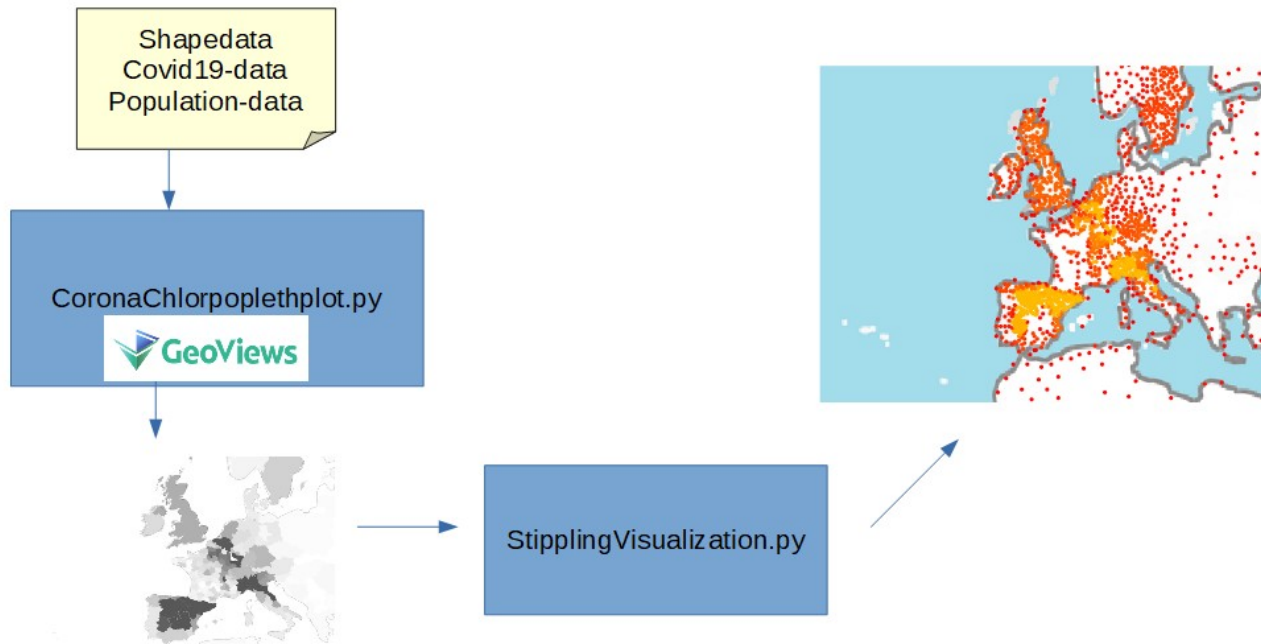


Coronoa-Stippling-Generator



This program collects different data sources of the covid-19-pandemic to present it in a stippling-visualisation. The generation of the plot is processed in two steps: At first, a choropleth map is created from the raw data. In the second stage, out of the intensity values in the choropleth images the stippling visualisation plots are generated.

This is done for every day. In a GUI-program, the visualizations can be explored.

Chloropleth Generator

The following files belong to the choropleth generation:

DataDownloader.py: Here functions to get the pandemic-data for different geographical entities are stored. Included data sources are:

Global data (mostly on country level, but also american countys, chinese provinces, for recent dates mexico and brazil ...): <https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/>

Data per italian region: <https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-regioni/>

Data per spain region: <https://raw.githubusercontent.com/datadista/datasets/master/COVID%2019/>

Data per german state:

<https://raw.githubusercontent.com/jgehrcke/covid-19-germany-gae/master/>

Data per france department:

<https://raw.githubusercontent.com/opencovid19-fr/data/master/dist/>

PolyLoader.py: Here the functions to generate the geographical polygons are stored. For each integrated geographical unit, the covid19-data is collected by the respective downloader function (DataDownloader.py). To make the data relative to the population in the respective country, population data of the considered regions has been collected in csv-files. (data/Population/countryname.csv)

To generate the geoviews-polygon, this data is combined with the respective shape data (from data/GeoShapes/<countryname>). The color of the polygon is determined by the corona data and the given parameters.

CoronaChoroplethPlot.py: This file processes the configurable parameters to call the polyloader in the defined way.

With the defined covid19-data (deaths/confirmed, recovered) , date, minimum value to return black polygons it calls the poly-loader functions (PolyLoader.py) for the activated countries. (Germany, Italy, Brazil, Mexico, Spain, US, France, China). The returned polygon is combined with every other generated one to create the big choropleth map. This chloropleth map can be rendered as a bokeh-plot in html-format or as a png-file. For further processing, the png-files are used.

```
options = dict(
    feature="Deaths", # confirmed deaths, Recovered, |
    value_format="per1mio", #absolute, per1mio
    country_borders=0, #thickness
    colorvalue_limits=(0,300), #bounds for values to get
    htmlplot=False,
    pngplot=True,
    filename="plotrel",
    coastline=True,
    # renderworld= True
    renderGermany1=False,
    renderItaly1=False,
    renderBrazil1=True,
    renderMexico1=True,
    renderSpain1=False,
    renderUS2=False,
    renderFrance2=False,
    renderChina1=False,
    renderBackgroundMap=False)
```

MakeCoronaEveryDay.py: This file iterates over every date and calls the Chloroplethplot to generate a png for it.

Stippling

stippling.py: Here the actual algorithm is included. The method takes an image file and outputs the coordinates of each stipple.

StipplingVisualisation.py: This file gets as a parameter a date. For this date, the respective image-file is searched to feed it into the stippling algorithm. The resulting stipples are plotted on a background map and the plot is saved in a file.

MakeStipplingEveryDay.py: This file iterates over every date and calls the StipplingVisualisation.py to generate the respective plot.

GUI

The GUI (**StipplingGUI.py**) is a simple Matplotlib interface with two buttons and a slider. Using the slider and/or the buttons, one can move through the visualization over time, which shows the development of the pandemic over the last months. The GUI supports zooming to get a better look at details and an overview for the zoomed-out image. Since we use raw data to ensure that the information doesn't get lost regardless of zoom level, the GUI can be slow in changing between days.