Visualizing COVID19 data from different countries

732A76 Research Project
Linköping University
2020

Martin Benes

Abstract

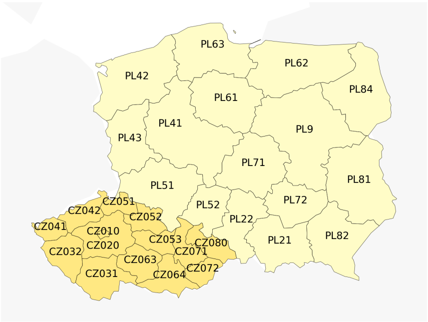
The aim of the project is to develop software that downloads COVID19, mortality and population data and then to provide visualizations comparing different countries through the similarity of their regions. First one identifies regions that are most similar (e.g. similar population size and/or density). Then one aligns the time-series curves of their statistics and scores how much they differ.
Based on the differences between regions one can cluster, seriate the countries. There are multiple ways of clustering, visualizing such similarities (e.g. classical k-means or Czekanowski's diagrams). The implemented software's visualizations should be both general comparison of countries and also detailed regional comparisons (between multiple countries).

At the moment countries are being compared between each other through their COVID19 related statistics: *confirmed cases*, *hospitalizations*, *case fatalities*, *excess* and *total mortality*, sometimes per capita.
However, this might not be the most informative way. This is as the dynamics of the virus are regional and different countries will have different population structures (e.g. UK: many large cities, Sweden: a huge amount of the population in Stockholm). Therefore, comparing country-wide statistics could be misleading.

Data

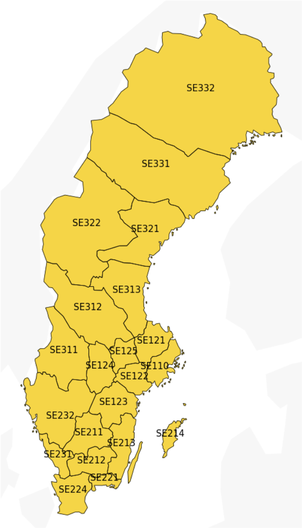
For the analysis various data were needed. A common regional identifiers NUTS[*] of Eurostat[†] could have been used. The current version is 2016.

Figure 1. Administrative division of Czechia and Poland.



NUTS geometries The borders, area polygons and centers of the NUTS regions are reachable from the Eurostat webpage. (*1*) The data files are transformed into cartographical visualizations in Figures 1 and 2 by a function `plot_map()` in file `src/cartography.py`.

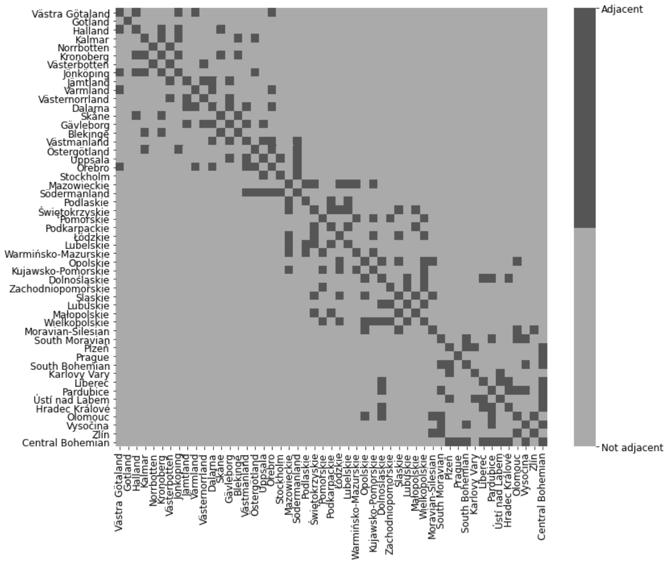
Figure 2. Administrative division of Sweden.



Viewing the map, one can write down the neighbors of each region into an adjacency matrix. The adjacency matrix as shown in Figure 3 is stored in file `data/adjacency.json`.

[*] Nomenclature of Territorial Units for Statistics
[†] European Statistical Office

Figure 3. Adjacency matrix.



Regional characteristics Population and area were read from several sources into `data/regions.csv`. (2) (3) (4) (5) The NUTS geometry centroids are put in the file as well.

Deaths Eurostat publishes regional deaths in a unified format for all the countries reporting it. As part of this paper, Python package *eurostat_deaths* was written to envelop the API and fetch the data. (6)

Covid-19 Deaths

Czechia An official source of Czechia, managed by MZ ČR[‡], although only in Czech language mutation, contains API with CSV/JSON machine-readable data files such as

- `orp`: incidence and prevalence (all, 65+, 75+), deaths, recovered, tests and newly hospitalized per LAU2
- `osoby`: confirmed cases (date, age, sex, LAU1)
- `umrti`: deceased cases (date, age, sex, LAU1)

Most data files cover the whole epidemy since the early March, the epidemy in Czechia. Some newly added datasets start later. (7) As part of this paper, a Python package *covid19czechia* was created to envelope the API. (8)

```
import covid19czechia as CZ
x = CZ.covid_deaths()
```

Listing 1: covid19czechia usage example

Sweden In Sweden Folkhälsomyndigheten[§] is responsible for publishing the official website for Covid-19. The data are summarized in XLSX format containing

- daily confirmed per region
- daily total deaths and ICU[¶] occupancy
- weekly deaths and ICU per region

To point out, the data of the interest for this paper, deaths per region, are accessible only per week and not per day.(9)

The data are accessible through a Python envelope *covid19sweden* developed as part of this paper. (10)

```
import covid19sweden as SE
x = SE.covid_deaths()
```

Listing 2: covid19sweden usage example

Poland For a long time the Polish responsible authority MZ[‖] has been publishing the daily statistics only via their Twitter. Their website (11), contains only currently active cases per NUTS-2. Only at the end of November 2020, they added the daily incidence and deaths per NUTS2 and LAU1 in CSV format.

Currently the only source for data prior to 24th November is the Twitter feed of MZ. (12) Information published there are

- confirmed cases per hospital (LAU1)
- deaths per NUTS2 (incl. sex and age)
- total tests performed (rounded to hundreds)

Some other information is published as image summary (hospitalizations, precise test counts). The format very impractical and hard to process by a software. A library *covid19poland* has been developed to automatically scrape the tweets and convert them into structured csv. (13)

Since the tweets with the statistics are formulated similarly, the chosen method is simple regex matching. Github workflow run the scraper every day and publishes the result of the scraping into a file in the repository, the package can fetch it to return daily updated data

```
import covid19poland as PL
x = PL.covid_deaths()
```

Listing 3: covid19poland usage example

The figure 4 shows the age distribution per gender and country. The deaths are mostly elderly people older than 60 years old, comparison of gender distributions are slightly unequal - this could be caused by a sociological or behavioral phenomena, distribution of the genders in population or other factor.

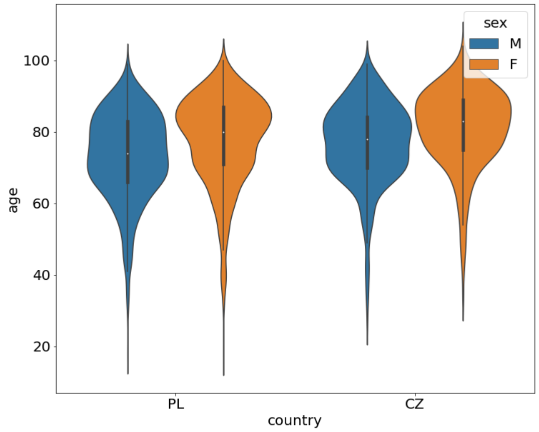
[‡] Ministry of Health of the Czech Republic

[§] Swedish Public Health Agency

[¶] Intensive care unit

[‖] Ministry of Health of Poland

Figure 4. Age distribution per sex.



Method

The choice of the units for comparison is given by what regions are used in Covid-19 statistics for the countries. Countries follow (with minor exceptions) areal granularity of NUTS regional codes from Eurostat. (*14*)

However comparing the administrative units is not easy. While Sweden publishes the weekly deaths per NUTS3 (*län*) (*9*) and Czechia daily deaths per NUTS3 (kraje) and LAU1 (*okresy*) (*7*), Polish authorities report the daily deaths per NUTS2 (*województwa*), and in addition data from Warsaw are published only for NUTS1, PL9 *Mazowieckie*.

Regional comparison

Before looking at actual *SARS-CoV-2* numbers we might have to analyze the administrative division of the data itself, the regions of different countries and their similarities, based on such information as

- adjacency, geographic distance
- population, area, density

In terms of location, similarity can be measured as geographic distance or common neighbors. Location explains similar data as mobility between close regions and hence interchange of infection. Other characteristic such as population density might be responsible for the infection pace within a single region.

Distance Distance is the most obvious metric of location similarity. Regions (areas) are represented by a single geographic point and their physical distance $d_{GC}(x_1, x_2)$ makes a reasonable and easily interpretable metric.

The region is represented with its centroid. These data are contained in the Eurostat's NUTS metadata. (*1*) The centroid distance is only one of the possibilities how to construct a single-point construction of complex objects. (*15*)

To emphasize the closest regions only and drop the unnecessary gradient for distant regions, distance is mapped by a RBF kernel.

$$kd(x_1, x_2) = \exp\Big(-\frac{d_{GC}(x_1, x_2)^2}{2h^2}\Big) \tag{1}$$

Adjacency Adjacency of a region can be utilized for seeking similarities between regions. An option is a simple adjacency matrix, but just the direct region adjacency does not necessarily imply similarity, especially if their border is very short (e.g. *PL41-PL52*, *CZ031-CZ064*). It can be although expanded according to the hypothesis that similar regions share a lot of common neighbors.

$$d(x, y) = 1 - \frac{\big|neighbors(x) \cap neighbors(y)\big|}{\big|neighbors(x) \cup neighbors(y)\big|} \tag{2}$$

Location It is possible to combine both metrics, getting the adjacency weighted by spacial relevance. If at least one is scaled to $[0; 1]$, we can use a simple product. To make the result match only the closest as dependent, the score is *non-linearly* projected using a square root.

$$d(x, y) = \sqrt{kd(\overline{x}, \overline{y}) \cdot d_{adj}(x, y)} \tag{3}$$

Population Analysis of the population in various regions can also bring valuable insight into the underlying connections of the regional disease spread. The data of interest, collected by local statistical authorities are population counts, region area and the population density.

The latter, roughly understood as normalized population, could be in particular connected to the contagion pace since more people in the area means more critical contacts and higher risk of virus spread. However the relation might not be necessarily linear. The average population density is

$$\text{region density} = \frac{\text{region population}}{\text{region area}} \tag{4}$$

Covid-19 comparison

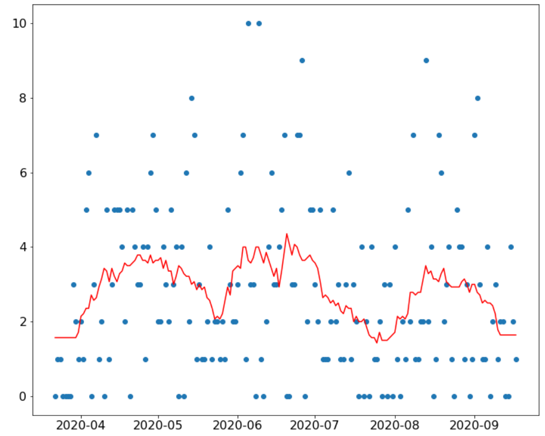
The deaths reported for Covid-19 for each region are normalized per 1000 inhabitants. To reduce the noise, deaths are filtered by a K-nearest neighbor regression to receive the trend only, as shown in figure 5.

Dynamic Time Warping Assumption that the regions make sense to be compared due to some underlying factors is legitimate, however temporal lags between the regional trends make it impossible to use the direct comparison (e.g. correlation or euclidean distance). One way to do this is *a dynamic time warping* (DTW). (*16*)

DTW matches the input vectors to find the past timeslot coupling. Once the coupling is done, any distance metric can be applied to get the distance score. All the region couples as two-way plots can be found in the appendix.

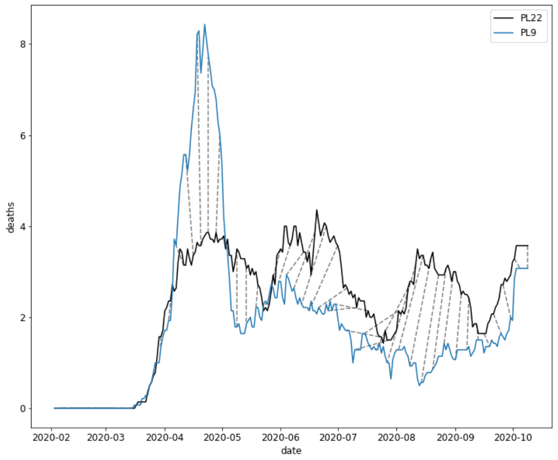
An example of method is `dtw_PL22_PL9.png`, shown in the figure 6. The restrictions in Poland such as as closing schools and cultural events were being gradually introduced

Figure 5. Deaths of PL22 with kNN filtered trend.



during March and April, (17) (18) from May the restrictions were put down. 28^{th} June and 12^{th} July there were two rounds of presidential elections. The schools were reopened after the summer (19) and closed again in October when the situation got more serious, (20) (21) eventually nationwide lockdown was imposed at the end of December. (22)

Figure 6. DTW of PL9 and PL22.



The drop at the end of May most likely corresponds with the restrictions introduced during March and April, their loosening explains increase in the June. Presidential elections don't seem to have a visible influence on the death statistics.

In the summer, the numbers were in particular low, probably due to people on vacation and hence less social interactions among people. In August and September there is a peak in PL9, that is hard to explain with regard to restrictions. The school start launches exponential growth, that explodes in October.

Czekanowski diagram One of the generally lesser known charts used for multivariate data visualization is *Czekanowski diagram*, an example is shown in the figure 16. The method of graphical mapping is fairly similar to heatmaps: a grid of rows and columns with similarities (or distances) between pairs of variables. Both methods define a metric to transform a pair of N-sampled vectors to a single scalar. (23)

The main difference is what graphical feature these scalars are mapped onto. For heatmap it is color, Czekanowski diagram on the other hand uses round circles with similarity proportional to the circle size.

Behind the dominance of heatmaps there are two possible factors: the color perception channel is more than twice as wide as marker size. a good color-mapping enables a healthy human eye to spot more distinct object classes. (24) As second, circles in the Czekanowski diagram often overlap and create a single continuous mass unless lower α-level is used. This behaviour is not present in heatmaps because their grid of squares change only the color fills.

Both mentioned sidelines Czekanowski diagram for general usage, but in certain cases it might still be appropriate. The *mass merging* if exploited using nonlinear kernel and thresholding favors cluster exploration. The good hyperparameter configuration (kernel h, cutoff) is essential; the results of the analysis change dramatically due to hard borders of the observed clusters.

Several projects, e.g. program MaCzek (25) or R package RMaCzek (26) have been done, but the bottleneck for popularity is a missing implementation for other popular data science languages such as Python, Scala, etc.

As heatmap, Czekanowski diagram needs to be seriated in order to present comprehensible patterns as well. In the optimal arrangement non-zero objects lay close to the diagonal, in particular the high-scored ones. The objective function should thus look at both the distance from diagonal and the score. (23) suggests

$$U_m = \frac{2}{n^2} \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} \frac{(i-j)^2}{W_{ij}+1} \tag{5}$$

It is not that crucial to find the the global maximum if the inference results in the same conclusions.

Experiments

Administrative divisions

The choice of the administrative division is important and it will be analyzed prior the Covid-19 deaths analysis itself.

The dataset consists of three countries (CZ, PL, SE), the summary of the their regions is shown in the table 7.

Tests A two-tailed t-test of means equality between divisions of countries will be used. Test hypotheses are

Figure 7. Summary of the regional divisions.

Statistics		Countries		
		Czechia	*Sweden*	*Poland*
	N	14	21	16
Population	μ	753845	491790	2402327
	σ	343842	587474	1266901
Area	μ	5634	19394	19542
	σ	2759	22605	6836
Density	μ	297	51	129
	σ	651	78	76

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 \\ H_A &: \mu_1 \neq \mu_2 \end{aligned} \tag{6}$$

However due to strong assumption for the distribution of data, the reliability of the results can be questioned if the data are far from t distribution.

$$\begin{aligned} H_0 &: \text{Data} \sim t(\cdot) \\ H_A &: \text{Data} \nsim t(\cdot) \end{aligned} \tag{7}$$

The hypothesis 7 whether data follows the student t-distribution can be tested using a non-parameteric Kolmogorov-Smirnov test. This test is implemented by the function `hypotheses.is_t_distributed()`.

Figure 8. P-values for Kolmogorov-Smirnov test (eq. 7).

Country	Attributes		
	Population	*Area*	*Density*
Czechia	0.141	0.001	0.097
Sweden	0.116	0.009	0.083
Poland	0.001	0.001	0.129

According to the results in the table 8 the t-test results for *area* are not reliable, because it does not follow the t-distribution, the hypotheses are rejected for all three countries. Polish regions' *population* is also rejected.

The used two-sample test for the hypothesis 6 is conditional on variance equality: t-test or Welch's t-test, which does not assume equal population variance. (*27*) The variance equality is tested using Levene's test with hypotheses in the equation 8.

This two-sample test is implemented in `administrative_divisions_similar()` in the module `src/hypotheses.py`.

$$\begin{aligned} H_0 &: \sigma_1 = \sigma_2 \\ H_A &: \sigma_1 \neq \sigma_2 \end{aligned} \tag{8}$$

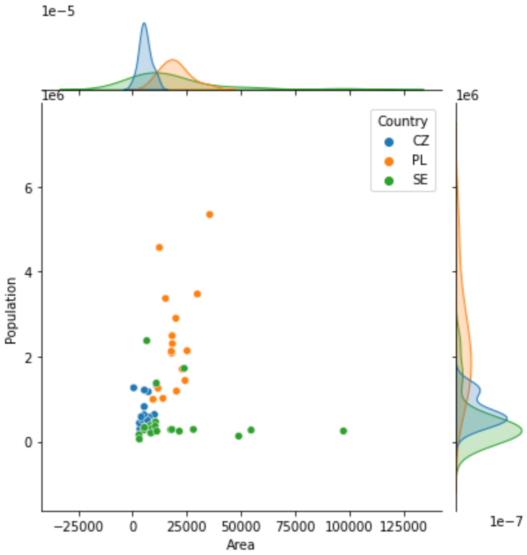
Figure 9. P-values for t-test test (eq. 6).

Country		Attributes		
		Population	*Area*	*Density*
Sweden	Poland	$1.82 \cdot 10^{-5}$	0.98	$4.2 \cdot 10^{-3}$
Sweden	Czechia	0.143	0.031	0.094
Poland	Czechia	$1.01 \cdot 10^{-4}$	$3 \cdot 10^{-7}$	0.314

Population density of Sweden and Poland have different are different between Sweden-Poland. Czechia and Sweden have similarly large regions in population. We can't rely on other results of the test in the figure 9.

Visual analysis Plots of the data helps to validate whether the results of the hypothesis testing correspond with the reality visualized in the plots.

Figure 10. Regional Area vs. Population.

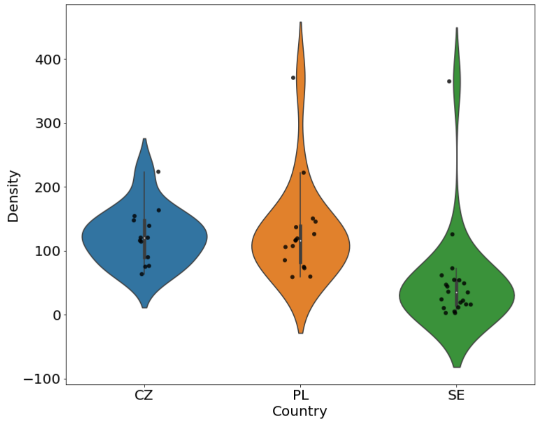


The scatterplot of regions in the figure 10 shows that the distributions in both population and area are skewed left for all three countries. Except of bimodal marginal density of population of Czech regions, all other marginals of all the countries are unimodal. Many regions of Sweden and Czechia overlap and are quite small in both area and population.

Polish regions varies in population, Swedish in area. Czech regions are very similar in both area and population. It is expected that in density Czech regions are all very similar and Polish and Swedish will have higher variance. Population of Czechia and Sweden seems to have similarly sized regions according to the plot and the test result.

According to the figure 11 Swedish regions are in general lower in density than the Czech and Polish, which are quite comparable. In the plot, CZ010 must have been removed,

Figure 11. Regional population densities (without CZ010).



because it's density 2554.826 is a way out the range of y axis. The region contains only the metropolitan area of the Prague without the surrounding agglomeration, which is in CZ020.

Outliers Outliers has been observed using *inter-quartile range* (IQR) method, several regions has been marked, results are presented in the figure 12.

Country	**Outliers**		
	Population	*Area*	*Density*
Sweden		SE322, SE331, SE332	SE110
Poland	PL9, PL22		PL22
Czechia			CZ010

Figure 12. IQR outliers.

Usually capitals have their own region which results in very high density and an outlier, as detected for Czechia and Sweden: CZ010 (*Prague*) and SE010 (*Stockholm*). In case of Poland, PL9 (*Mazowieckie*) contains a large land around the capital and hence spreads the population over the land with sparser settlement. On the other hand PL22 (*Ślaskie*) containing Upper Silesian metropolitan area was detected.

Regarding population, the only Polish regions PL9 and PL22 are outliers. Capitals of Czechia and Sweden have high density, but in population they are comparable to others in the country.

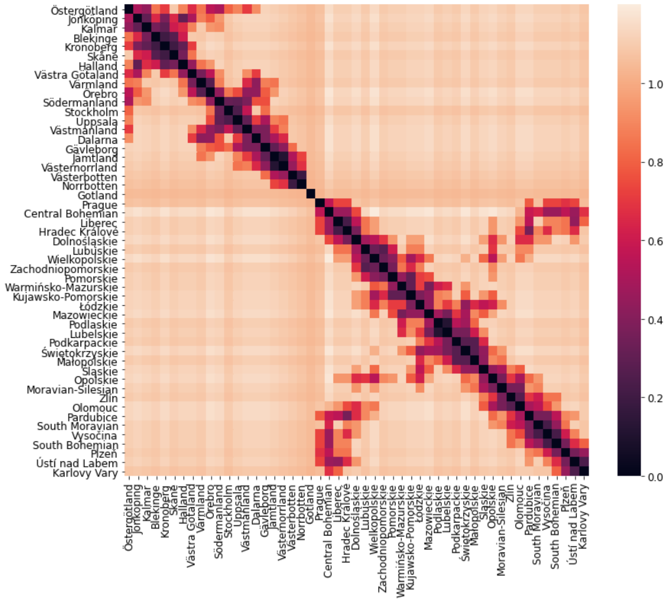
In area the outliers are SE322 (*Jämtland*), SE331 (*Västerbotten*) and SE332 (*Norrbotten*), sparsly populated large regions in the north.

Adjacency The figure 13 shows adjacency score distance matrix.

The only significant outlier is Gotland, a region on an island without any direct land borders.

There is a plenty of small, significantly overlapping clusters.

Figure 13. Adjacency score of regions.



- Östergötland, Jonköping, Kalmar, Blekinge, Kronoberg, Skåne, Halland (*Southern Sweden*)
- Örebro, Södermanland, Stockholm, Uppsala, Västmanland (*Stockholm*)
- Jämtland, Västernorrland, Västerbotten, Norrbotten (*Northern Sweden*)
- Prague, Central Bohemian, Liberec, Hradec Králové, Dolnoślaskie (*Bohemia*)
- Dolnoślaskie, Lubuskie, Wielkopolskie, Zachodniopomorskie (*Western Poland*)
- Łódzkie, Mazowieckie, Podlaskie (*Northern Poland*)
- Podkarpackie, Świetokrzyskie, Ślaskie (*Eastern Poland*)
- Ślaskie, Opolskie, Moravian-Silesian, Zlín, Olomouc (*Silesia*)
- Pardubice, South Moravian, Vysočina (*Moravia* + *Bohemia*)
- Plzeň, Ústí nad Labem, Karlovy Vary (*Bohemia*)

Several regions belong to multiple clusters: Dolnoślaskie, Ślaskie. However the clusters are very fuzzy borders.

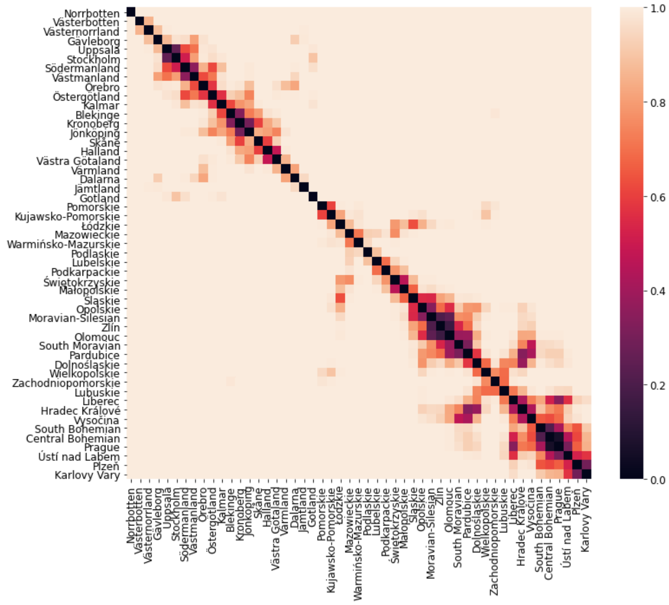
The closer regions have lower value. Far regions' score does not raise with greater distance between the regions - according to the metric, non-adjacent regions are considered just different.

The similarities of the Northern Sweden's regions, the big regions show the another disadvantage of the method - adjacent regions with only few neighbors overrate the score.

Distance The figure 14 shows the centroid distance matrix.

It is an important to realize prior to the analysis that the Polish regions are in average much greater in area than the Czech which necessarily implies the distances of their centroids will be larger as well. (2) (3)

Figure 14. Distances of region centroids.



The radial base kernel has width $h = 100$. This number has an utter impact on the result of the analysis, greater value of h will make more regions being more similar. The value 100 (about 100 km) makes sense from the point of view of epidemy spread.

The outliers are regions of Northern Sweden due to their great size and Gotland again. Many regions do not construct any visible clusters or do not have any other very similar regions. However several clusters are present such as

- Uppsala, Stockholm, Södermanland, Västmanland
- Blekinge, Kronoberg, Jönköping
- Opolskie, Moravian-Silesian, Zlín, Olomouc, South Moravian, Pardubice
- South Bohemian, Central Bohemian, Prague, Ústí nad Labem
- Plzeň, Karlovy Vary

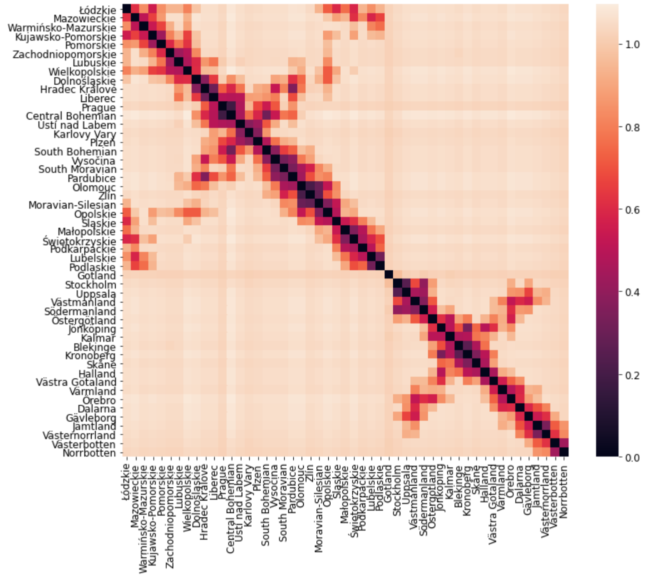
Location Figure 15 shows the location score, combination of the adjacency and distance score.

The new metric brought a gradient to the matrix along the diagonal scores. There are no new clusters except the ones already observed.

Czekanowski diagram

To plot a Czekanowski diagram, one starts by defining a similarity metric and constructing a similarity matrix. The second step is to find a single order rearrangement for both rows and columns of the similarity matrix such that maximizes the objective function defined by equation 5.

Figure 15. Combined location scores of regions



Genetic algorithm The optimization is implemented by a genetic algorithm. Whole population is randomly paired, each couple produces two children by a single point crossover (alternative for permutation). Child undergoes mutation (with given probability) by swapping $1-3$ randomly chosen positions. The new generation is created by a *war* procedure, choosing N best individuals from all parents and children (original and mutated).

Algorithm usually converges after ~ 300 iterations, stopped by early stopping measuring a small difference ε of current best score and average of best scores of k last generations.

The genetic algorithm is implemented by the function `_czekanowski_ga_seriate()` in module `src/_czekanowski.py` and its basic blocks in the module `src/_ga.py`. The code 4 illustrates the idea.

```
# fitness of each chromosome
fitness = _ga.population_score(pop,obj)

# crossover
parents,pscore  = _ga.select_parents(pop,fitness,
                                       Nparents)
children,cscore = _ga.crossover(parents,obj)
# create mutants
mutants,mscore = _ga.mutate(children,obj,mutprob)

# war
pop = _ga.war(popsize,(parents,pscore),
              (children,cscore),(mutants,mscore))
```

Listing 4: Genetic algorithm

The method for the region comparison can be described by a code chunk, briefly taken from `czekanowski_dtw()` in the module `src/_covid.py`. The listing 5 shows how Czekanowski plot is constructed.

```
# distance matrix (metric dtw)
D = _covid.dtw_distance(data = data)
# rbf kernel
D = _czekanowski.distance_rbf(D)
# column permutation
P = _czekanowski.plot(D, cols = columns)

# Czekanowski diagram
import matplotlib.pyplot as plt
plt.scatter(P.x, P.y, s=P.Distance); plt.show()
```

Listing 5: Czekanowski DTW method

Regions comparison

Upsampling Because the deaths data published by Folkhälsomyndigheten are only weekly and not daily as the data for Czechia and Poland, (9) a certain postprocessing must be done in order to work with all the data as a single data frame.

Let's denote the known weekly death counts w, day of a week $d \in \{1, \dots, 7\}$ and the wanted daily death counts w_d. If an independence of week day and the death happening, an equal probability for all the days of the week, the upsampling distribution for w_i is defined according to the equation 9.

$$w_i \sim \text{Multinomial}(n = w, \pi_i = \frac{1}{7}), i = 1, \dots, 7 \quad (9)$$

DTW A module *dtw-python* was utilized for the dynamic time warping method. (28) The euclidean distance is used as the inner metric.

The filtering of the deaths timeseries is done with kNN regression, implemented by `KNeighborsRegressor` from `sklearn` package. The parameter *n_neighbors* is set to 14 (±1 week window). (29)

The results for the data for dates from 3rd February to 17th September (all available data), the parameter $h = 0.008$ is shown as Czekanowski diagram in the figure 16.

Visualizing all the regions, only Polish and Czech are significant in the parameter configuration. Similarity between progress of the contagion of Polish and Czech regions is much greater that of Swedish regions, and even greater than within Swedish regions, however the radial-based kernel exponentially transforms the sizes, so the chart perception might be biased.

The visible outliers of Poland and Czechia are Vysočina, Prague, Łódzkie and Karlovy Vary. The clusters detected are

- Central Bohemian, South Bohemian, Hradec Králové, Liberec, South Moravian, Zachodniopomorskie, Pomorskie, Kujawsko-Pomorskie
- Warmińsko-Mazurskie, Zlín, Lubelskie
- Ústí nad Labem, Świetokrzyskie, Lubuskie, Plzeň
- Podlaskie, Podkarpackie
- Mazowieckie, Opolskie, Dolnoślaskie

Figure 16. Czekanowski diagram of regional deaths; parameters $h = 0.008$, $c = 300$.

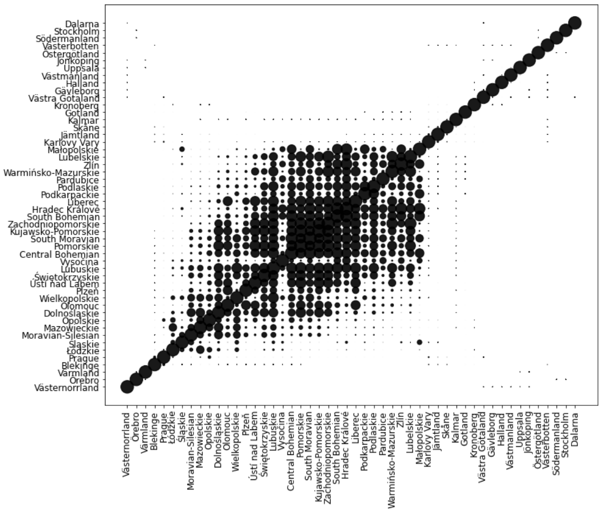
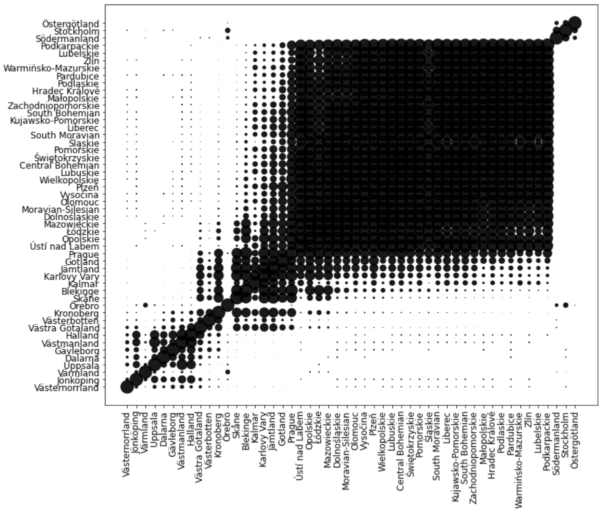


Figure 17. Czekanowski diagram of regional deaths; parameters $h = 0.035$, $c = 250$.



To focus chart to the Swedish regions, the value of the kernel width parameter is changed to $h = 0.035$. The figure 17 reveals two outliers: Örebro and Värmland. Two main clusters are

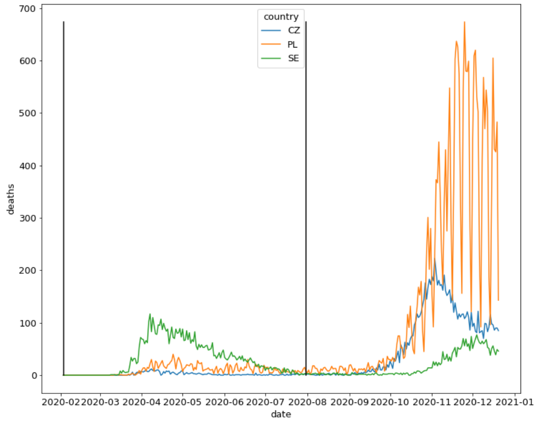
- Uppsala, Dalarna, Gävleborg, Västmanland, Halland
- Skåne, Blekinge, Kalmar, Karlovy Vary, Jämtland, Gotland, Prague

Many discovered clusters seem to have an geographical reasoning. Several clusters contain unexpected and unexplainable regions, e.g. Karlovy Vary and Prague among regions of Southern Sweden or mixed cluster of Northern Polish and Northern Bohemian regions.

First wave During the Covid-19 pandemics, media and scientists often use the term *second wave*. (*30*) The epidemics in general have several phases called waves as a rise and fall in the statistics. (*31*)

The first wave of Covid-19 pandemics can be detected just by observing the death counts traceplot per country.

Figure 18. Covid-19 daily deaths per country.



There is no hard border between the waves. In Sweden deaths visibly descends at the end of July, and start raising in Czechia with 1st September, the second wave starts during August. Numbers start raising again during September and October in Poland and Sweden as well as other countries.

Poland had a problem with regional epicenter in Upper Silesia. (*33*) This is most likely the reason why similar flat region is not that apparent in the Polish data.

From observed and referenced information, the first wave is considered between the first reported case and the end of July 2020, from 3rd February to 31st July 2020.

In the figure 19 the data form several clusters.

- Mazowieckie, Opolskie, Wielkopolskie
- Wielkopolskie, Świetokrzyskie, Olomouc, Dolnoślaskie
- Pomorskie, South Moravian, Central Bohemian, Ústí nad Labem, Plzeň, Kujawsko-Pomorskie
- Kujawsko-Pomorskie, Vysočina, Pardubice, Zachodniopomorskie, Podkarpackie, Liberec, South Bohemian, Podlaskie, Małopolskie, Lubelskie, Hradec Králové, Zlín, Warmińsko-Mazurskie

Another plot with different value for $h = 0.035$ is produced to discover more clusters.

Additional clusters appeared in the figure 20

- Gotland, Jämtland, Kalmar, Karlovy Vary
- Karlovy Vary, Blekinge, Skåne, Kronoberg, Västra Götaland
- Gävleborg, Halland, Dalarna, Jönköping, Västmanland, Uppsala

Figure 19. Czekanowski diagram of first-wave regional deaths; parameters $h = 0.008$, $c = 250$.

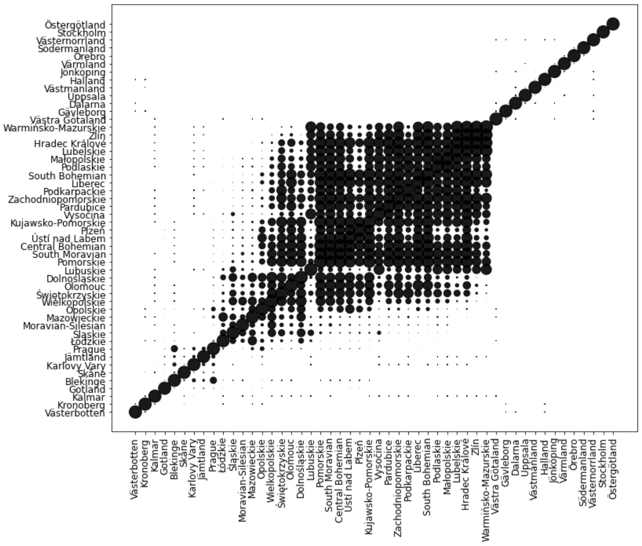
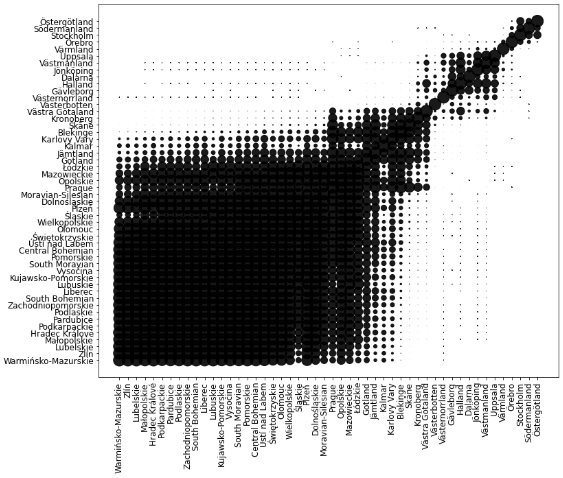


Figure 20. Czekanowski diagram of first-wave regional deaths; parameters $h = 0.035$, $c = 300$.



- Stockholm, Södermanland, Östergötland

Most of the clusters discovered in the figures 19 and 20 from the data of first wave correspond with the clusters observed on the whole data.

Weekday distribution Assumption of independence between week day and the deaths can be tested with hypotheses 10 against the data.

$$\begin{aligned} H_0 &: \mu_i = \frac{1}{7} \\ H_A &: \mu_i \neq \frac{1}{7} \end{aligned} \tag{10}$$

Surprising results of the test in the table 21 are acquired for Poland, that have significantly different ratio on Sundays

Figure 21. P-values for equal ratio t-test (eq. 10).

Day	**Country** *Czechia*	 *Poland*	 *Sweden*
Monday	0.581	0.001	0.429
Tuesday	0.496	0.06	0.088
Wednesday	0.784	0.112	0.731
Thursday	0.375	0.181	0.924
Friday	0.298	0.764	0.507
Saturday	0.112	0.737	0.394
Sunday	0.294	0.044	0.947

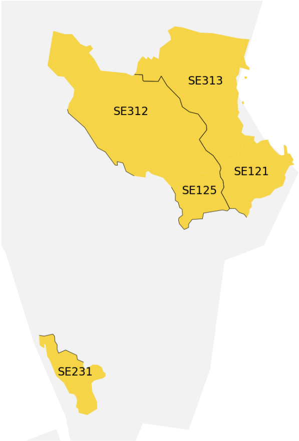
and mainly Mondays. This phenomenon is according to (*32*) caused by the diagnosis and reporting procedure.

In Czechia data there is no evidence for unequal distribution of death for different week days. Swedish distribution is artificially created by upsampling to be multinomial (equation 9).

Discussion

Results The results show similar characters of the deceased statistics in several groups of regions, explainable as a community transmission of the disease.

Figure 22. First cluster in Sweden.



Figures 22 and 23 show clusters with noisy regions - separated from the rest of the cluster and the similarity is not simple explain with different factor, connecting cluster and the separated region.

The figure 24 shows a single cluster, merged over two separated groups of regions. It could be possibly split into two.

Limitations The method is very sensitive on the value of kernel parameter h, the behavior is similar to hierarchical clustering. Additional goodness-of-fit score could be used to reduce the noisy regions and the merged regions shown in the results.

Figure 23. Second cluster in Sweden.

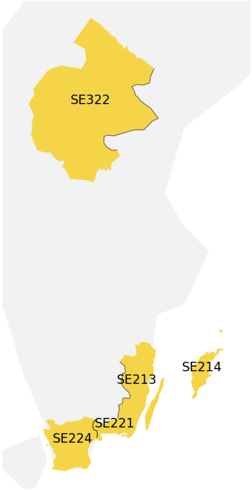
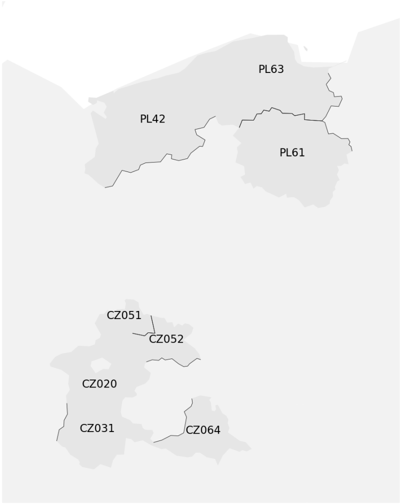


Figure 24. Joint cluster in Poland and Czechia.



Appendix

Appendix: DTW plots

File `dtw.zip` contains 1275 *PNG* files with plots of DTW results for all combinations of regions, as shown in figure 6.

- `dtw/`
  - `dtw_CZ010_CZ020.png`
  - `dtw_CZ010_CZ031.png`
  - `...`
  - `dtw_SE331_SE332.png`

Appendix: Code

All computations and plots were produced by the code in the Github repository https://github.com/martinbenes1996/732A76.

Acknowledgements

This class file was developed by Sunrise Setting Ltd, Brixham, Devon, UK.
Website: http://www.sunrise-setting.co.uk

References

1. Eurostat (2020). NUTS. https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/nuts
2. Regen Eco Sp. z o. o. (2015). Ranking województw. http://gminy.pl/Rank/W/Rank_W.html
3. MMP ČR (2019). Regionální informační servis. https://www.risy.cz/cs/krajske-ris/ustecky-kraj/kraj
4. SCB (2019). Folkmängd i riket, län och kommuner. https://www.scb.se/
5. SCB (2020). Land- och vattenareal efter region och arealtyp. https://www.scb.se/
6. Martin Beneš (2020). Eurostat deaths. https://pypi.org/project/eurostat-deaths/
7. MZČR (2020). Onemocnění aktuálně. https://onemocneni-aktualne.mzcr.cz/covid-19
8. Martin Beneš (2020). Web Scraper of COVID-19 data for Czechia. https://pypi.org/project/covid19czechia
9. Folkhälsomindigheten (2020). Bekräftade fall i Sverige. https://www.folkhalsomyndigheten.se/smittskydd-beredskap/utbrott/aktuella-utbrott/covid-19/statistik-och-analyser/bekraftade-fall-i-sverige/
10. Beneš M. (2020). Web Scraper of COVID-19 data for Sweden. https://pypi.org/project/covid19sweden
11. MZ (2020). Pliki archiwalne. https://www.gov.pl/web/koronawirus/pliki-archiwalne-wojewodztwa
12. MZ, Twitter (2020). Ministerstwo Zdrowia (@MZ_GOV_PL). https://twitter.com/MZ_GOV_PL
13. Beneš M. (2020). Web Scraper of COVID-19 data for Poland. https://pypi.org/project/covid19poland
14. Eurostat (2020). Regions and cities. ec.europa.eu/eurostat/web/nuts
15. Manning C., Raghavan P., Schütze H. (2008). Introduction to Information Retrieval, Cambridge University Press. ISBN: 0521865719.
16. Müller M. (2007). Dynamic time warping. Information Retrieval for Music and Motion. 2. 69-84. 10.1007/978-3-540-74048-3_4.
17. Fahey R. (11 March 2020). Poland closes all schools, museums and cinemas. https://www.dailymail.co.uk/news/article-8099501/Poland-closes-schools-museums-cinemas-amid-coronavir html
18. Reuters (9 April 2020). Poland's coronavirus restrictions extended towards election date. https://www.reuters.com/article/us-health-coronavirus-poland-restriction-idUSKCN21R1
19. Reuters (24 August 2020). Poland to reopen schools despite the new coronavirus record. https://www.reuters.com/article/us-health-coronavirus-poland-idUSKBN25K1MF
20. BBC (31 October 2020). Coronavirus: Hungary and Poland see record cases. https://www.bbc.com/news/world-europe-54760766
21. Martewicz M. (21 November 2020). Poland Opens Malls But Keeps Schools Closed on Wait for Vaccine. https://www.bloomberg.com/news/articles/2020-11-21/poland-opens-malls-but-keeps-schools-closed-on-wait-
22. Xinhua (17 December 2020). Poland to impose nationwide lockdown from Dec. 28 to Jan. 17. http://www.xinhuanet.com/english/2020-12/18/c_139598453.htm
23. Sołtysiak A., Jaskulski, P. (1999). Czekanowski's diagram. A method of multidimensional clustering.
24. Ward, M., Grinstein, G. and Keim, D. (2015). Interactive Data Visualization. 2nd ed. Boca Raton: CRC Press, pp.127 - 128.
25. Sołtysiak, A. (1997). UMCzek 1.00. Ewolucyjny algorytm porzadkowania diagramu Czekanowskiego. Biuletyn Antropologiczny. 1. 21-24.
26. Vasterlund A. (2019). Czekanowski's Diagram: Implementing and exploring Czekanowski's Diagram with different seriation methods. Master's thesis, Linkoping University, Sweden.
27. The SciPy community (2020). SciPy ttest_ind. https://docs.scipy.org/
28. T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. J. Stat. Soft., doi:10.18637/jss.v031.i07.
29. Scikit-learn developers (2020). Scikit KNeighborsRegressor. https://scikit-learn.org/
30. Catherine Edwards (2020). Is it right to say Sweden is still not experiencing a 'second wave' of Covid-19? https://www.thelocal.se/
31. Adam Kucharski (2020). The Rules of Contagion: Why Things Spread - and Why They Stop. ISBN 978-1-78816-019-3.
32. Kopeć J. (2020). COVID-19 nie lubi poniedziałków. https://biqdata.wyborcza.pl/
33. Monika Sieradzka (2020). Upper Silesia region becomes Poland's coronavirus epicenter. https://www.dw.com/