



National Technical University of Athens
MSc - Data Science and Machine Learning

Geospatial Big Data Analytics

Lab 2
Geospatial Services and Web Applications

Vasileios Depastas
MSc student

A.M: 03400131
vasileiosdepastas@mail.ntua.gr

May 2022

Contents

1	Introduction	1
2	Question 2	1
2.1	Question 2.1	1
2.2	Question 2.2	4
2.3	Question 2.3	6
2.4	Question 2.4	8
3	Question 3	9
3.1	Question 3.1	9
3.2	Question 3.2	11
3.3	Question 3.3	16
3.4	Question 3.4	18
4	Question 4	21

List of Figures

1	Kastoria region - image: 1, band: blue, matplotlib	2
2	Kastoria region - image: 1, band: blue, rasterio	2
3	Kastoria region - image: 1, bands: red, green, blue	3
4	Kastoria region - image: 1, RGB	3
5	Kastoria region - image: 1, ndvi histogram	4
6	Kastoria region - image: 1, ndvi	4
7	Kastoria region - image: 1, RGB, EPSG:32634	5
8	Kastoria region - image: 1, RGB, EPSG:4326	5
9	Kastoria ROI mask	6
10	Kastoria ROI, Image: 1, RGB	6
11	Kastoria ROI, Clipped dataset image collection, band: NDVI . .	7
12	Kastoria ROI, average NDVI pixel value	8
13	Kastoria ROI, average NDVI per DOY timeseries	8
14	Greece, airports table	9
15	Greece, RGB raster basemap	10
16	Greece, RGB raster airports	10
17	Greece, RGB raster airports hydrographic network	11
18	Greece, RGB raster counties' borders	11
19	Greece, Top 10 regions in heating energy demand 2015 table . .	12
20	Greece, RGB raster top 10 regions in heating energy demand 2015	12
21	Greece, Top 5 municipalities having most airports in their territories	13
22	Greece, RGB raster municipalities containing airports	13
23	Athens, RGB raster	14
24	Athens, RGB raster public transport stations	15
25	Athens, RGB raster public transport stations by means of transport	16
26	Belgium, RGB raster	17
27	Belgium, RGB raster CLC 2018	18
28	Ghent, geojson ROI	18
29	Ghent, ROI temperature at 2 meters timeseries	19
30	Ghent, ROI wind speed at 10 meters timeseries	20

31	Ghent, ROI relative humidity at 2 meters timeseries	20
32	Ghent, ROI precipitation (corrected) timeseries	21
33	Greece, interactive map snippet	22

List of Tables

1 Introduction

In this exercise, we experiment with some of the most widely used libraries and APIs for importing, processing and visualizing geospatial data. During the implementation, we also make use of relevant web services-applications and the standards of the Open Geospatial Consortium (OGC).

Some of the libraries that were used are the following ones:

Library	Usage
OWSLib	Python package for client programming with OGC web service interface standards.
rasterio	Python module for raster image processing used for reading and writing several raster formats.
geopandas	Extension of Python's pandas library's datatypes to allow spatial operation on geometric type. Depends on shapely library for geometric operations, fiona for file accessing and matplotlib for plotting.
fiona	Python library for reading and writing geographic vector data files.
folium	Python's library used for visualizing geospatial data. A wrapper of leaflet.js javascript library for interactive maps plotting.
leafmap	Python package for interactive mapping and geospatial analysis in a Jupyter notebook environment.

2 Question 2

Here, we present raster image data by visualizing different bands and spectral indexes as well as their variation over time.

2.1 Question 2.1

First of all, we download Sentinel-2 images of the broader **Kastoria region** (ROI) in northern Greece from [here](#). A pdf file explaining the dataset accompanies the image series. According to that, the dataset contains two *.tif* large files:

1. *Kastoria.tif*: 24 raster images spanning from 25-01-2016 until 30-12-2016 (corresponding doy range: 25 - 365) with 10 bands per image. This results in a spectro-temporal cube of dimension 240 x 2017 x 2281.
2. *Kast_RefData_26Classes.tif*: image with 26 class codes for different land usages classification.

To load the data, we make use of the rasterio library. From the image collection of *Kastoria.tif*, we then select the first image's blue band according to

the tiff's pdf documentation downloaded. We can then use either matplotlib or rasterio to plot the image's band. However, using rasterio's `plot.show()` function allows us to maintain proper geo-referenced labeling for the two axes. Therefore, we will prefer this plot type moving forward.

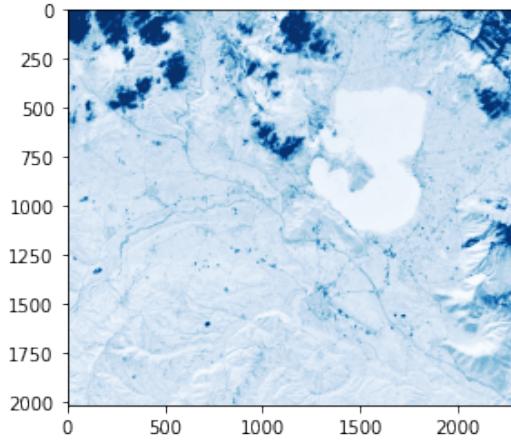


Figure 1: Kastoria region - image: 1, band: blue, matplotlib

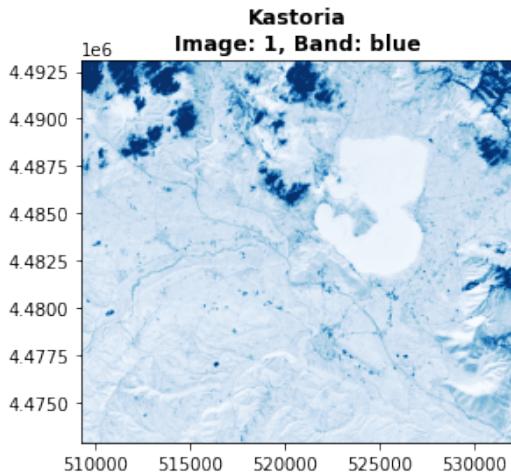


Figure 2: Kastoria region - image: 1, band: blue, rasterio

Then, for this first image of the dataset, we plot alongside each other the three bands, Red, Green and Blue.

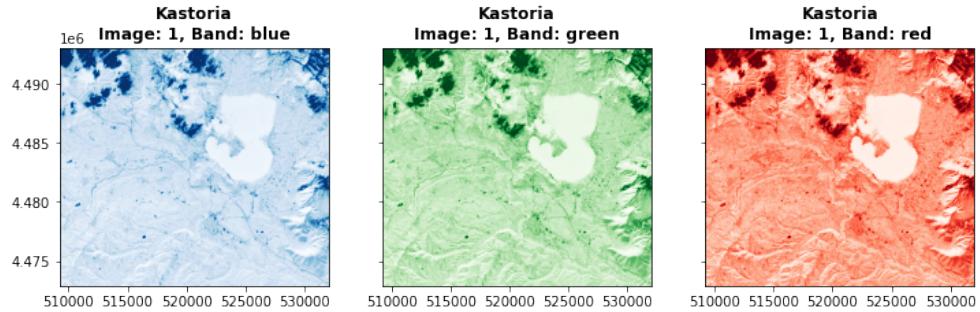


Figure 3: Kastoria region - image: 1, bands: red, green, blue

We also plot the RGB image of the dataset's first image for the Kastoria region by combining the three bands, Red, Green and Blue.

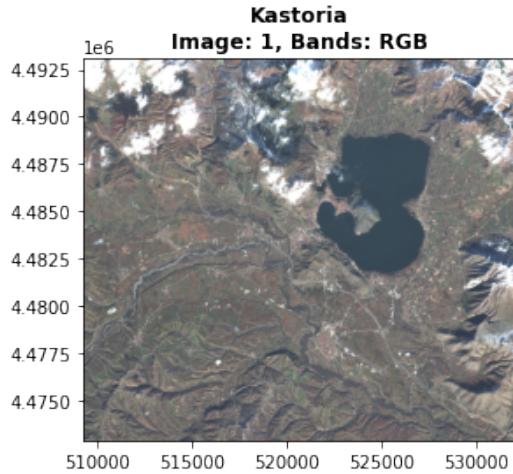


Figure 4: Kastoria region - image: 1, RGB

Moving forward, we calculate the NDVI index band for the first image of the image collection dataset using the near infrared (NIR) and RED bands as $NDVI = \frac{NIR-RED}{NIR+RED}$. Using this band, we can then create a histogram of the values for the ndvi band for all the image's pixels as well as plot the image of the ndvi band in red-yellow-green color map

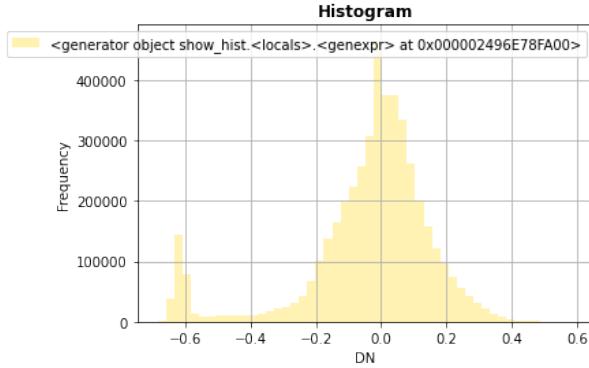


Figure 5: Kastoria region - image: 1, ndvi histogram

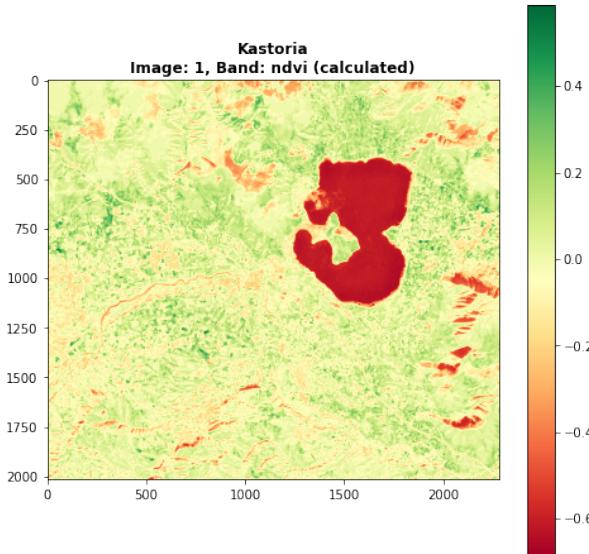


Figure 6: Kastoria region - image: 1, ndvi

In general, NDVI values range from -1.0 to 1.0, with negative values indicating clouds and water, positive values near zero indicating bare soil, and higher positive values of NDVI ranging from sparse vegetation (0.1 - 0.5) to dense green vegetation (0.6 and above). Given that the first image is taken on 25-01-2016, and hence during winter time, the vegetation observed is mostly sparse rather than dense.

2.2 Question 2.2

Now, we select a polygon as a region of interest (ROI) within the broader Kastoria region for which we will retrieve raster, vector and timeseries data using different data sources. Towards this purpose, we use geojson.io to draw a polygon region of interest (ROI) in Kastoria and download the shapefile related to that in order to clip it afterwards from the images of the dataset and concentrate

subsequent steps on that ROI solely. A shape file stores both the geometric location and attribute information of geographic features like lines, points or polygons. Clipping an image is also called masking (of a raster file by a shape file). To read a shapefile we utilize the *geopandas* library.

Loading the shapefile with geopandas, we were able to identify its coordinate reference system (crs) as EPSG:4326, which is different to the one of the Kastoria raster images dataset, EPSG:32634. Therefore, a transform was performed to the dataset.

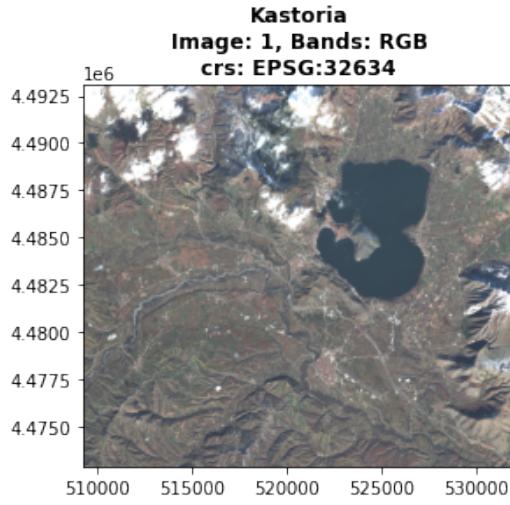


Figure 7: Kastoria region - image: 1, RGB, EPSG:32634

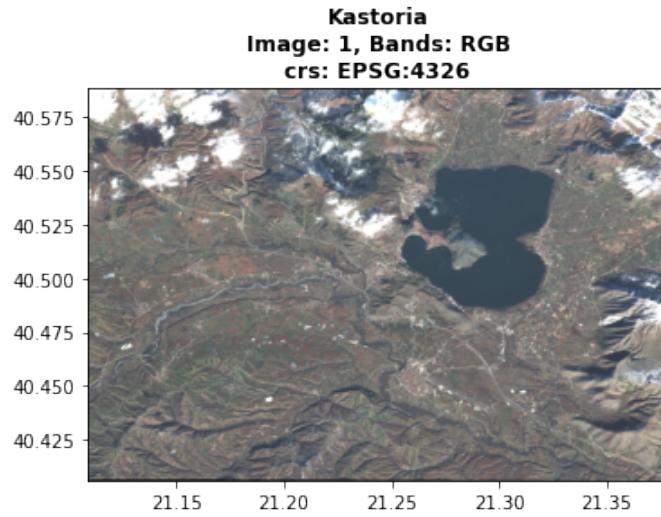


Figure 8: Kastoria region - image: 1, RGB, EPSG:4326

Once the crs transform has been completed, we can see that the two axes

values have changed. The image can be now clipped with the selected polygon. First, we open the shapefile with fiona this time and then mask the raster dataset with the vector data using rasterio's mask function. We end up with 240 images with size $width \times height = 534 \times 389$ with EPSG:4326 crs. In the next image, we can see in blue the selected polygon ROI in Kastoria. Any subsequent analysis and steps will concern this ROI.

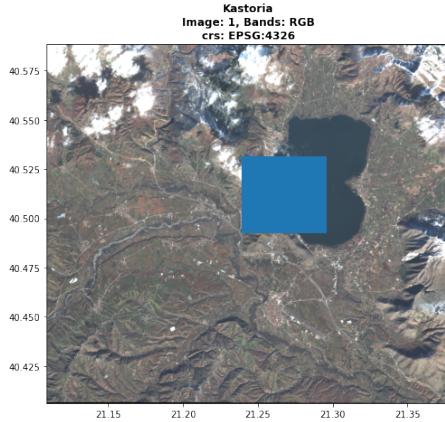


Figure 9: Kastoria ROI mask

We can see that the clipped area selected (ROI) is a rectangular that contains both water and land areas in Kastoria in order to spot the differences in different types of land usage in subsequent analysis.

Next, we plot the first image (first date) of the clipped dataset in RGB.

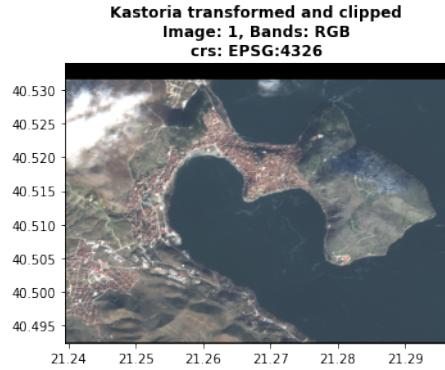


Figure 10: Kastoria ROI, Image: 1, RGB

2.3 Question 2.3

Upon transforming the crs of the dataset and clipping all 24 images of the dataset to the selected polygon ROI, we now move on to visualize the NDVI index for each available date as well as to calculate the average NDVI index value across all pixels in the ROI.

Katoria transformed and clipped dataset NDVI band

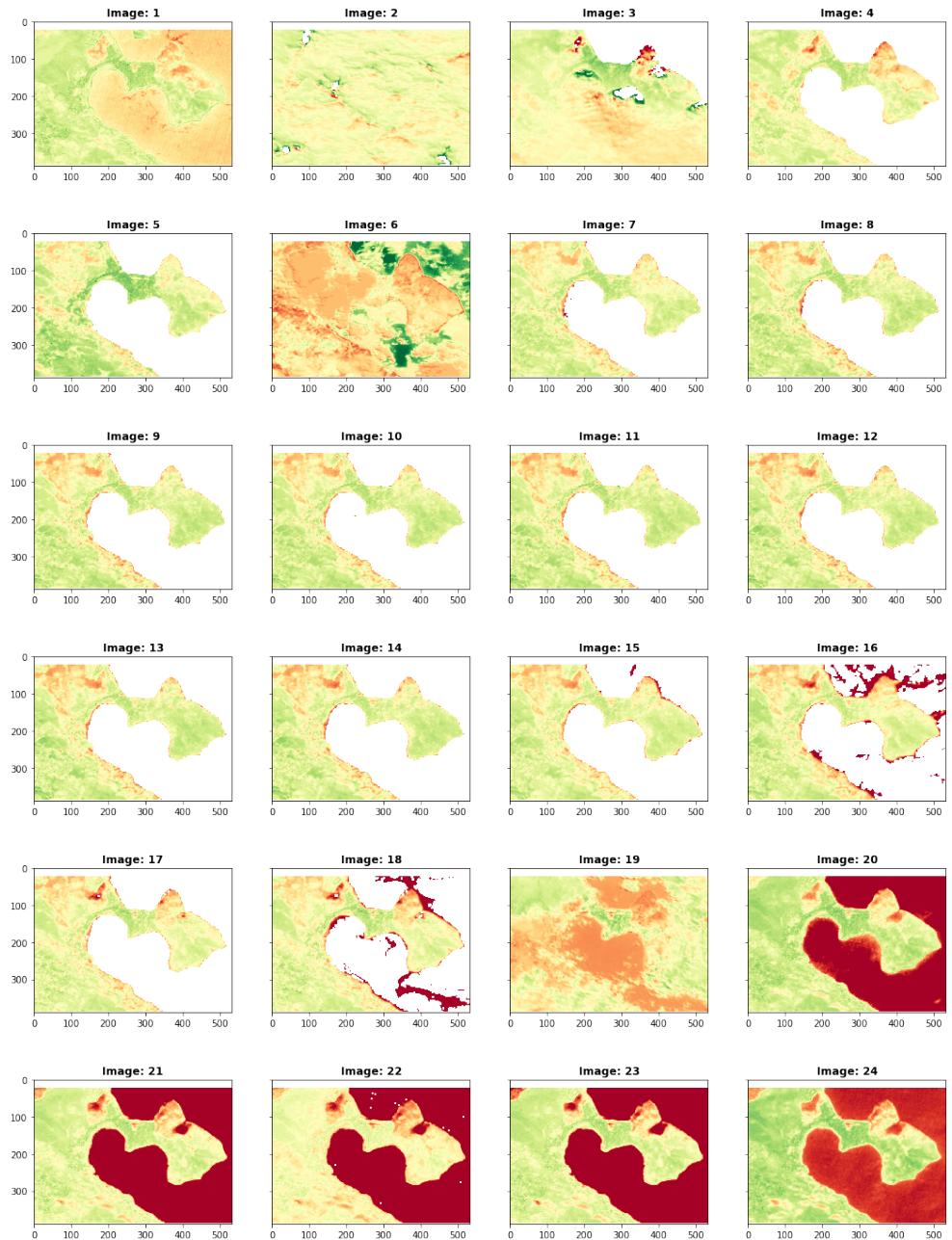


Figure 11: Kastoria ROI, Clipped dataset image collection, band: NDVI

The previous NDVI band plot for each date was presented in a red-yellow-green color map.

Also, the average NDVI index value for each pixel over the 24 images has been calculated and summarized in a single plot that follows. The minimum pixel ndvi value in this image that contains average ndvi pixel values is -0.635 whereas the maximum value is 0.641 .

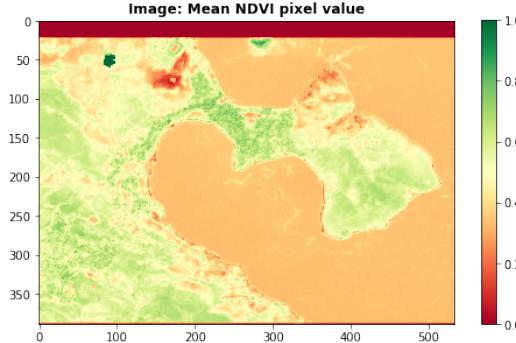


Figure 12: Kastoria ROI, average NDVI pixel value

2.4 Question 2.4

Following on with our NDVI band analysis, we present a graph with the progress of the NDVI index in time for the ROI. For this task we will calculate the mean value of all pixels of the NDVI band for each image/date and then plot the resulting time series.

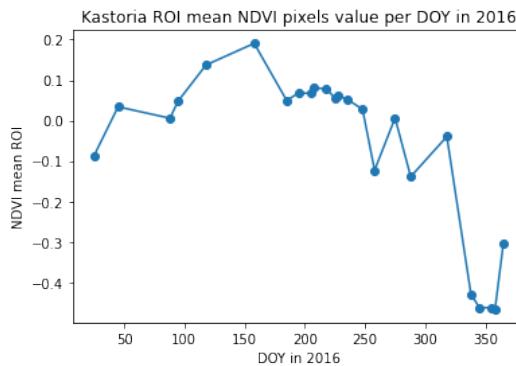


Figure 13: Kastoria ROI, average NDVI per DOY timeseries

We observe that the highest mean NDVI pixels value is found on the 158th day of 2016 (2016-06-06), right in the summer which is what we would expect. On the contrary, we observe lower values for winter months towards the end of year 2016 and at the beginning of it. Also, given that the ROI selected contains a large part of water area, the NDVI mean values are fluctuating around 0,

whereas if the area was mostly land, we might have expected higher mean ndvi values.

3 Question 3

In this section of the assignment, we experiment with several geospatial queries and operations utilizing the geopandas library for vector data and we subsequently create static maps. Also, we do present graphs with the time evolution of meteorological variables.

3.1 Question 3.1

Firstly, we download vector data from <https://geodata.gov.gr>. Geodata offers vector data for a range of topics eg. points of the country's airports, public transport stations and playgrounds among others. The way we download the data is by connecting to the WFS service geodata offers through OWSLib library. We can then inspect the available vector data to retrieve from the service.

We inspect first the airports of Greece. Below is a table of the 74 airports fetched from geodata Web Feature Service (WFS).

	id	ονόμα	geometry
0	1	Δασκαλογιάννης CHQ Χανιά, Σούδα	POINT (512886.533 3932001.696)
1	2	Μάλεμε	POINT (484540.131 3931694.148)
2	3	Τυμπάκι LG54\ln	POINT (568297.267 3880601.588)
3	4	Νίκος Καζαντζάκης LGIR HER Ηράκλειο	POINT (607090.104 3911052.691)
4	5	Καστέλι LGTL\ln	POINT (620681.353 3894883.025)
...
69	70	Μήλου LGML MLO	POINT (542178.120 4061029.554)
70	71	Πάρος LGPA PAS	POINT (600203.447 4096454.090)
71	72	Νάξου LGNX JNK	POINT (621458.775 4104866.922)
72	73	Μύκονος LGMK JMK	POINT (619008.074 4143753.749)
73	74	Σύρου LGSO JSY	POINT (583893.698 4141916.215)

Figure 14: Greece, airports table

Prior to placing markers on a map of Greece, eg a marker for each airport, we need to find a raster RGB image of Greece. Towards this purpose we can use a Web Map Service (WMS) to download a raster RGB image that contains Greece (eg European raster image) and then select a bounding box for Greece. We selected [this](#) WMS service for Europe that was found through [Geoseer](#) spatial data search engine. Upon making a getmap request to the WMS service, we get a EPSG:4326 crs image of Greece, where we set the bounding box based on the coordinates found [here](#). The image is saved as a geotiff and we present it below (in RGB):



Figure 15: Greece, RGB raster basemap

Then, to add on top of the map any vector geometries, we need to first convert their crs to the same crs as the raster, which is EPSG:4326. The first layer we add on the map is that of airports, where we denote each airport's location with a black marker.

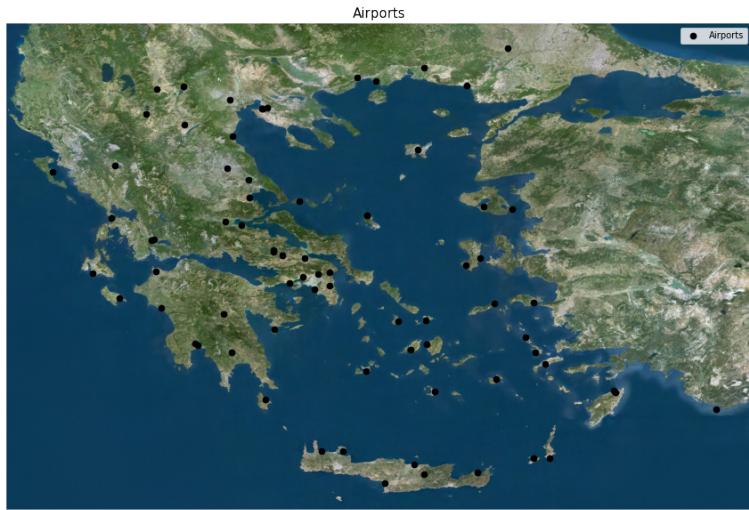


Figure 16: Greece, RGB raster airports

We can stack layers on a single raster image, for instance we next present a map with two layers, the previous one with the airports of Greece and another one containing the hydagraphic network of the country.

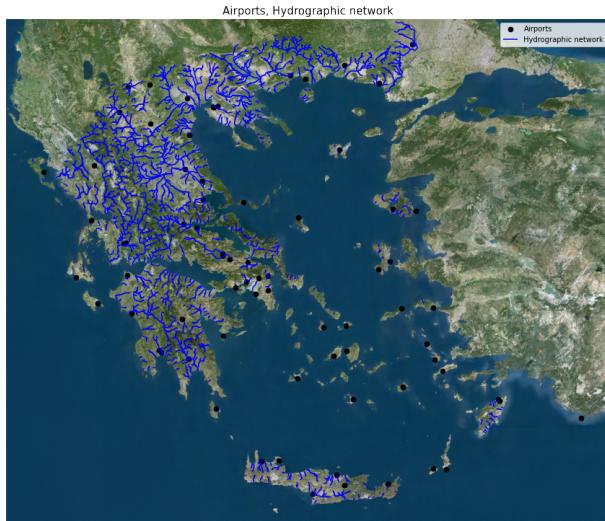


Figure 17: Greece, RGB raster airports hydrographic network

On top of the above, we create another map by combining the raster image with the layer containing Greece's counties below. We can then distinguish the borders of different counties in Greece.

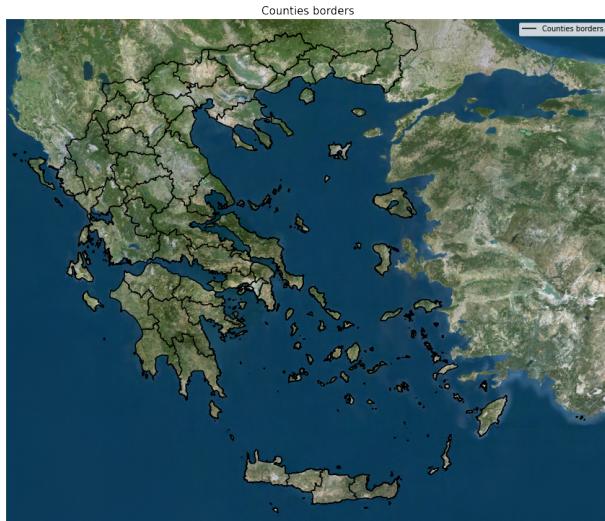


Figure 18: Greece, RGB raster counties' borders

3.2 Question 3.2

Next, we implement spatial queries with the geopandas library. The first query we implemented concerns finding of the top 10 regions in Greece with the highest energy demand levels with regards to household heating based on 2015 [data](#).

Household energy demand for heating, 2015, top 10 regions in Greece:

- Δ. ΑΘΗΝΑΙΩΝ
- Δ. ΘΕΣΣΑΛΟΝΙΚΗΣ
- Δ. ΠΕΙΡΑΙΩΣ
- Δ. ΠΑΤΡΕΩΝ
- Δ. ΠΕΡΙΣΤΕΡΙΟΥ
- Δ. ΗΡΑΚΛΕΙΟΥ
- Δ. ΛΑΡΙΣΗΣ
- Δ. ΚΑΛΛΙΘΕΑΣ
- Δ. ΝΙΚΑΙΑΣ
- Δ. ΚΑΛΑΜΑΡΙΑΣ

Figure 19: Greece, Top 10 regions in heating energy demand 2015 table

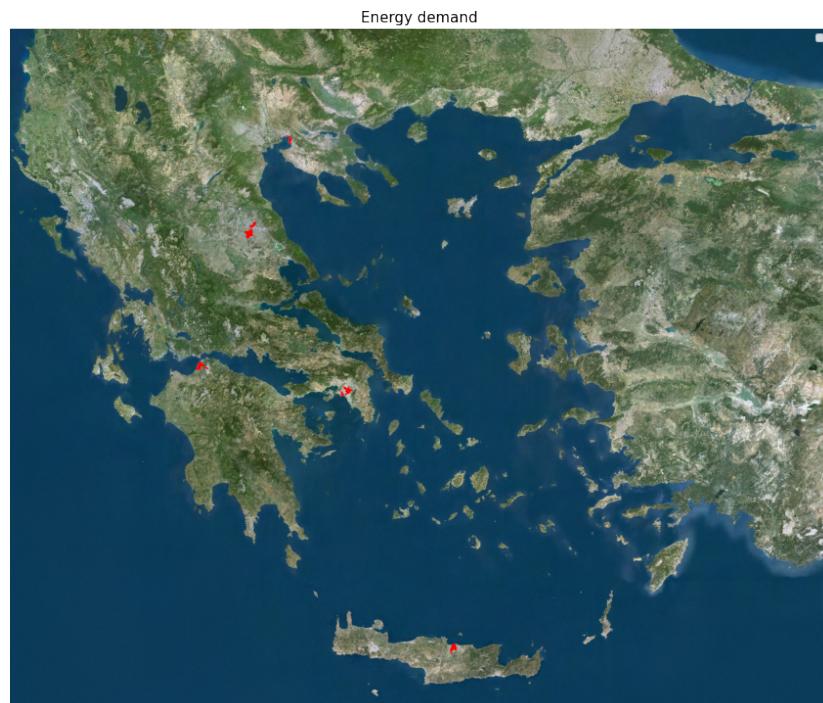


Figure 20: Greece, RGB raster top 10 regions in heating energy demand 2015

Up next, we calculate the municipalities of Greece that contain at least one airport. Below, we present the top 5 municipalities in terms of the number of airports as well as a map with the municipalities that contain airports in their territories.

NAME	
Θέρμης	3
Αγρινίου	2
Χίου	2
Λέσβου	2
Ρόδου	2

Figure 21: Greece, Top 5 municipalities having most airports in their territories

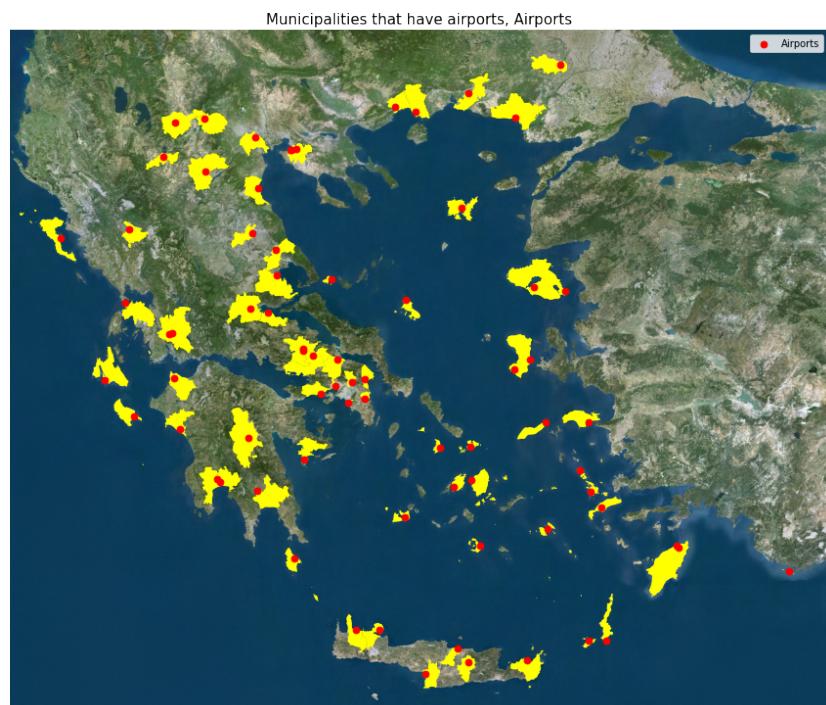


Figure 22: Greece, RGB raster municipalities containing airports

For the next query, we further narrow down the raster image to present Athens in the raster image as a basemap.



Figure 23: Athens, RGB raster

A first layer we can add is one containing markers for the public transport's stations in Athens. There is a total of 7935 stations in Athens.

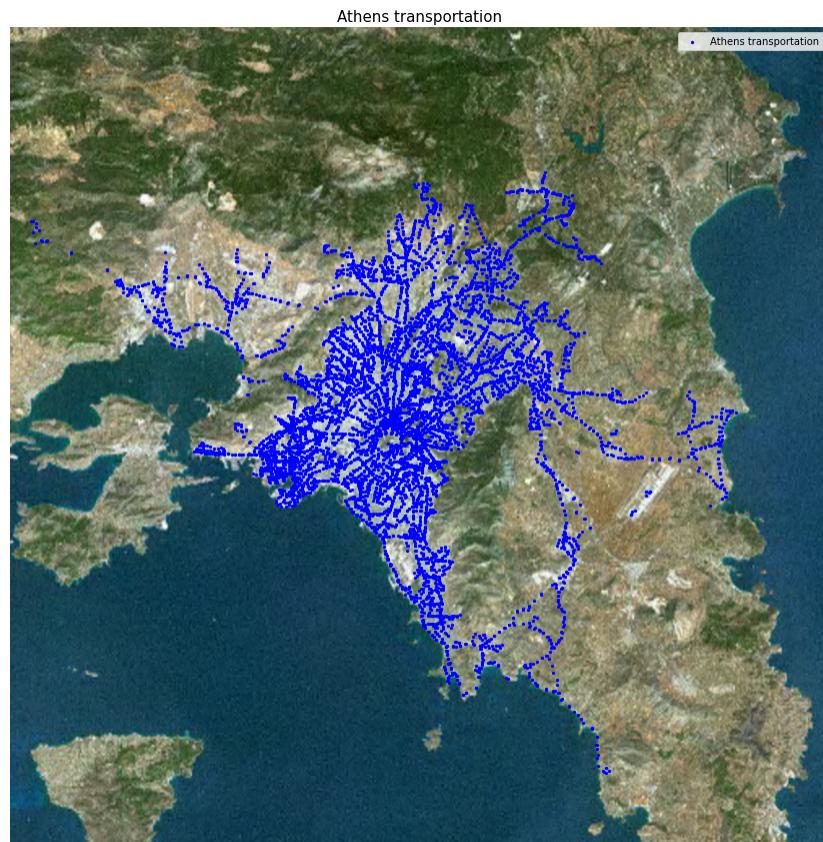


Figure 24: Athens, RGB raster public transport stations

We now distinguish the means of transport (ie bus-trolley, suburban railway, metro-isap, metro-suburban railway, tram) that the stations benefit and present them as different layers on the same raster basemap of Athens. It becomes obvious from the plot, that the majority of stations are bus-trolley stations.

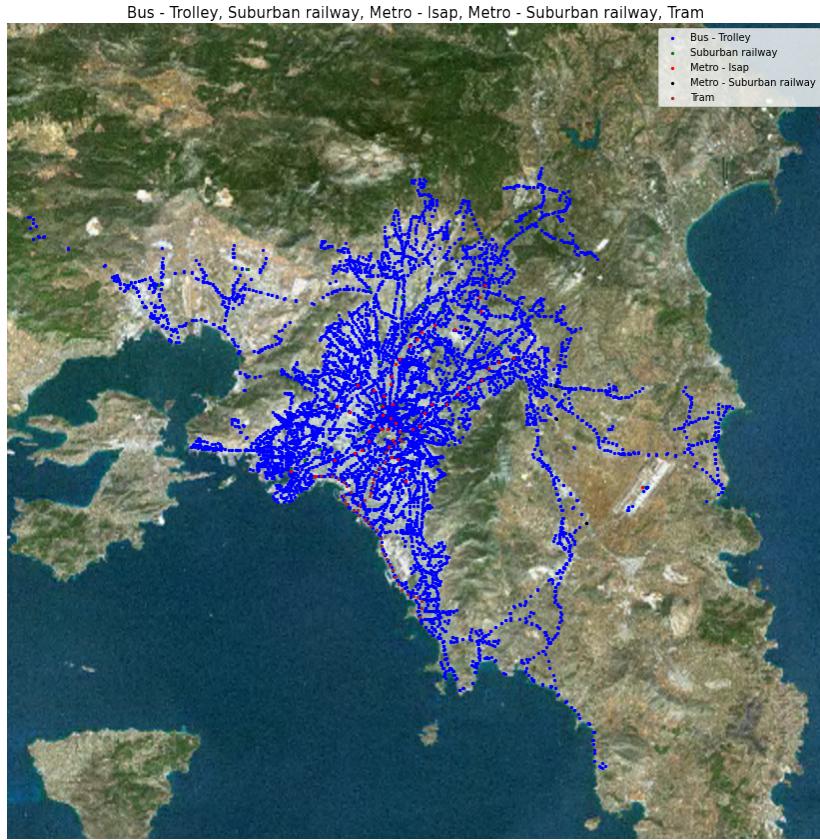


Figure 25: Athens, RGB raster public transport stations by means of transport

3.3 Question 3.3

In this section, we now load Corine Land Cover (CLC) 2018 data for a selected ROI from a WFS service. As described by [Copernicus](#), it is one of the most widely used products from the Copernicus Land Monitoring Service that consists of an inventory of land cover in 44 classes.

We now select Belgium as the region of interest (ROI). Searching in [Geoseer](#), we came up with the [wfs service](#) to use for retrieving the CLC 2018 data. Within the data downloaded, the land cover class is given under `lcclass` variable as a 3-digit integer and there are 44 discrete numbers for a matching number of classes. A mapping of the 3-digit classes encoding to their label/usage is provided [here](#).

Also, we located a [wms service](#) again through [Geoseer's](#) search in order to retrieve Belgium's raster landcover image.

In order to convert the 3-digit numbers of the vector data, we download an [xlsx file](#) that contains mappings of the 3-digit codes to the string labels with the land cover usage. From this exact same file we also extract the RGB hex code for each code. This is the RGB color that is assigned to each pixel based on its 3-digit code characterization of its land usage.

As a next step, the European raster image is clipped in order to retrieve an RGB raster basemap of Belgium.

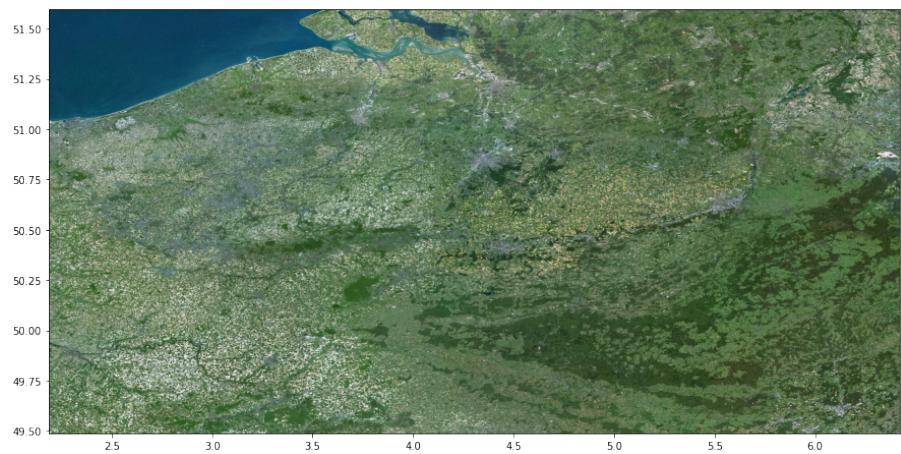


Figure 26: Belgium, RGB raster

Then, we add the wms CLC layer on top and we receive the following image.

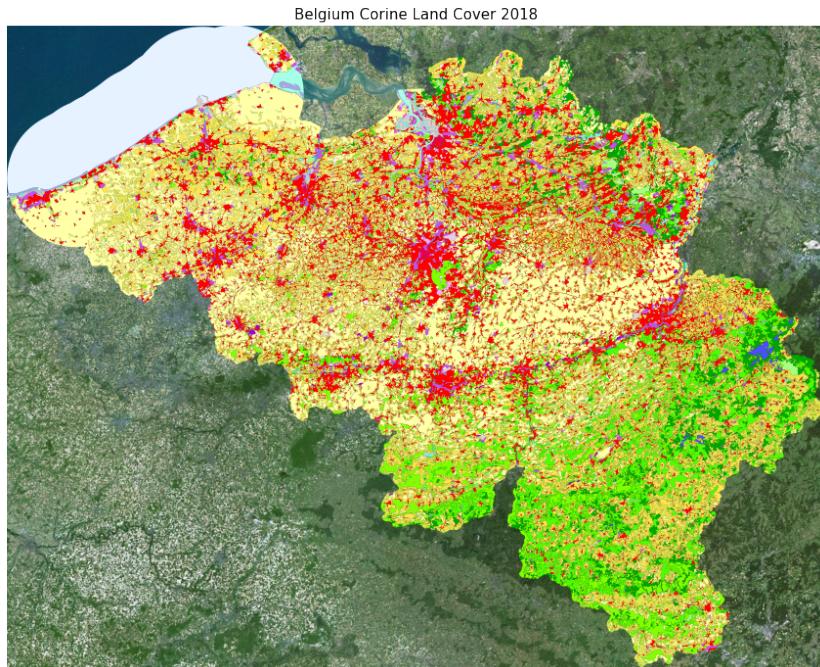


Figure 27: Belgium, RGB raster CLC 2018

3.4 Question 3.4

Up next is the retrieval and presentation of meteorological time series data for an area of interest. We now select a point in the center of Ghent, one of the major cities of Belgium (north Belgium). We select the exact point and coordinates using [geojson](#) and download the generated shapefile. In the image below, we can see the selected point in geojson's user web interface.

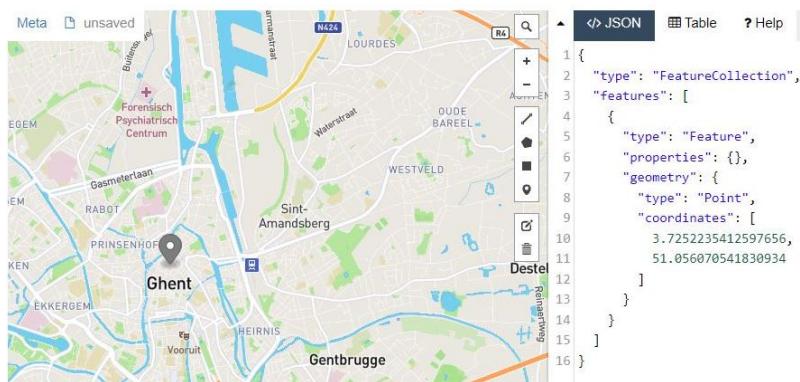


Figure 28: Ghent, geojson ROI

We then download meteorological data from [NASA](#) for the ROI (point/marker in Ghent) and for the period between 01-01-2020 - 01-01-2022 and more specifically the following data:

- Temperature at 2 Meters (C)
- Relative Humidity at 2 Meters (%)
- Precipitation (Corrected, mm/day)
- Wind Speed at 10 Meters (m/s)

We then proceed to generating the plot of the temperature at 2 meters for the point of interest over time.

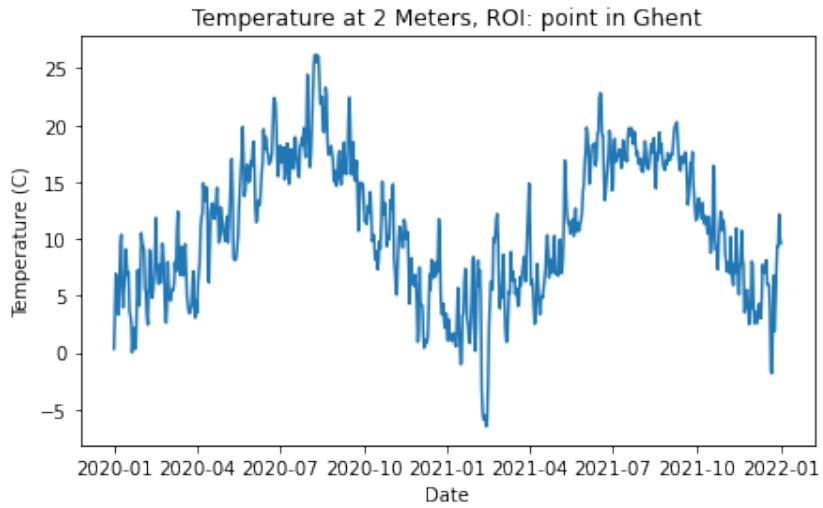


Figure 29: Ghent, ROI temperature at 2 meters timeseries

We can see there is periodicity on the temperature levels for the selected point, peaking during summer months (July-August) as one would expect.

Next, we plot the timeseries for the wind speed at 10 meters in m/s. We can observe that windspeed, unlike temperature, exhibits very quick changes in its value within short time periods. However, no clear pattern can be detected apart from the observation that during some periods smaller peaks are detected.

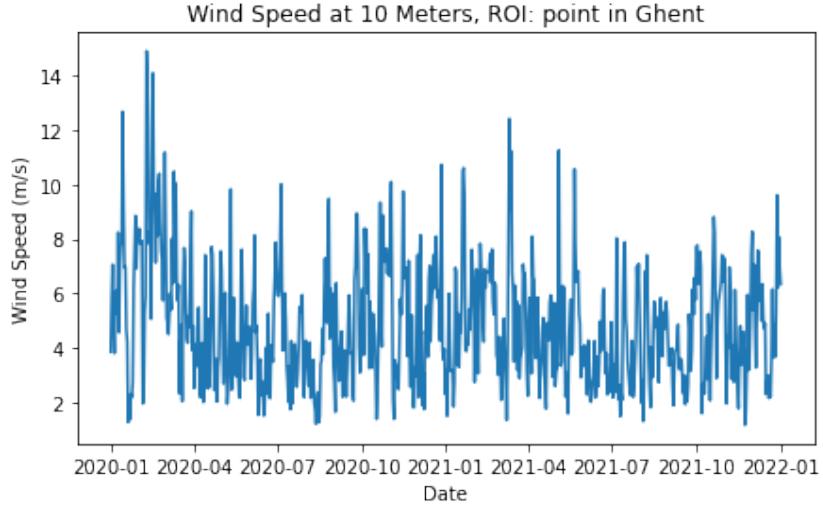


Figure 30: Ghent, ROI wind speed at 10 meters timeseries

Below, we additionally plot the timeseries for the relative humidity percentage for the ROI. Here, we can see there is some periodicity on the humidity levels where the high peaks are observed during winter time and the low peaks in summer time.

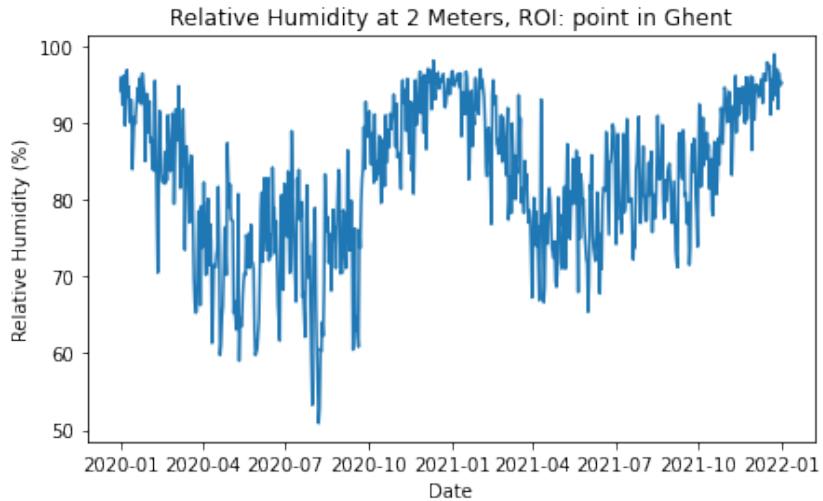


Figure 31: Ghent, ROI relative humidity at 2 meters timeseries

Lastly, we present the plot for the corrected precipitation (mm/day) time-series, where we can see that there is a continuous of days with relatively high precipitation values during winter months, however this is not the case in spring/summer months.

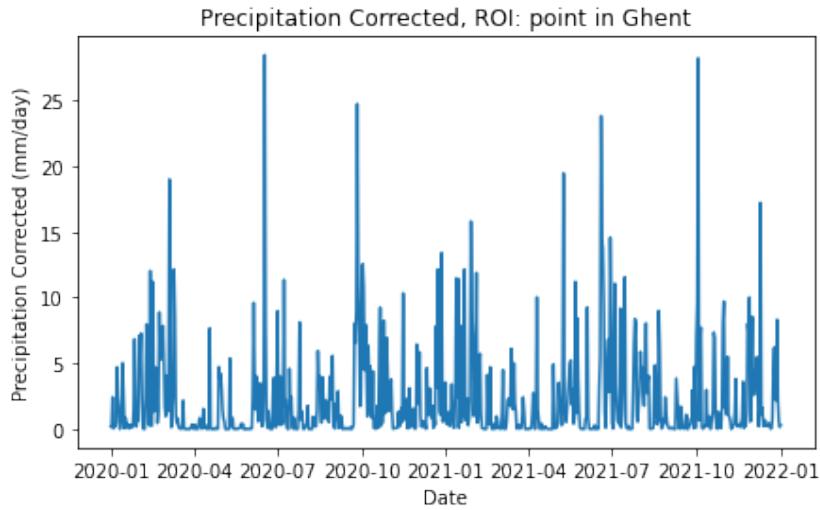


Figure 32: Ghent, ROI precipitation (corrected) timeseries

4 Question 4

In this final part of the assignment, we present an interactive map of Greece using folium and combining raster and vector data of our choice.

We select to present a raster image as a layer on the map on top of a street map basemap. The raster image is a European one and centered to Greece. Then, we add vector data as layers too. The interactive map is available also in html format as a standalone file apart from the *ipynb* file. The file can be viewed on any browser and users can interact with the map.

All the layers presented on the map are listed below:

1. Raster:
 - (a) european raster image centered to Greece
2. Vector:
 - (a) Markers:
 - i. Airports
 - ii. National Parks average centroids
 - iii. Lakes average centroids
 - iv. Wifi public spots
 - v. Beaches awarded with blue flag
 - vi. Dams
 - (b) Polygons:
 - i. National Parks
 - ii. Lakes



Figure 33: Greece, interactive map snippet

The markers on the map are straightforward to implement, since the vector data such as the airports contain Point geometries that upon being converted to the right crs can be easily added to a Feature Group (eg airports feature group) and then the feature group is added as a layer to the map.

However, for polygons such as national parks and lakes, there are several entries in the relevant geopandas dataframe containing the vector data for the same item (eg. several rows in the dataframe for a single national park). This means that taking the centroids of the polygons and adding them to the map would end up in several markers for each national park. In order to add a single marker for those instead, we then proceed to calculating the centroids of the polygon geometries for each entry and then aggregate the centroids for each national park and lake by taking the average coordinate of those centroids. This results in two layers for national parks and lakes, one for the polygons of the parks and lakes, and another with those centroids denoting the center of each lake and park.

References

- [1] OWSLib <https://www.osgeo.org/projects/owslib>
- [2] rasterio <https://rasterio.readthedocs.io/en/latest/>

- [3] geopandas <https://geopandas.org/>
- [4] folium <https://python-visualization.github.io/folium/>
- [5] leafmap <https://leafmap.org/>
- [6] fiona <https://fiona.readthedocs.io/>