуанга, який має сім класифікацій та чотири підтипи, в залежності від нахилу ліній і ми вибрали саме шосту класифікацію та підтип “c”, який нам допоможе відобразити алгоритм dtw. Графік на рисунку 9:

```
> plot(dtw(timese1,timese3,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

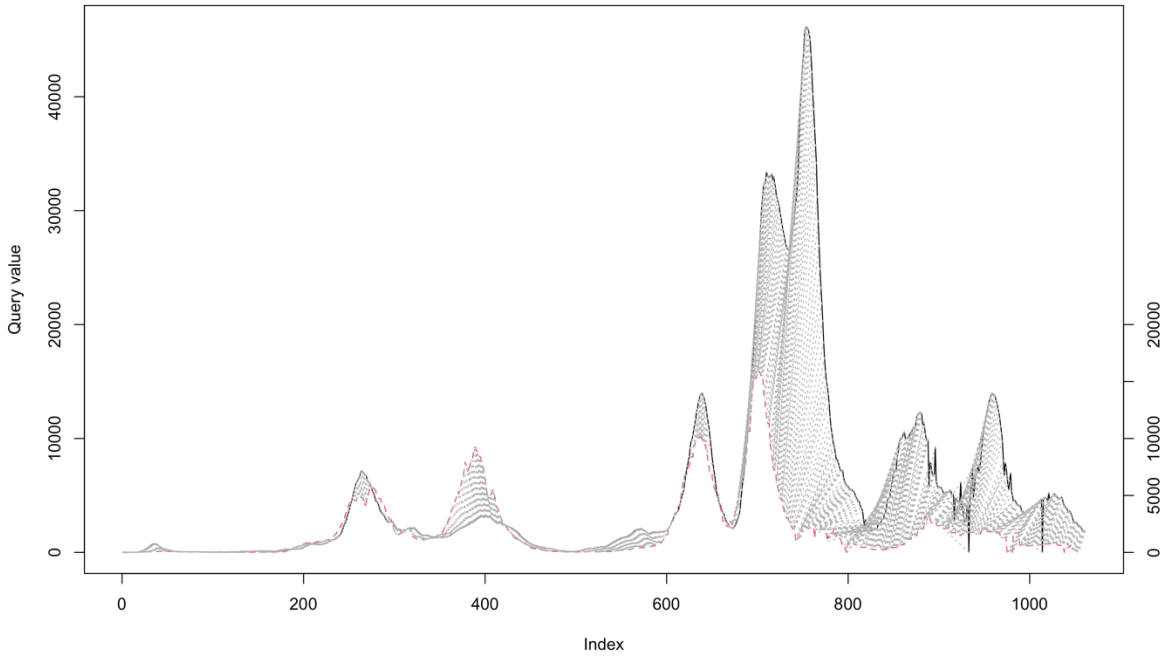


Рис.9

Якщо подивитися дані dtw, рисунок 9.1, то там написано, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 2356393, а нормована відстань – 1114. Також з рисунку 9 та кривої трансформації на рисунку 8 видно, що графіки відрізняються суттєво, в Австрії захворюваність була більшою ніж в Угорщині. На кривій трансформації добре видно стрибки, особливо в правому верхньому кутку, де велика відстань між точками графіків (місця, де графіки дуже відрізняються).

```
t2                      List of  20
  $ costMatrix          : num [1:1061,
  $ directionMatrix     : int [1:1061,
  $ stepPattern         : 'stepPattern'
   ..- attr(*, "npat")= num 3
   ..- attr(*, "norm")= chr "N+M"
  $ N                   : int 1061
  $ M                   : int 1054
  $ call                : language dtw(
  $ openEnd             : logi FALSE
  $ openBegin           : logi FALSE
  $ windowFunction      :function (iw,
  $ jmin                : int 1054
  $ distance            : num 2356393
  $ normalizedDistance: num 1114
```

Рис.9.1

Крива трансформації для часових рядів Австрії та Польщі, рисунок 10:

```
> t3 <- dtw(timese1, timese4, keep=TRUE)
> plot(t3,type="threeway")
```

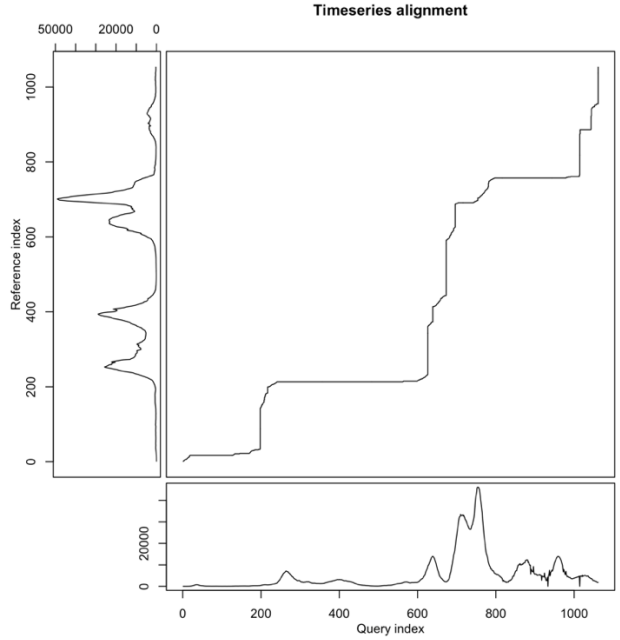


Рис.10

Відповідний графік twoway dtw, рисунок 11:

```
> plot(dtw(timese1,timese4,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

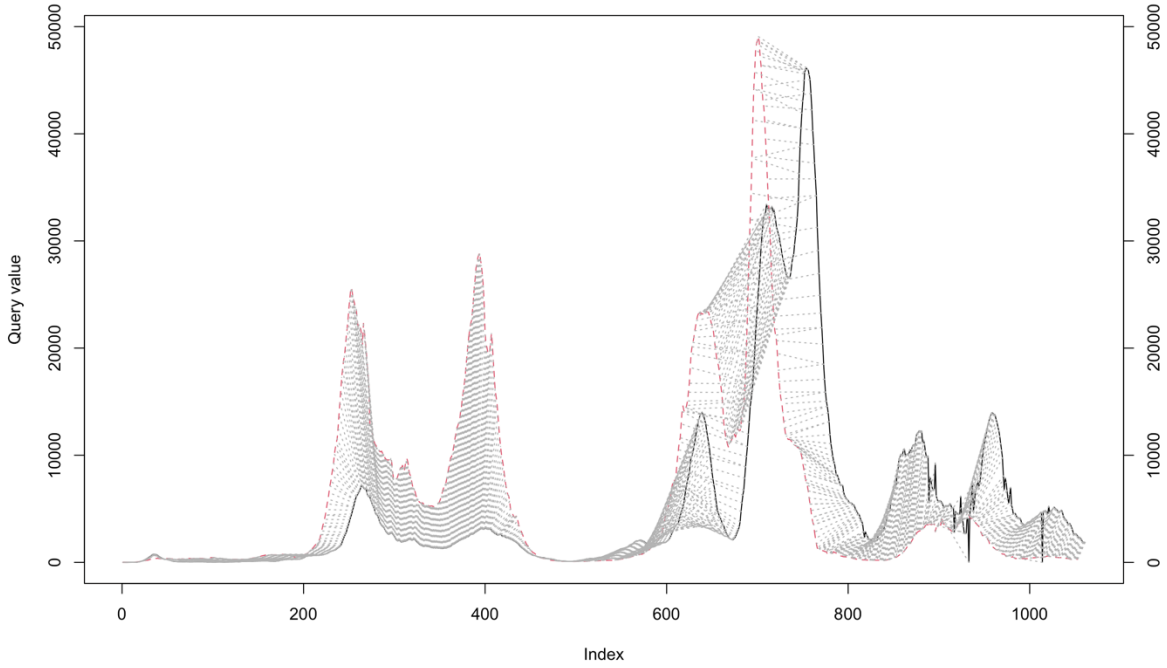


Рис.11

По даним dtw написано, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 2789111, а нормована відстань – 1319. З рисунку 10 та кривої трансформації на рисунку 11 та з відстаней видно, що графіки відрізняються суттєвіше ніж попередні. Спочатку захворюваність переважала в Польщі і піковий стрибок був вище, але з часом в Австрії захворюваність виросла та стала більшою ніж у Польщі, в кінці часового ряду. На кривій трансформації добре видно стрибки, де велика відстань між точками графіків (місця, де графіки дуже відрізняються).

Крива трансформації для часових рядів Австрії та Румунії, рисунок 12:

```
> t4 <- dtw(timese1, timese5, keep=TRUE)
> plot(t4,type="threeway")
```

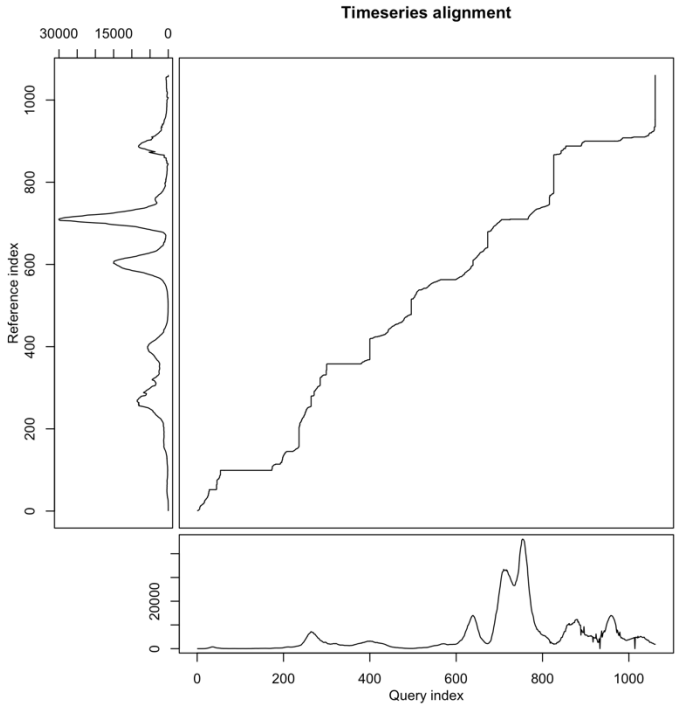


Рис.12

Відповідний графік twoway dtw, рисунок 13:

```
> plot(dtw(timese1,timese5,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

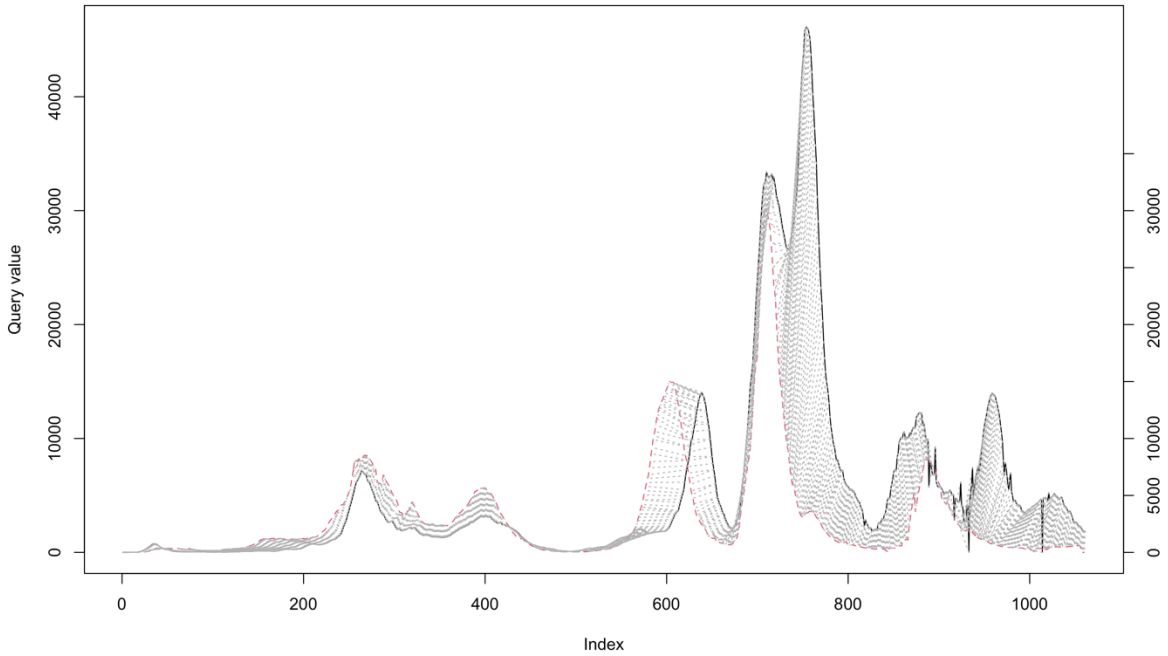


Рис.13

По даним dtw написано, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 1226970, а нормована відстань – 578. З рисунку, кривої трансформації та з відстаней видно, що графіки Австрії та Румунії відрізняються не дуже суттєво, видно зміщення у часі майже однакових даних. Хоча загалом захворюваність в Австрії була трішки вища ніж в Румунії. На кривій трансформації видно стрибки, де велика відстань між точками графіків (місця, де графіки дуже відрізняються).

Крива трансформації для часових рядів Польщі та Угорщини, рисунок 14:

```
> t8 <- dtw(timese3, timese4,keep=TRUE)
> plot(t8,type="threeway")
```

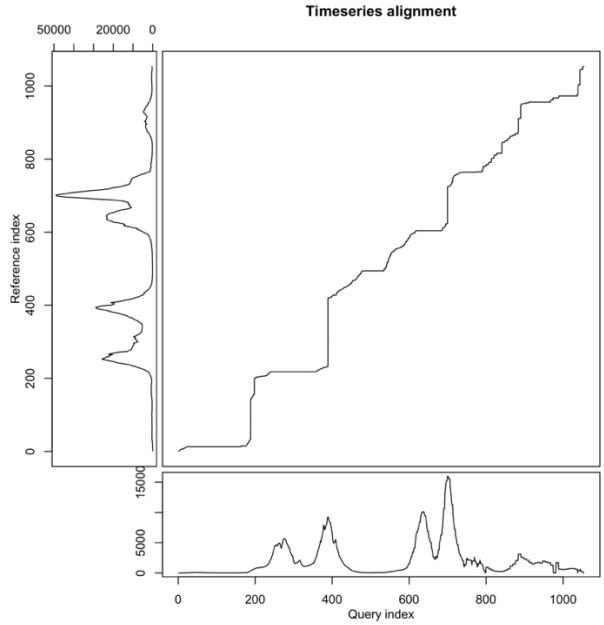


Рис.14

Відповідний графік twoway dtw, рисунок 15:

```
> plot(dtw(timese4,timese3,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

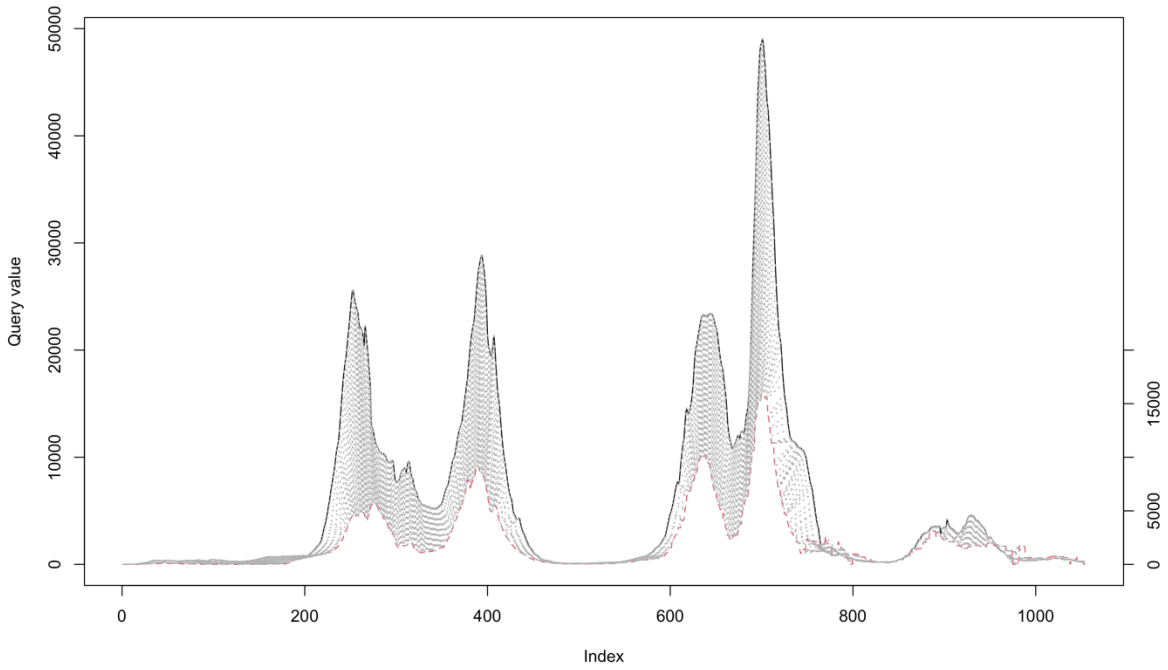


Рис.15

По даним dtw видно, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 2556869, а нормована відстань – 1214. З рисунку, кривої трансформації та з відстаней видно, що графіки Польщі та Угорщини відрізняються дуже суттєво. В Польщі суттєво переважає захворюваність на covid-19. На кривій трансформації добре видно стрибки, особливо ближче до лівого нижнього кутка, де велика відстань між точками графіків (місця, де графіки дуже відрізняються).

Крива трансформації для часових рядів Угорщини та Румунії, рисунок 16:

```
> t9 <- dtw(timese3, timese5, keep=TRUE)
> plot(t9,type="threeway")
```

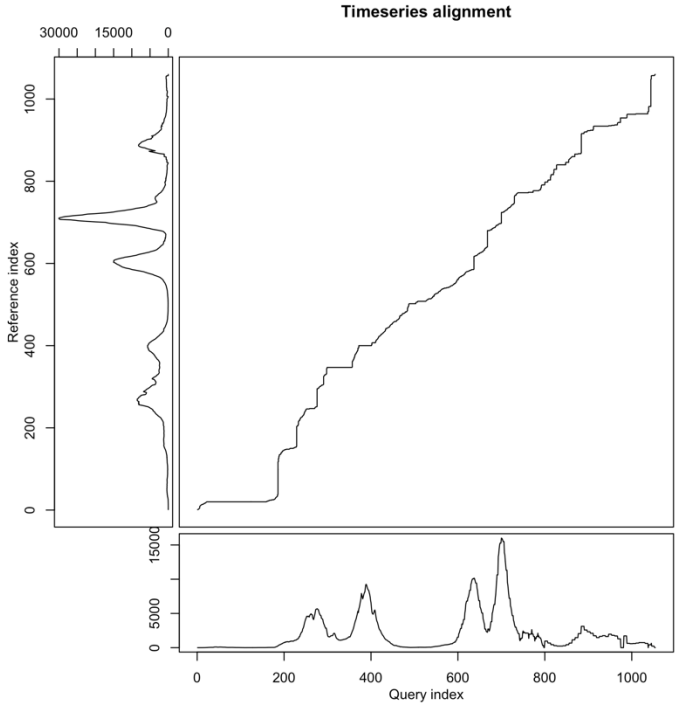


Рис.16

Відповідний графік twoway dtw, рисунок 17:

```
> plot(dtw(timese3,timese5,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

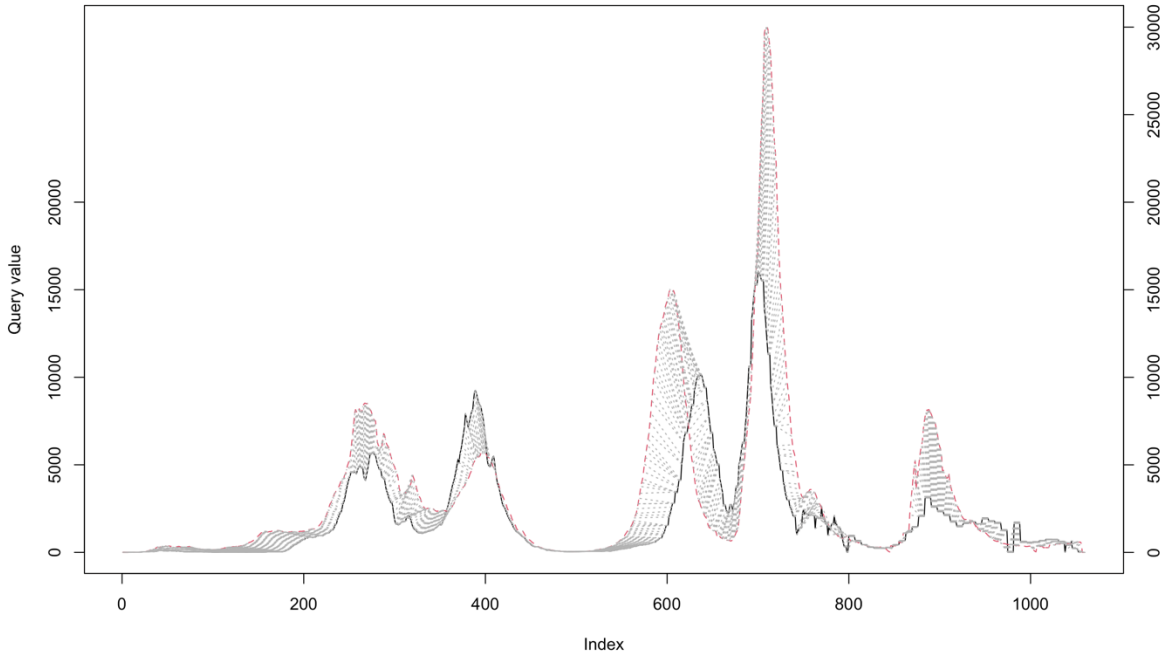
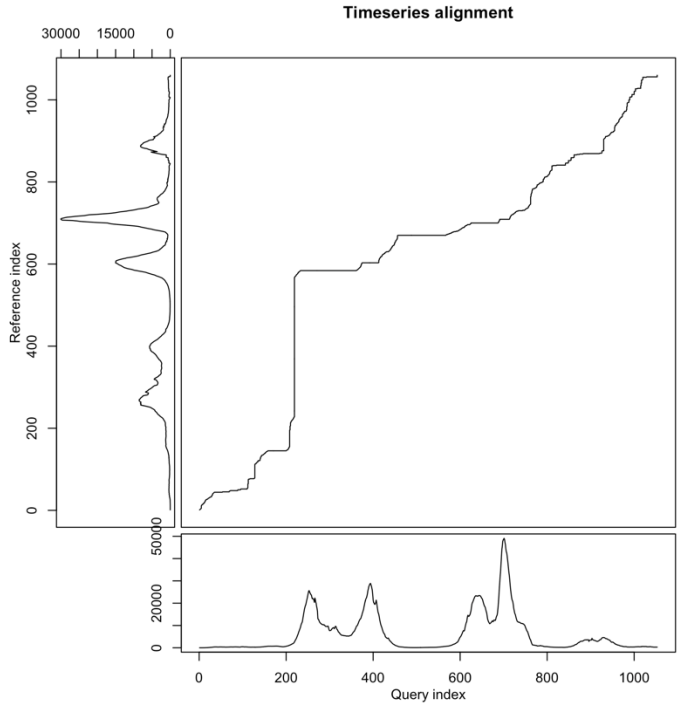


Рис.17

По даним dtw видно, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 1023122, а нормована відстань – 379. З рисунку, кривої трансформації та з відстаней видно, що графіки Румунії та Угорщини відрізняються не дуже суттєво. Крива трансформації більш менш без стрибків, якщо порівнювати з минулими кривими, це пов’язано з тим, що нормована відстань – 379.

Крива трансформації для часових рядів Польщі та Румунії, рисунок 18:

```
> t10 <- dtw(timese4, timese5, keep=TRUE)
> plot(t10,type="threeway")
```



Відповідний графік twoway dtw, рисунок 19:

```
> plot(dtw(timese4,timese5,keep=TRUE,
+ step=rabinerJuangStepPattern(6,"c")),
+ type="twoway",offset=-2);
```

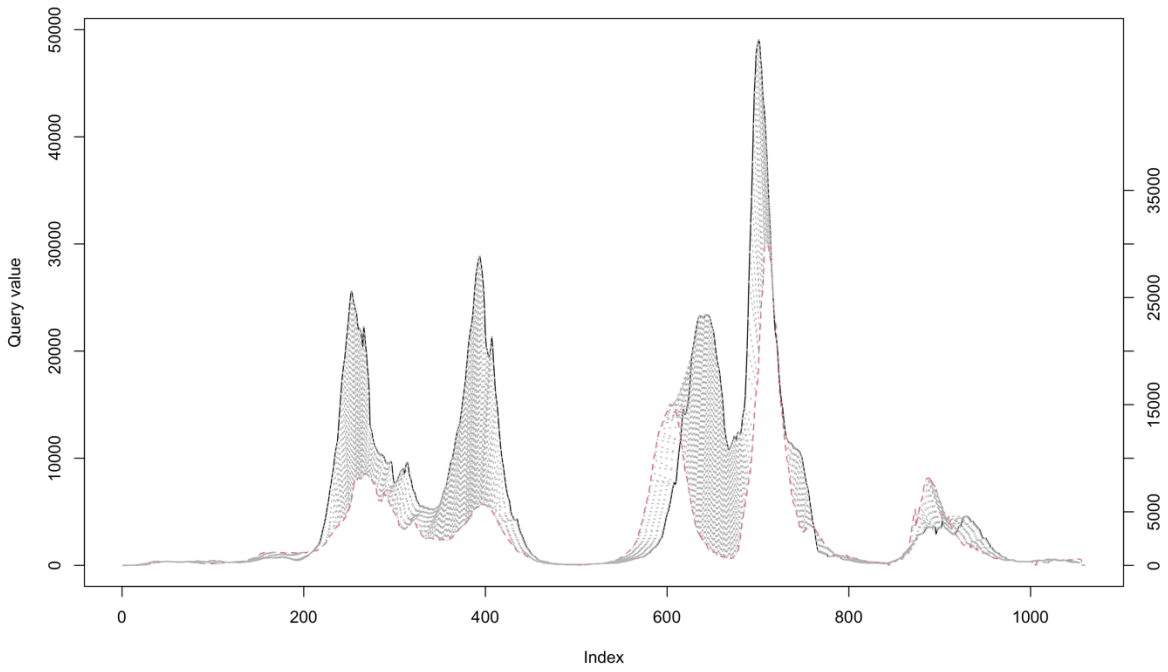


Рис.19

По даним dtw видно, що загальна відстань від точок одного графіку до відповідних точок іншого дорівнює 2301613, а нормована відстань – 1089. З рисунку, кривої трансформації та з відстаней видно, що графіки Польщі та Румунії відрізняються дуже суттєво. В Польщі суттєво переважає захворюваність на covid-19, але в кінці часового ряду в Румунії захворюваність була трішки вищою ніж в Польщі. Тут на кривій трансформації добре видно перший стрибок, де найбільша відстань між точками (місця, де графіки дуже відрізняються).

Висновки

В даній роботі можна побачити, що алгоритми динамічної трансформації часу можна широко використовувати не тільки в машинному навчанні для розпізнавання мови, а й в статистиці для аналізування даних. Використовуючи dtw, я проаналізувала захворюваність на covid-19, подивилася наскільки сильно вона відрізнялась в усіх країнах одна від іншої і зробила висновок: найбільше випадків зараження covid-19 було в Німеччині. Далі порівнюючи графіки, на другому місці опинилась Австрія. Далі йшла Польща, потім Румунія. І найменша, в порівнянні з іншими країнами, захворюваність була в Угорщині. Якщо порівнювати площі та населення країн, то не дивно, що Німеччина зайняла перше місце – вона серед цих країн найбільша та з найбільшим населенням. Австрія має значно меншу площу та населення, найменше серед усіх країн. Але випадків зараження там було настільки багато, що навіть серед країн з більшим населенням, там хворіли найбільше, якщо порівнювати з Польщею, Румунією та Угорщиною. В цих трьох країнах місця в рейтингу захворюваності залежать напряму від кількості населення. В Польщі кількість найбільша, в Румунії майже вдвічі менше населення, і в Угорщині ще вдвічі менше.
Загалом, можна сказати, що завдяки алгоритму динамічної трансформації часу, аналіз даних відбувається точніше та швидше. Dtw дозволяє аналізувати дані якісніше.

Список першоджерел

Covid-19 statistics. *Ourworldindata*. URL : https://ourworldindata.org/explorers/coronavirus-data-explorer?time=earliest..2022-07-01&facet=none&uniformYAxis=0&Metric=Confirmed+cases&Interval=7-day+rolling+average&Relative+to+Population=false&Color+by+test+positivity=false&country=AUT~DEU~HUN~POL~ROU (дата звернення 22.01.23).

Література, яку я використовувала для теоретичної роботи з бібліотекою dtw:

Al-Naymat, G., S. Chawla, and J. Taheri. *SparseDTW: A Novel Approach to Speed up Dynamic Time Warping*. Melbourne : The 2009 Australasian Data Mining, ACM Digital Library, 2009.

Eamonn J. Keogh, Michael J. Pazzani. *Derivative Dynamic Time Warping*, Section 1. California : Department of Information and Computer Science University of California, 2001.

Pavel Senin. *Dynamic Time Warping Algorithm Review*. Honolulu : Information and Computer Science Department, University of Hawaii at Manoa, 2008.

Stan Salvador and Philip Chan. *Fast DTW: Toward Accurate Dynamic Time Warping in Linear Time and Space*. Melbourne, FL : Dept. of Computer Sciences, Florida Institute of Technology, 2007.

Література, яку я використовувала для практичної роботи з бібліотекою dtw:

Toni Giorgino. *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.* 2009. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

Lawrence Rabiner, Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall; United States Ed edition, 1993.

How to Plot Multiple Plots on Same Graph in R. *Statology*. URL : **https://www.statology.org/r-multiple-plots-on-same-graph/** .