UNIVERSITÀ
DEGLI STUDI
FIRENZE

Data Mining and Organization – Covid–19

Armand Palla

Università degli Studi di Firenze

15 February 2022

UNIVERSITÀ DEGLI STUDI FIRENZE

Table of contents:

1. Understanding the data

- Civil Protection Department - Coronavirus emergency.
- The Italian Civil Protection Department receives daily data by the Italian Ministry of Health, analyzes them and updates the database.
- The database is freely accessible at github.com/pcm-dpc/COVID-19.
- This database contains data of integrated surveillance for the "Coronavirus disease 2019" in Italy.
- The additional value of these data relies on the real-time (daily update) integrated surveillance of COVID-19 in Italy and on their reliability due to their official source (Italian Civil Protection Department).

1. Understanding the data

- These data are useful because:
  1. they provide insight on the spread of SARS-CoV-2.
  2. to inform Italian and foreign citizens on the SARS-CoV-2 spread in Italy.
  3. to support organizations in the evaluation of the efficiency of current prevention and control measures.
  4. to support governments in the future prevention decisions

PROTEZIONE CIVILE
Presidenza del Consiglio dei Ministri
Dipartimento della Protezione Civile

1. Understanding the data

- The database consists of different folders such as: ‘aree’, ‘dati-andamentonazionale’, ‘dati-json’, ‘dati-province’, ‘dati-regioni’ ,‘schede-riepilogative etc, but we are using:

1. dpc-covid19-ita-andamento-nazionale.json
2. dpc-covid19-ita-regioni.json
3. dpc-covid19-ita-province.json

1. Understanding the data

1. dpc-covid19-ita-andamento-nazionale.json

The folder called 'dati-andamento-nazionale' contains data relating to the national trend of SARS-CoV-2 spread.

I have used **pandas.DataFrame** in order to elaborate with data in .json format.

Inside each file, data are structured in the 24 fields (one column per field).


```
[
  {
    "data":"2020-02-24T18:00:00",
    "stato":"ITA",
    "ricoverati_con_sintomi":101,
    "terapia_intensiva":26,
    "totale_ospedalizzati":127,
    "isolamento_domiciliare":94,
    "totale_positivi":221,
    "variazione_totale_positivi":0,
    "nuovi_positivi":221,
    "dimessi_guariti":1,
    "deceduti":7,
    "casi_da_sospetto_diagnostico":null,
    "casi_da_screening":null,
    "totale_casi":229,
    "tamponi":4324,
    "casi_testati":null,
    "note":null,
    "ingressi_terapia_intensiva":null,
    "note_test":null,
    "note_casi":null,
    "totale_positivi_test_molecolare":null,
    "totale_positivi_test_antigenico_rapido":null,
    "tamponi_test_molecolare":null,
    "tamponi_test_antigenico_rapido":null
  },
```

1. Understanding the data

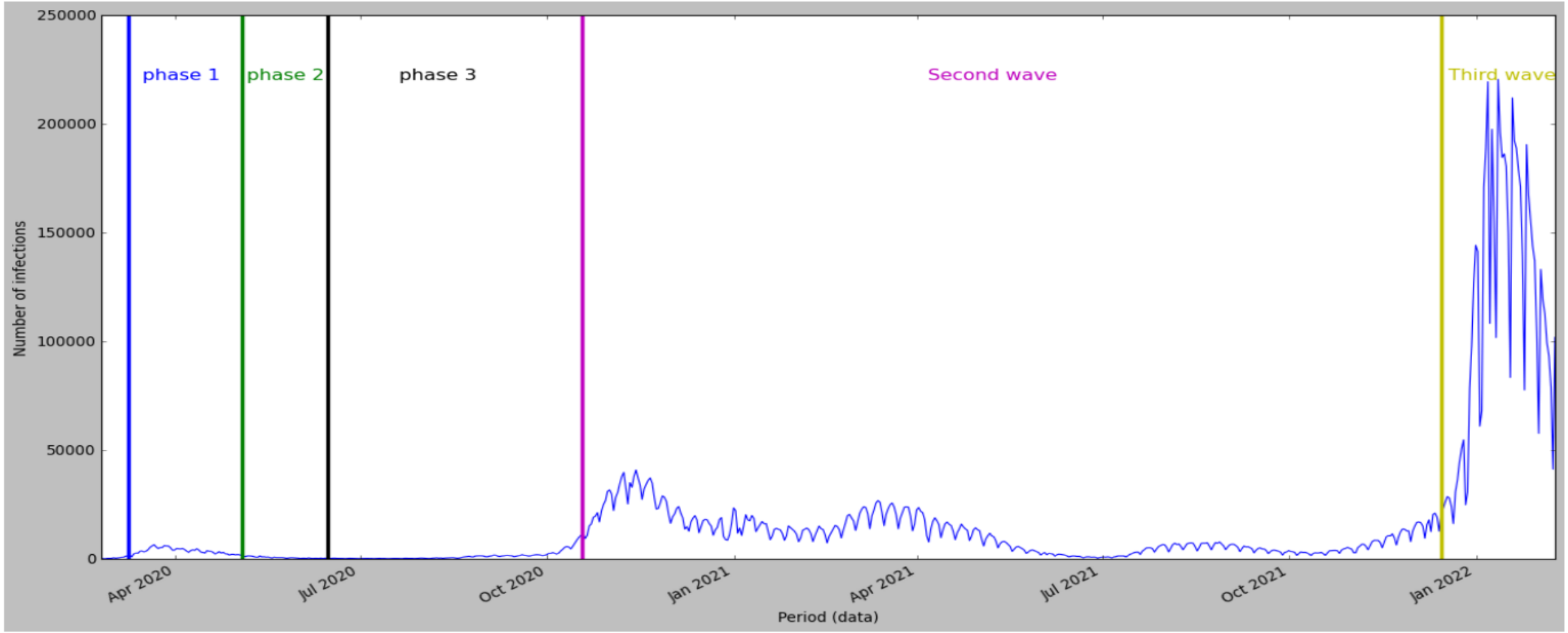
1. dpc-covid19-ita-andamento-nazionale.json

Then, we have done an analysis based on different **covid time intervals:**

- **Phase 1:** 09/03/2020 – 03/05/2020; Quarantine Period.
- **Phase 2:** 04/05/2020 - 14/06/2020 , Relaxation of containment measures.
- **Phase 3:** 15/06/2020 - 06/11/2020, living with COVID-19 .
- **Second wave:** 06/11/2020 - 14/12/2021, 06/11/2020-new DPCM that divides Italy into 3 zones(Yellow zone, Orange zone, Red zone).
- **Third wave:** 15/12/2021 - until now, the last period when we have seen an increase in positive cases.

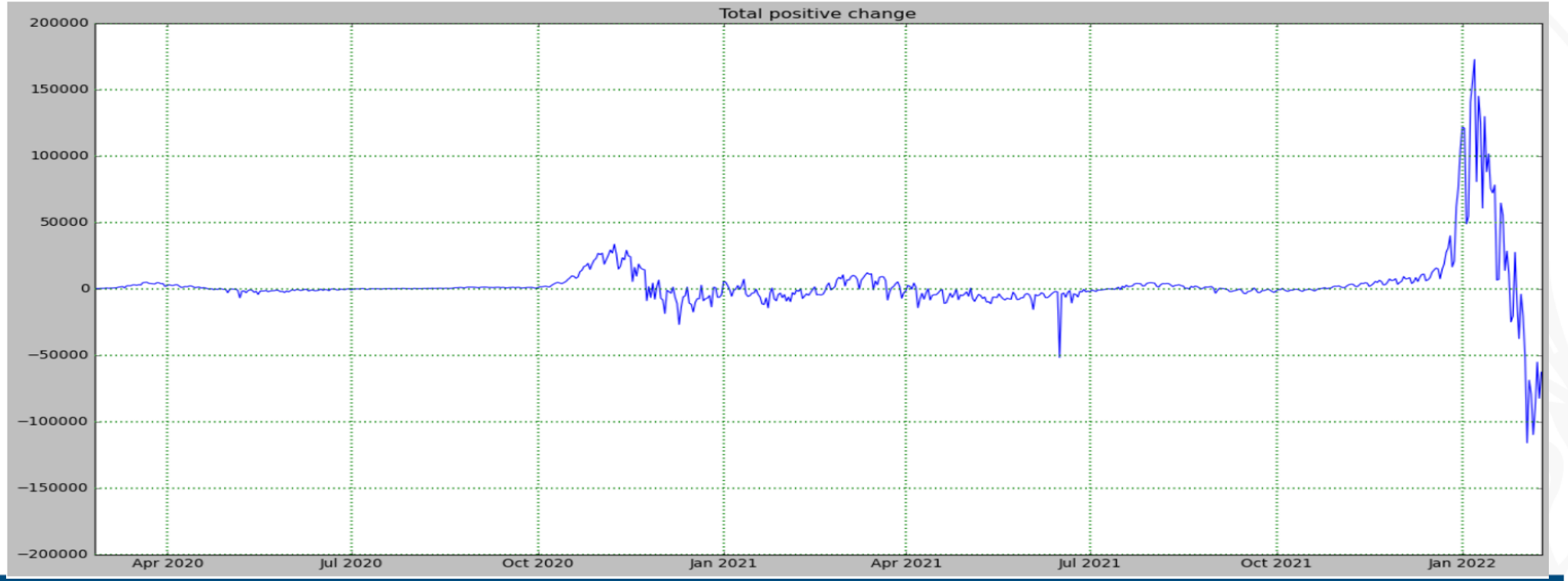
UNIVERSITÀ DEGLI STUDI FIRENZE

1. Understanding the data



1. Understanding the data

In another graph, we have represented the **total positives change** for every day from the initial until now and we can see that in the last week we have a decrease of positive cases.



2. Working with data

2. dpc-covid19-ita-regioni.json

The folder called ‘dati-regioni’ contains data relating to the regional trend of SARS-CoV-2 spread.

Inside each file, data are structured in the 24 fields (one column per field).


```
{
  "data":"2020-02-24T18:00:00",
  "stato":"ITA",
  "codice_regione":13,
  "denominazione_regione":"Abruzzo",
  "lat":42.35122196,
  "long":13.39843823,
  "ricoverati_con_sintomi":0,
  "terapia_intensiva":0,
  "totale_ospedalizzati":0,
  "isolamento_domiciliare":0,
  "totale_positivi":0,
  "variazione_totale_positivi":0,
  "nuovi_positivi":0,
  "dimessi_guariti":0,
  "deceduti":0,
  "casi_da_sospetto_diagnostico":null,
  "casi_da_screening":null,
  "totale_casi":0,
  "tamponi":5,
  "casi_testati":null,
  "note":null,
  "ingressi_terapia_intensiva":null,
  "note_test":null,
  "note_casi":null,
  "totale_positivi_test_molecolare":null,
  "totale_positivi_test_antigenico_rapido":null,
  "tamponi_test_molecolare":null,
  "tamponi_test_antigenico_rapido":null,
  "codice_nuts_1":null,
  "codice_nuts_2":null
},
```

2. Working with data

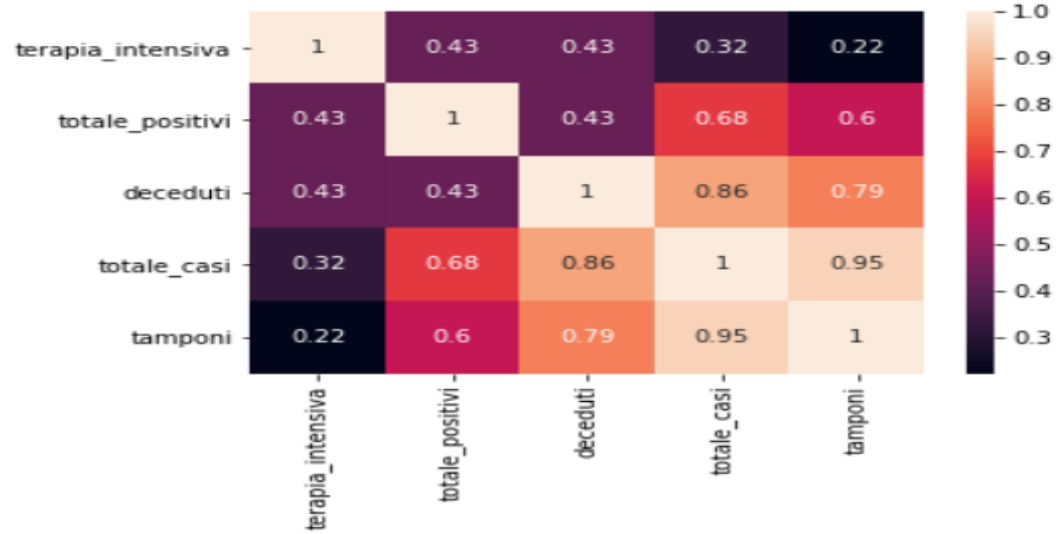
We have created another dictionary object called **covid**, in which we decided to add the fields that we think that are the most importants.

Then, we have plotted a heatmap using seaborn library

in order to represent the correlation between the choosen

variables.

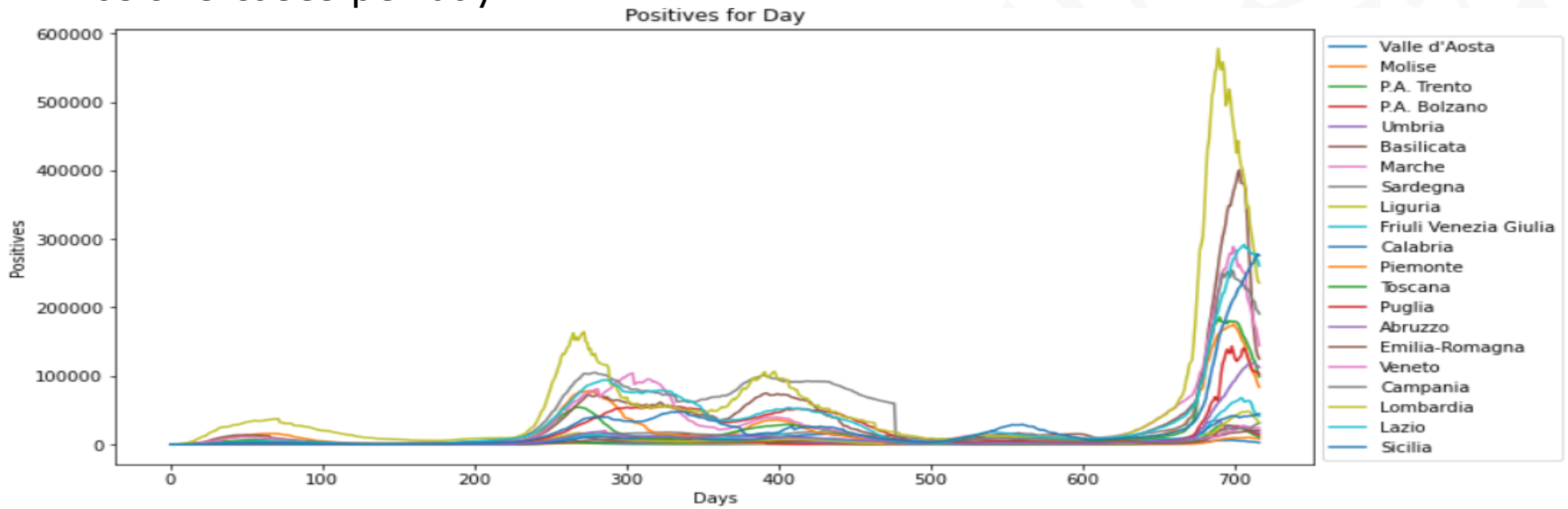
```
covid = {
    'date': json_data.data,
    'regioni': json_data.denominazione_regione,
    'terapia_intensiva': json_data.terapia_intensiva,
    'totale_positivi': json_data.totale_positivi,
    'deceduti': json_data.deceduti,
    'totale_casi': json_data.totale_casi,
    'tamponi': json_data.tamponi,
}
```



2. Working with data

After this, we started to plot some different graphs starting from:

1. Positive cases per day

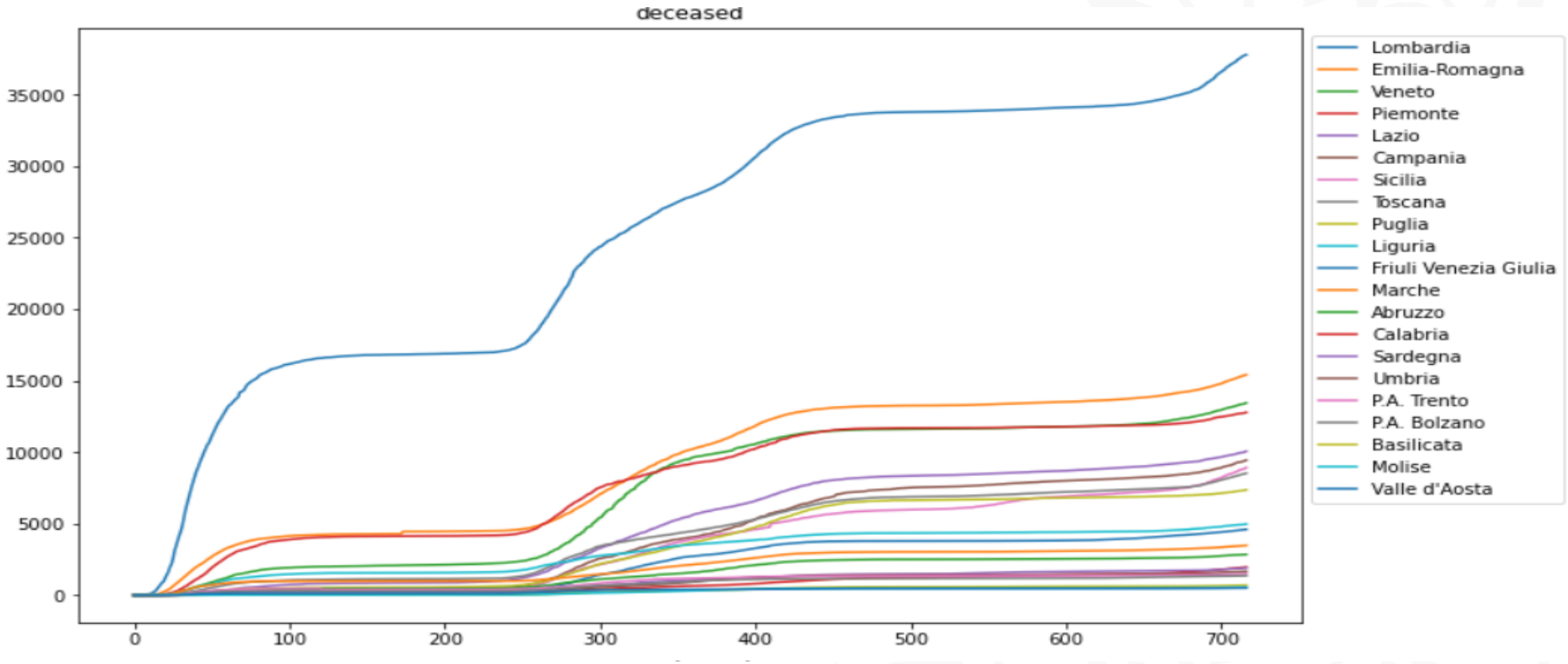


As we can see, the regions with the most number of positive cases are Lombardia, Emilia-Romagna and Veneto.

UNIVERSITÀ DEGLI STUDI FIRENZE

2. Working with data

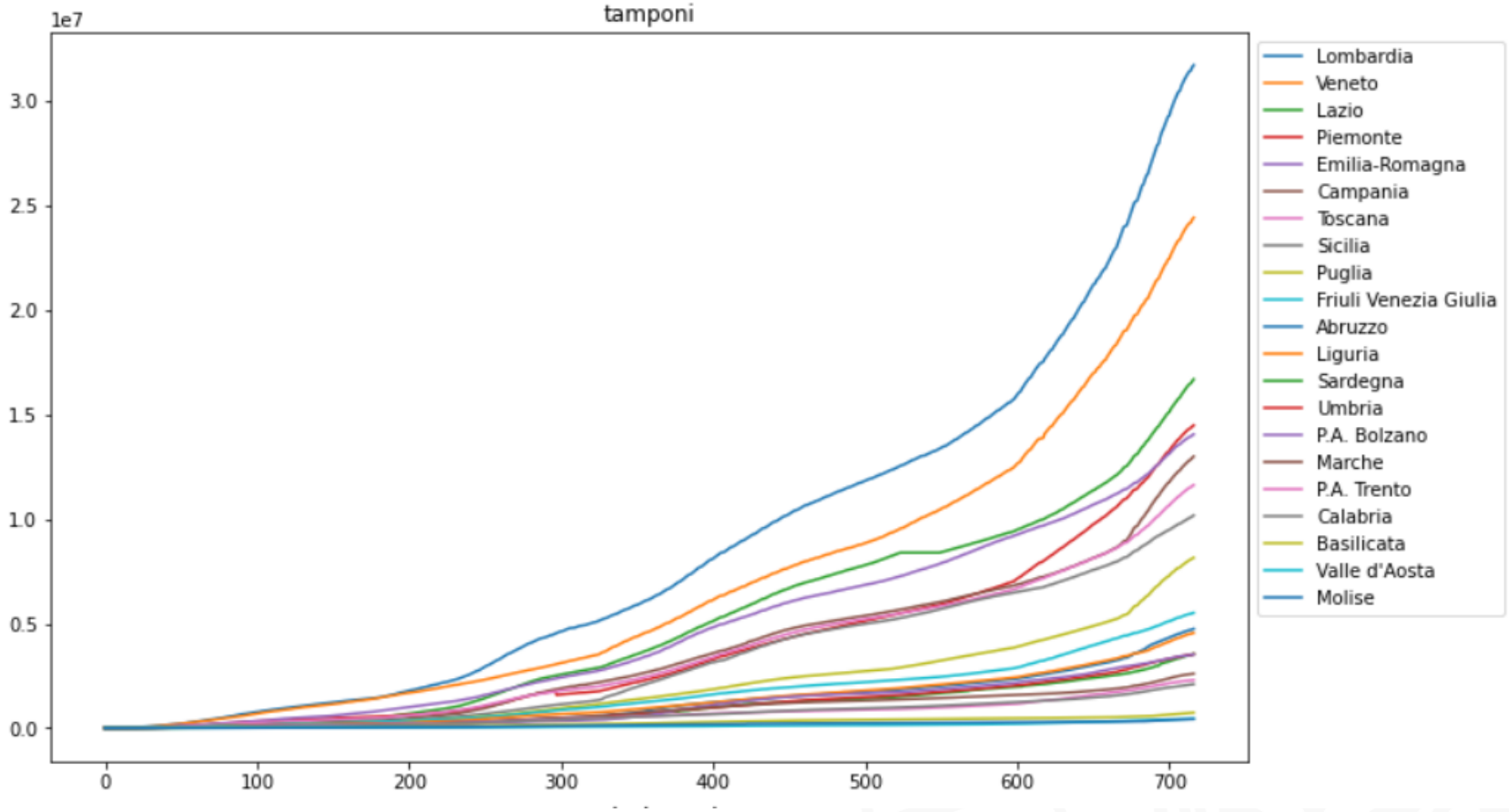
2. Number of deceased persons



UNIVERSITÀ DEGLI STUDI FIRENZE

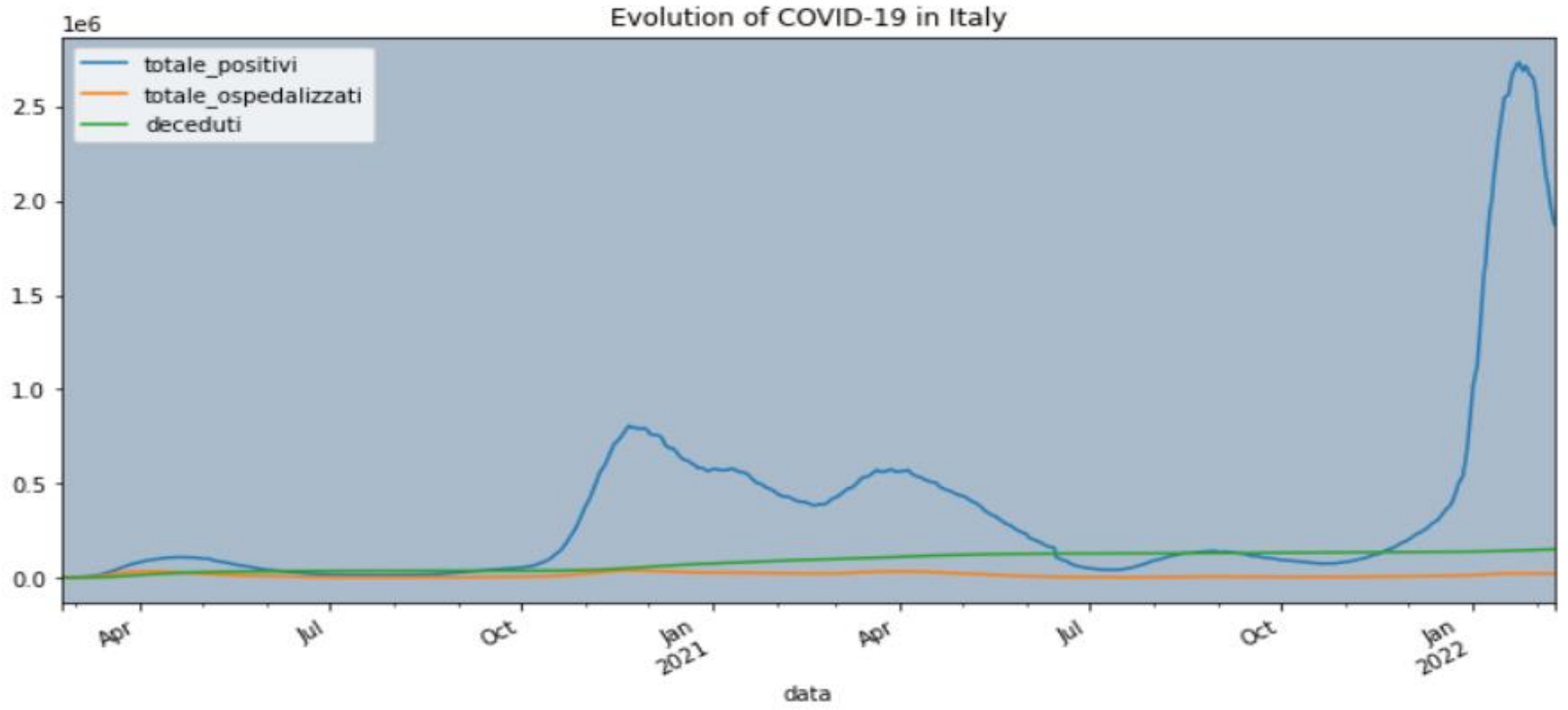
2. Working with data

3. Tampons for every regions



2. Working with data

4. Evolution of Covid-19, we make a comparison between (**totale_positivi, totale_ospedalizzati** e **deceduti**).



2. Working with data

3. dpc-covid19-ita-province.json

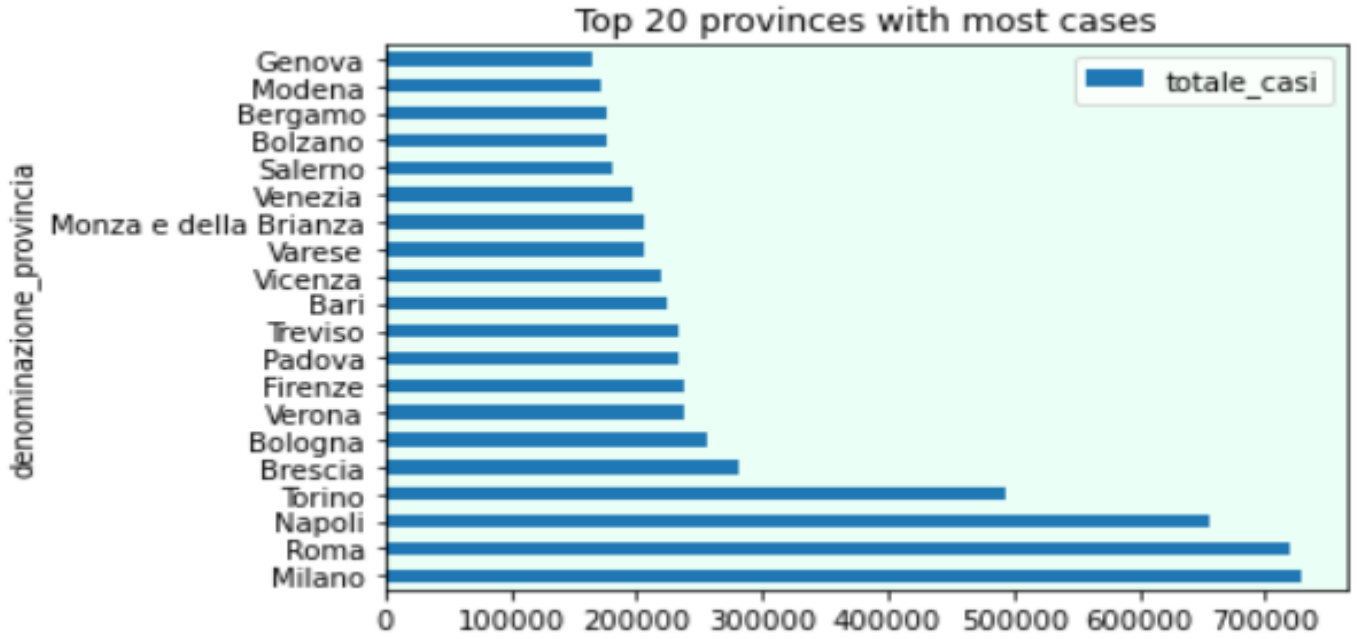
The folder called ‘dati-regioni’ contains data relating to the regional trend of SARS-CoV-2 spread.

Inside each file, data are structured in the 14 fields (one column per field).


```
{
  "data":"2020-02-24T18:00:00",
  "stato":"ITA",
  "codice_regione":13,
  "denominazione_regione":"Abruzzo",
  "codice_provincia":66,
  "denominazione_provincia":"L'Aquila",
  "sigla_provincia":"AQ",
  "lat":42.35122196,
  "long":13.39843823,
  "totale_casi":0,
  "note":null,
  "codice_nuts_1":null,
  "codice_nuts_2":null,
  "codice_nuts_3":null
},
```

2. Working with data

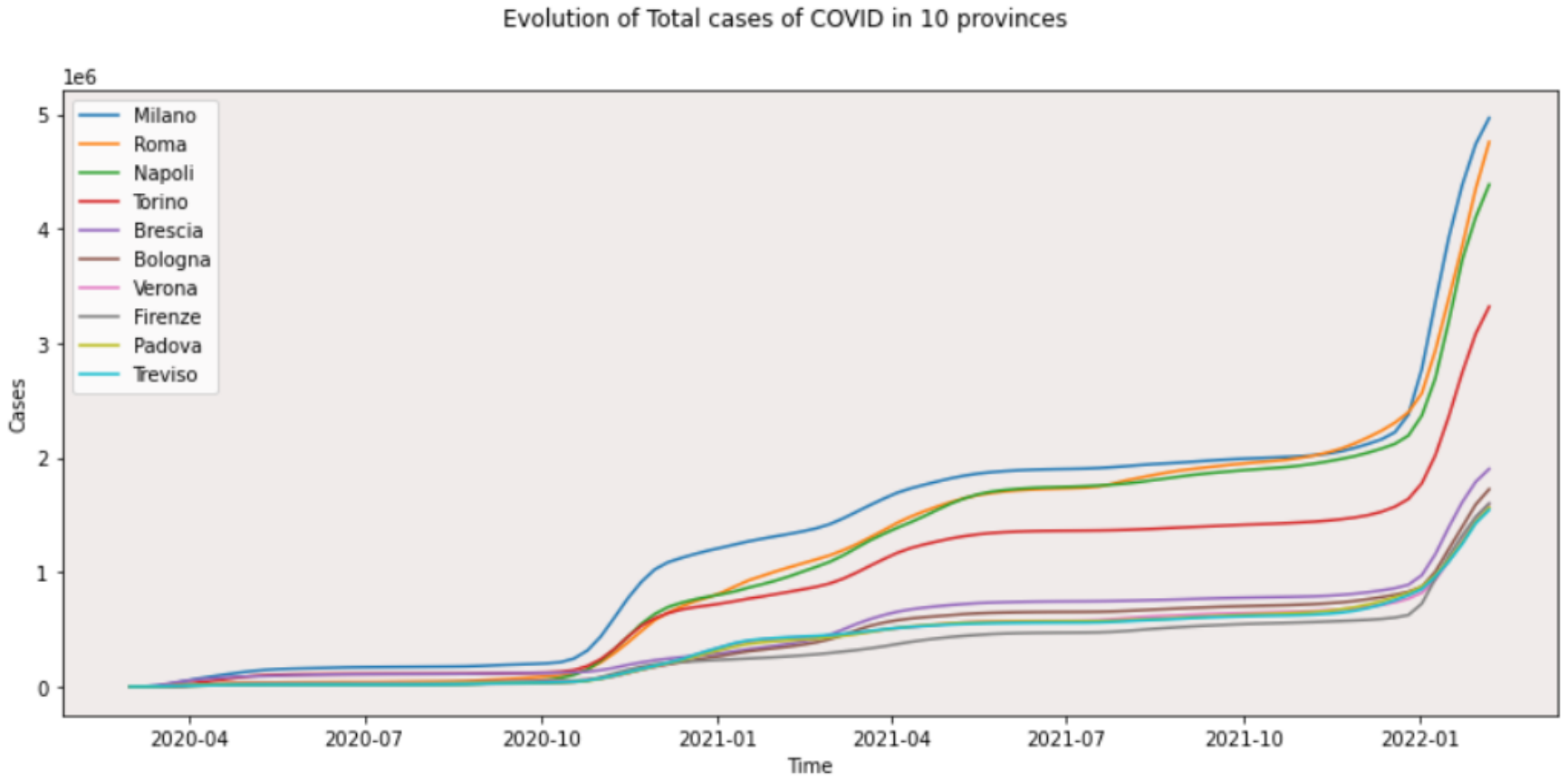
After this, we have displayed the Top 20 provinces with the most number of cases in an horizontal bar.



As, we can see from the results, the most infected provinces are Milano, Roma and Napoli.

2. Working with data

We have displayed the same results in a graph with 10 provinces.



3. Clustering algorithms

We work again with dpc-covid19-ita-regioni.json dataset but this time we are starting our analysis of data from 01/01/2022 until today.

```
var_of_interest["data"] = pd.DatetimeIndex(var_of_interest["data"]).normalize()
var_of_interest = var_of_interest.loc[var_of_interest["data"]>"2022-01-01"]
```

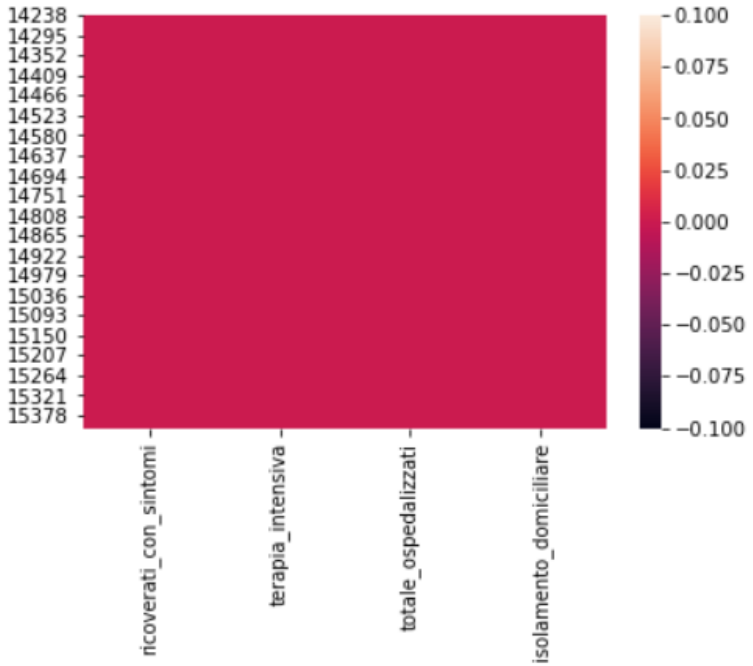
Then, after cleaning the data and transforming it in the right form, we have removed(dropped) column **data** and we have started our analysis of data based on numerical(integer and float) data.

```
{'isolamento_domiciliare',
 'ricoverati_con_sintomi',
 'terapia_intensiva',
 'totale_ospedalizzati'}
```

UNIVERSITÀ DEGLI STUDI FIRENZE

3. Clustering algorithms

After this, we have constructed a heatmap in order to control for NaN values and we have also built the correlation matrix between variables that we have decided to choose..



3. Clustering algorithms

The algorithms that we have used in this project are:

1. K-Means
2. Hierarchical Clustering
3. DBScan

The methods that we have used in order to calculate the number of clusters in K-Means and Hierarchical algorithms are:

1. Silhouette Coefficient
2. Elbow Method

3. Clustering algorithms

The data in our possession is highly aggregated and since it is a dataset without labels the technique that we have implemented is clustering.

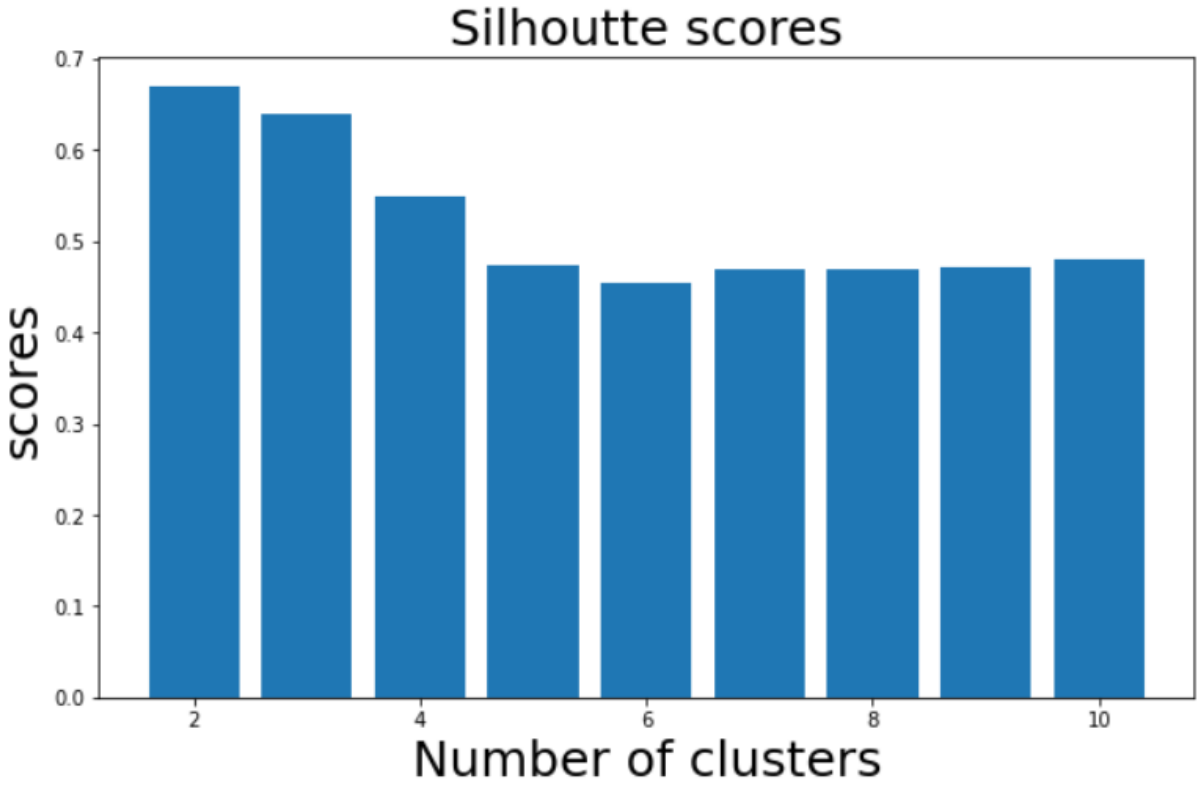
We have started with **K-Means** algorithm, but firstly we have used Silhouette Coefficient and Elbow Method in order to find the best number of clusters in our data.

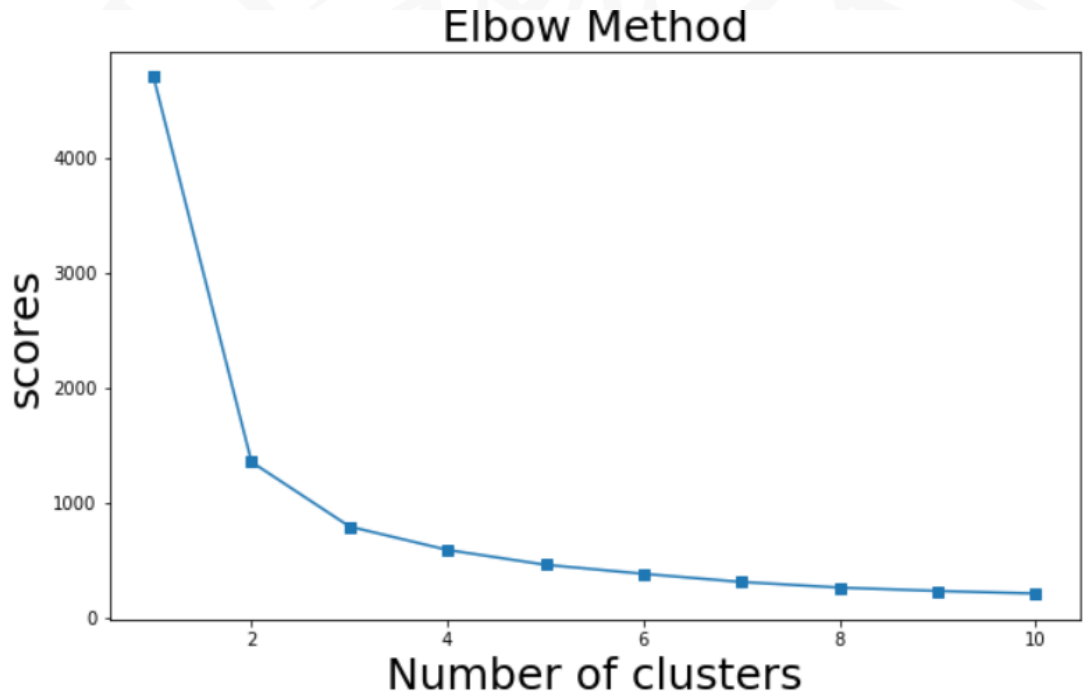
After this, we have worked with time series clustering, an unsupervised data mining technique for organizing data points into groups based on their similarity.

At the same time, we have tried to visualize the correlation between variables and the variation of a variable based on the other.

3. Clustering algorithms

Using Silhoutte coefficient and Elbow Method we can say that the number of clusters we need to use is k=2.





3. Clustering algorithms

What k=2 means?

There are 2 phases or 2 groups of data clustering, the first one represent the phase of where the spread of virus is very high(mostly from 01/01/2022 - 30/01/2022), the second one identifies the second phase(mostly from 01/02/2022 - until now) where the situation is more quiet.

In the last days we see an improvement of the situation, with lower value of data of the variables we are working on**(ricoverati con sintomi, totale_ospedalizzati, terapia_intesiva, isolamento_domiciliare).**

3. Clustering algorithms

As we can see:

There are 2 clusters, the values of the first one are identified with label 0, the seconds one are identified with label 1. The mean of every variable in each cluster is represented as below.

```
clusteringIntepretation = pd.DataFrame(data = X, columns=['ricoverati_con_sintomi' ,'terapia_intensiva' ,'totale_ospedalizzati','isolamento_domiciliare
clusteringIntepretation['labels'] = labels_2
mean=clusteringIntepretation.groupby('labels').mean()
mean
```

labels	ricoverati_con_sintomi	terapia_intensiva	totale_ospedalizzati	isolamento_domiciliare
0	336.036432	27.752513	363.788945	36236.462312
1	1770.197368	148.681579	1918.878947	215741.786842

3. Clustering algorithms

As we can see:

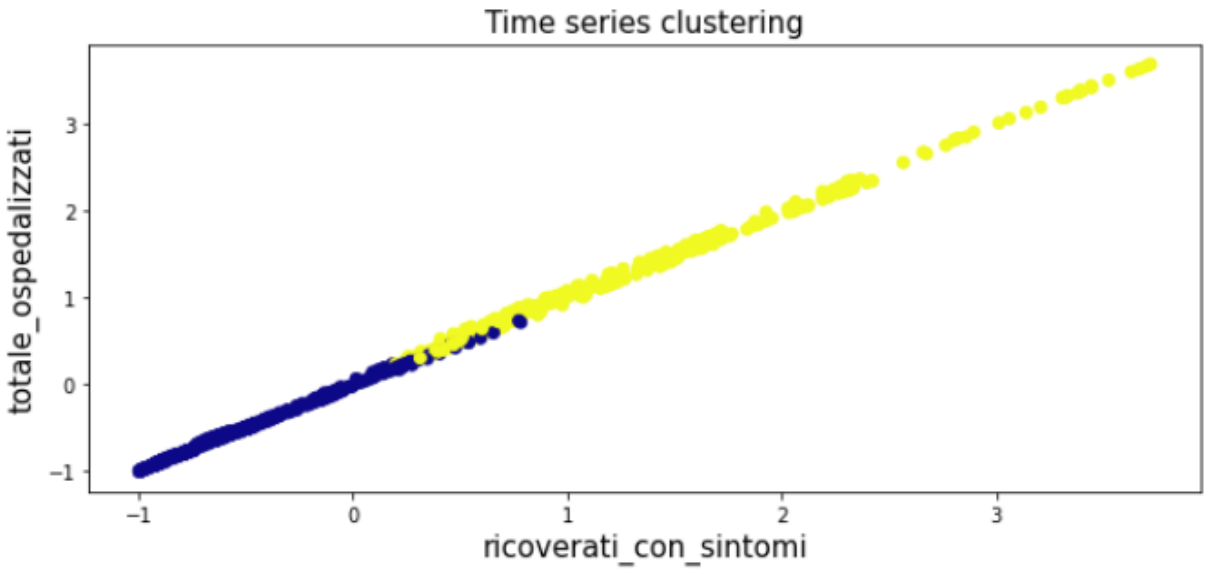
In the first cluster(label=0) there are 796 points and in the second cluster(label=1) there are 380 points.

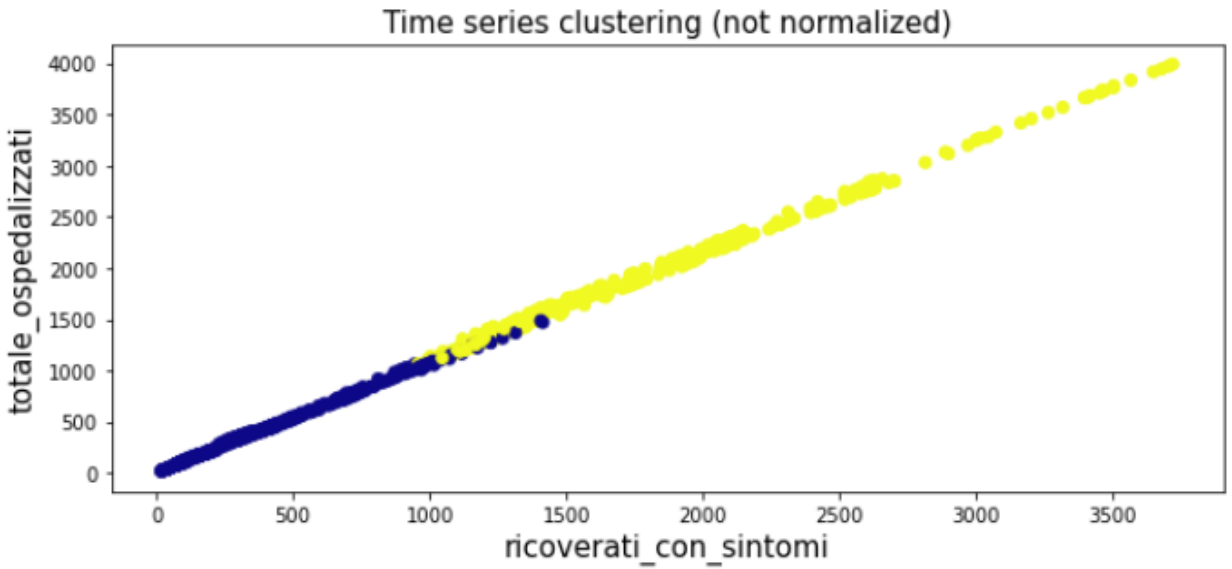
```
clusteringIntepretation.groupby('labels').count()
```

labels	ricoverati_con_sintomi	terapia_intensiva	totale_ospedalizzati	isolamento_domiciliare
0	796	796	796	796
1	380	380	380	380

3. Clustering algorithms

From this graph we can understand that the correlation between **totale_ospedalizzati** and **ricoverati con sintomi** represent the same line as in the case of a linear regression. This means that if we have an increment of data in **ricoverati con sintomi** variable, then we will an increment also in the **totale_ospedalizzati** variable.





3. Clustering algorithms

Interpreting the results:

So, the cluster is based on the variables that we have chosen. In the first cluster there are 796 point, in the second there are 380 points. Looking also the mean of the varaibles in each cluster we can conclude that:

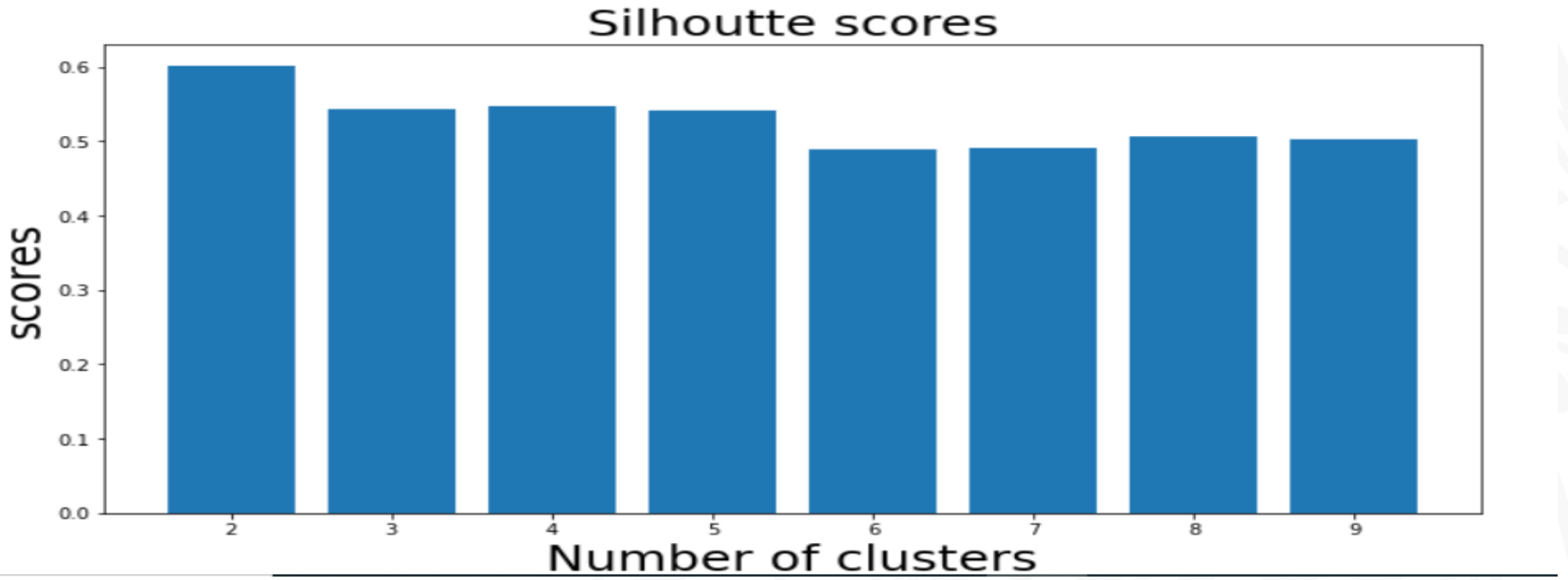
The number of data in the first cluster(label=0) is two times more larger than the number of data in the second cluster(label=1). Mostly, the regions with the most higher values of data are included in the second cluster. These regions are: Lazio, Piemonte, Lombardia, Emilia- Romagna and also a part of data in the region of Toscana.

3. Clustering algorithms

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively.

This hierarchy of clusters is represented as a tree (or dendrogram).

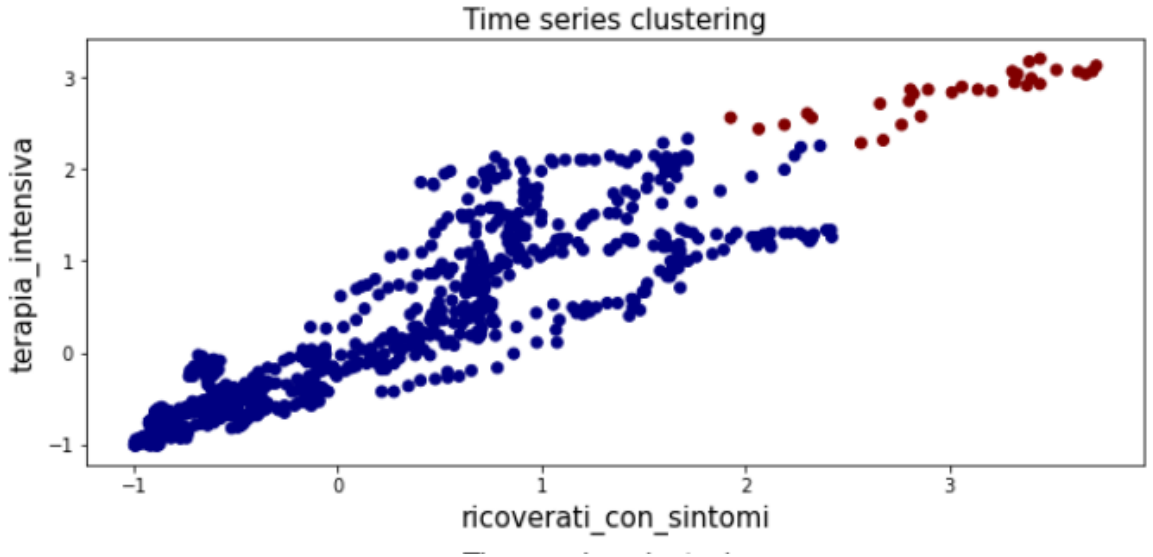
Using Silhoutte coefficient the ideal number of cluster, also for the hierarchical clustering is equal to 2.

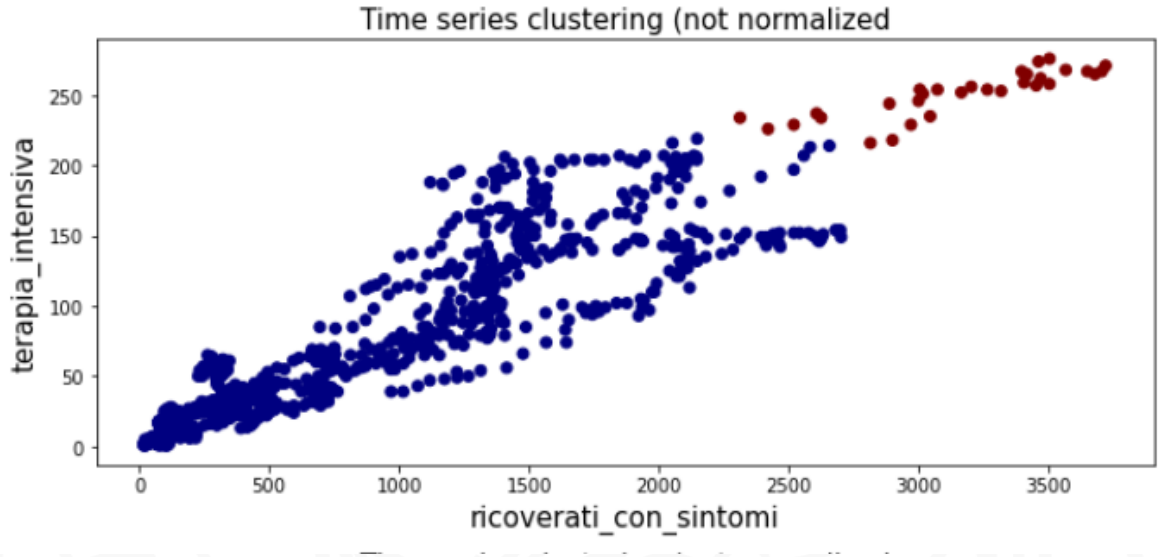


3. Clustering algorithms

In our example we have used agglomerative clustering that start with the points as individual clusters, than at each step it merges the closest pair of clusters until one cluster left.

In our example the linkage criterion that we have used is **'average':**





3. Clustering algorithms

As we can see:

There are 2 clusters, the values of the first one are identified with label 0, the seconds one are identified with label 1. The number of points and the mean in each cluster is represented as below.

```
clusteringIntepretation.groupby('labels').count()
```

labels	ricoverati_con_sintomi	terapia_intensiva	totale_ospedalizzati	isolamento_domiciliare
0	1144	1144	1144	1144
1	32	32	32	32

```
clusteringIntepretation.groupby('labels').mean()
```

labels	ricoverati_con_sintomi	terapia_intensiva	totale_ospedalizzati	isolamento_domiciliare
0	733.124126	61.667832	794.791958	84218.718531
1	3170.812500	251.312500	3422.125000	452496.531250

3. Clustering algorithms

Interpreting the results:

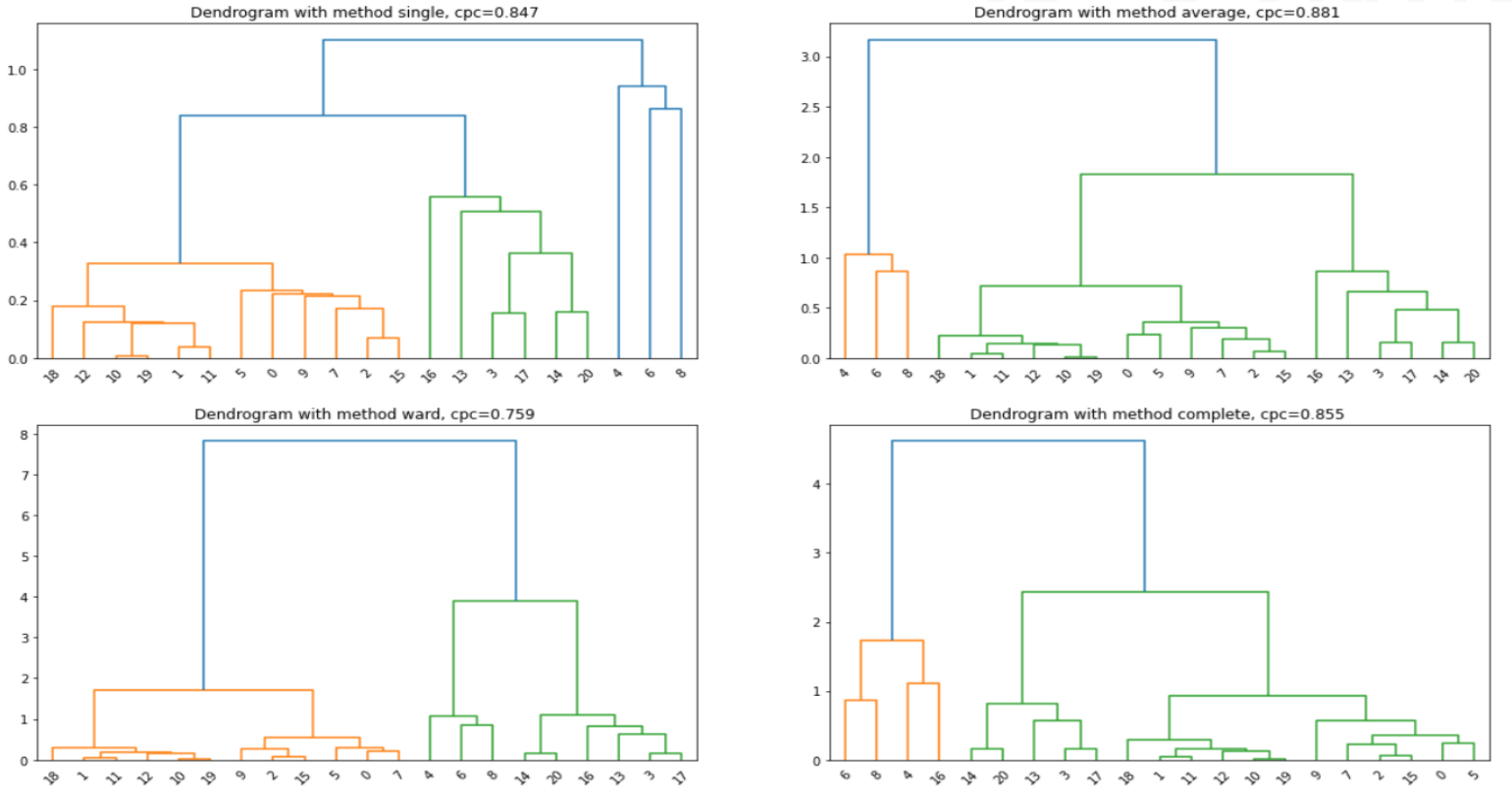
Using Hierarchical Clustering the number of clusters using Silhouette Coefficient did not changed, but the data in each cluster are different. In the second cluster there are just 32 points and this means that the values of the chosen variables in the second cluster are very high. The mean of the values in the second cluster is higher respect to the mean of the second cluster in the K-Means Algorithm.

The values part of the second cluster in Hierarchical Clustering come from:

- Lombardia

3. Clustering algorithms

A **dendrogram** is a type of tree diagram showing hierarchical clustering or relationships between similar sets of data. In our example we have constructed a dendrogram just for the last day of [dpc-covid19-ita-regioni.json] file in order to see how this diagram works.



3. Clustering algorithms

We are taking in consideration the last day of dataset in order to construct the dendrogram and the variables on which we are working are:

1. 'ricoverati_con_sintomi' ,
2. 'terapia_intensiva'

We have done this selection of data and variables for simplicity and and also to understand and interpret more easily the results of algorithms.

The **cophenet** function used in the Jupyteer notebook file compares how well the cluster tree generated by the linkage function reflects our data, returning a value called the cophenetic correlation coefficient. The closer the value of the cophenetic correlation coefficient is to 1, the more accurately the clustering solution reflects our data.

3. Clustering algorithms

Interpreting the results:

In our example the method that has the greatest cophenetic correlation coefficient is Average Method(0.881). This means that this method reflects better our data.

In the figure, the numbers along the horizontal axis represent the indices of the regions in the original data set. The links between objects are represented as upside-down U-shaped lines. The height of the U indicates the distance between the objects.

Branches in the dendrogram represent the similarities among the regions – the shorter the branch, the greater similarity of the regions, the height of the link represents the distance between the two clusters that contain those two objects.

3. Clustering algorithms

Interpreting the results:

Usually, the cut of dendrogram is made where it has jump, corresponding to the max distance. So, from the dendrogram we can say that the number of cluster is 2.

If we analyze the dendrogram with **average method and k=2,** we can see that the region with index 4(Emilia Romagna), 6(Lazio) and 8(Lombardia) are in the **first cluster** because they have more similarity between each other because in the last day of dataset they have the greatest number of 'ricoverati_con_sintomi' and 'terapia_intensiva'. If we see the numbers of 'ricoverati_con_sintomi' and 'terapia_intensiva' in other regions it seems lower. That's why they are part of **second cluster.**

3. Clustering algorithms

DBSCAN is one of the most common clustering algorithms, it groups together points that are closely packed together (points with many nearby neighbors), marking as outlier points that lie alone in low-density regions.

In our example the DBScan algorithm is not applied due to the different density present in the data(dpc-covid19-ita-regioni.json) but we have implemented another example in order to represent how it functions.

The DBSCAN algorithm uses two parameters:

minPts: The minimum number of points (a threshold) clustered together for a region to be considered dense.

eps (ε): A distance measure that will be used to locate the points in the neighborhood of any point.

3. Clustering algorithms

In our example we have used another example of DBScan algorithm due to the different density present in the data(dpc-covid19-ita-regioni.json).

The dataset is generated by **sklearn.datasets.make_blobs**.

The **make_blobs()** function draws samples from a special **Gaussian mixture model**.

A **Gaussian mixture model** is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

A general Gaussian mixture model with k clusters has a density of the form:

$$p(x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(\mu_i, \Sigma_i)$$

3. Clustering algorithms

Isotropic refers to the fact that the covariance matrices will all be diagonal

$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$$

with σi being the standard deviation that is passed in. By default, all clusters will have the same standard deviation.

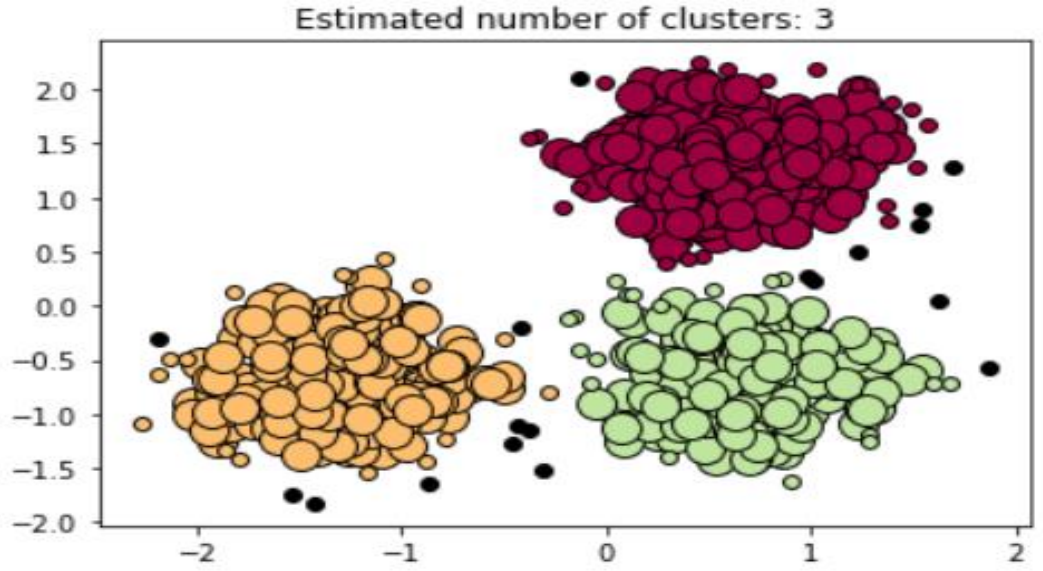
A Gaussian mixture model is not Gaussian unless there is only one cluster, but rather a combination of Gaussians.

3. Clustering algorithms

In our example we will have **eps =0.3**, **minPts=10**, **cluster_std=0.4** that means how tightly data are clustered around the mean, **n_samples=750** total number of points equally divided among clusters.

The results will be:

```
Estimated number of clusters: 3
Estimated number of noise points: 18
Homogeneity: 0.953
Silhouette Coefficient: 0.626
```



UNIVERSITÀ DEGLI STUDI FIRENZE

3. Conclusions

- **Data Understanding:** We have worked with github.com/pcm-dpc/COVID-19 dataset and we have analyzed the structure and main files of this directory.
- **Data Preprocessing:** We have normalized and standardized the data, dropping the unncessary data, we have represented different graphs in order to understand better the persistence of the virus in various provinces and regions of Italy.
- **Clustering algorithms:** The dataset in our possession is very aggregated and since it is a dataset without labels, we have applied K-means clustering and hierarchical clustering for classifying the data.
- We have used **Gaussian mixture model** as an example for understanding better DBScan algorithm.