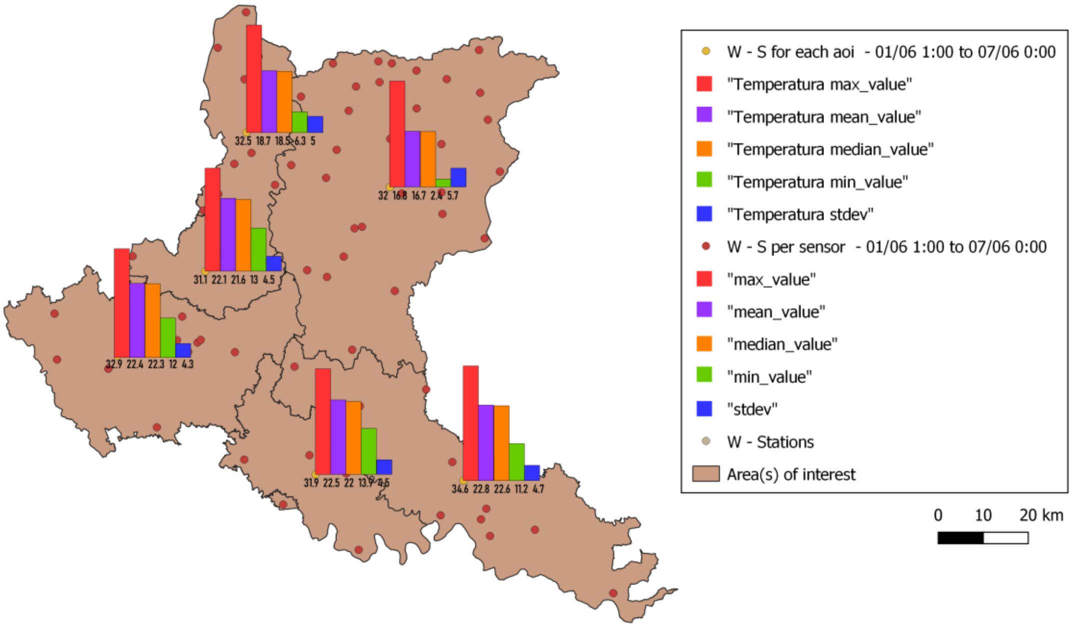
GEOINFORMATICS PROJECT

QGIS ARPA Lombardia Data Extractor



Author: Mathilde Puche

Project teacher: Maria Antonella Brovelli

Project tutor: Vasil Yordanov

GitHub repository: https://github.com/MatPuche/arpa_data

June 2021

Table of contents

I. Description of the project

The ARPA ground sensors network collects data regarding weather, air and water quality, as well as waste management in Lombardy, Italy. The aim of the project was to develop a QGIS Processing Toolbox module enabling to extract these data using the API of ARPA Lombardia and Python scripting.

As a first step, the possible data to extract and their properties (amount of information, measurement frequency, attributes...) were identified and analysed. Then, the plugin has been developed to allow the insertion of these data into QGIS. According to the previously identified properties, some filters have been added to enable the user to insert some parameters (as time period or sensor location) before requesting the data. Such a plugin is made to produce understandable maps from ARPA Lombardia data and to enable one user to manipulate them into QGIS.

II. Data analysis

The first step of the project was to analyse the data available through the ARPA API. Starting from the list of published datasets found on the Lombardy Open Data Portal, we had to inspect the 21 API links in order to determine the content of each dataset and its usability.

This first inspection led us to several observations. First, there are datasets of five different topics regarding Lombardy region: weather, air quality, flow data, water quality and waste collection. However, we noticed that only the weather and air quality datasets are providing data of a periodical monitoring. For this reason, we decided to focus on these two.

A deeper exploration brought a second discovery: the API for these two datasets provides only a subset of the data collected. Indeed, it provides:

- for the weather: the monitoring data of the current month, with a frequency of 10min.
- for the air quality: the monitoring data of the current year, with a frequency of 1hour.

The rest of the data is divided into CSV files per year and can only be downloaded manually. As we wanted to request our data directly from the plugin and not download it to the PC, we decided here again to focus only on the data available via the API.

Also, the dataset for weather monitoring and the one for air quality only provide the collected values, the date of measurement and the id of each sensor and don't contain any information on the location of the station and the type of sensor. Therefore, two other datasets had to be considered: the ones containing all the information of the monitoring stations for weather and air quality.

So, finally, the datasets we chose to focus on are as follows:

- Weather stations information (called "*Stazioni Meteorologiche*")
- Weather sensors monitoring data (called "*Dati sensori meteo*")
- Air quality stations information (called "*Stazioni qualità dell'aria*")
- Air quality sensors monitoring data (called "*Dati sensori aria*")

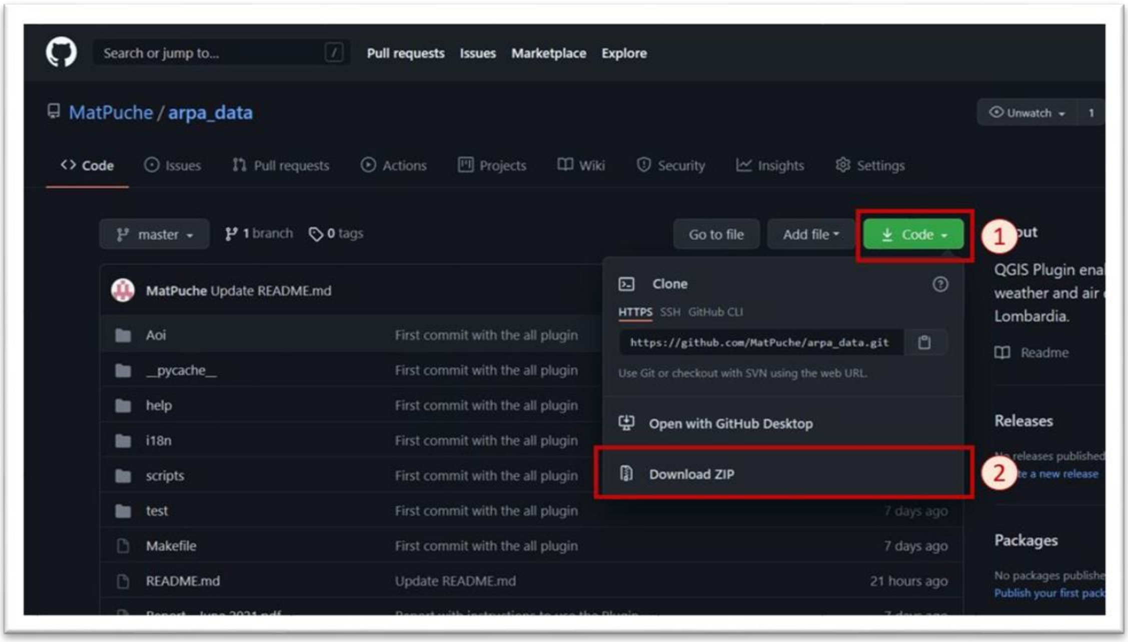
III. The plugin

After this initial analysis of the data, the implementation of the plugin could begin. In the following, a presentation of its general functioning is made, followed by the different stages of its implementation.

a. Installation

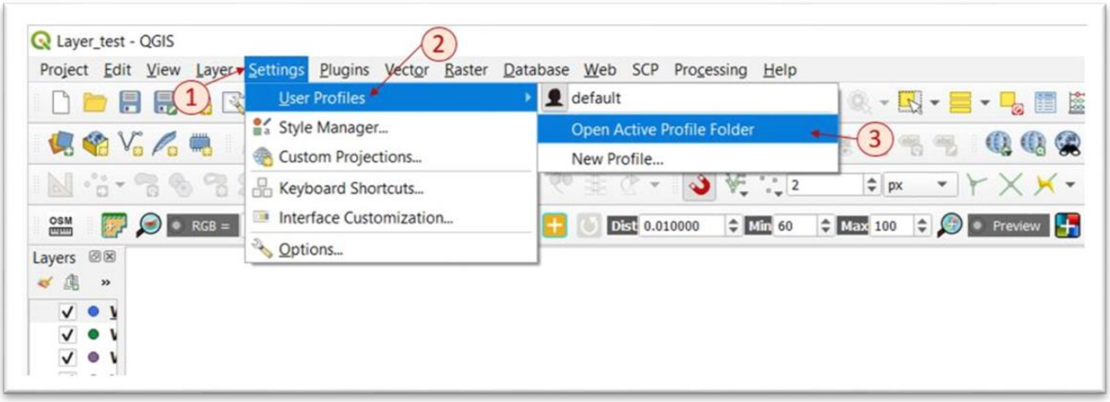
The plugin is called "ARPA data” and needs to be installed manually. Plugins in QGIS are stored in a special folder. We must copy the plugin directory to that folder before it can be used.

1. Download the entire directory from GitHub (https://github.com/MatPuche/arpa_data) and unzip it.



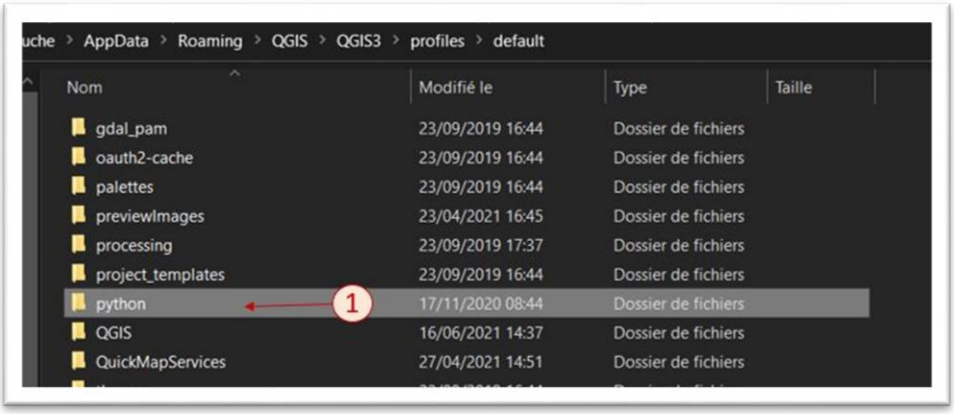
Download the zip file from GitHub

2. In QGIS, locate your current profile folder by going to *Settings* ‣ *User Profiles* ‣ *Open Active Profile Folder.*



Locate your current profile folder on QGIS

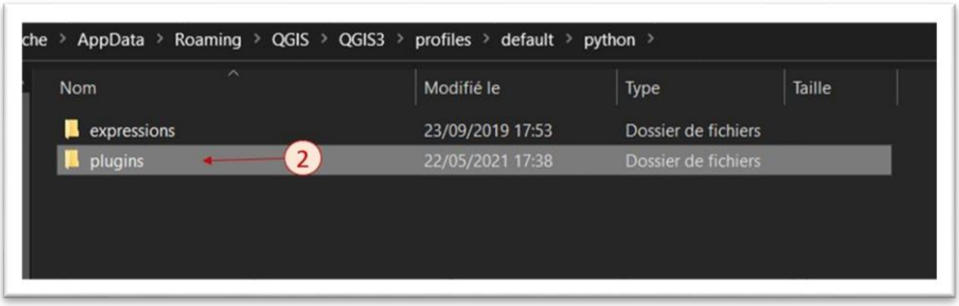
3. Copy the plugin folder previously downloaded to *python* ‣ *plugins* subfolder.



uche › AppData › Roaming › QGIS › QGIS3 › profiles › default

Nom	Modifié le	Type	Taille
gdal_pam	23/09/2019 16:44	Dossier de fichiers	
oauth2-cache	23/09/2019 16:44	Dossier de fichiers	
palettes	23/09/2019 16:44	Dossier de fichiers	
previewImages	23/04/2021 16:45	Dossier de fichiers	
processing	23/09/2019 17:37	Dossier de fichiers	
project_templates	23/09/2019 16:44	Dossier de fichiers	
python	17/11/2020 08:44	Dossier de fichiers	
QGIS	16/06/2021 14:37	Dossier de fichiers	
QuickMapServices	27/04/2021 14:51	Dossier de fichiers	

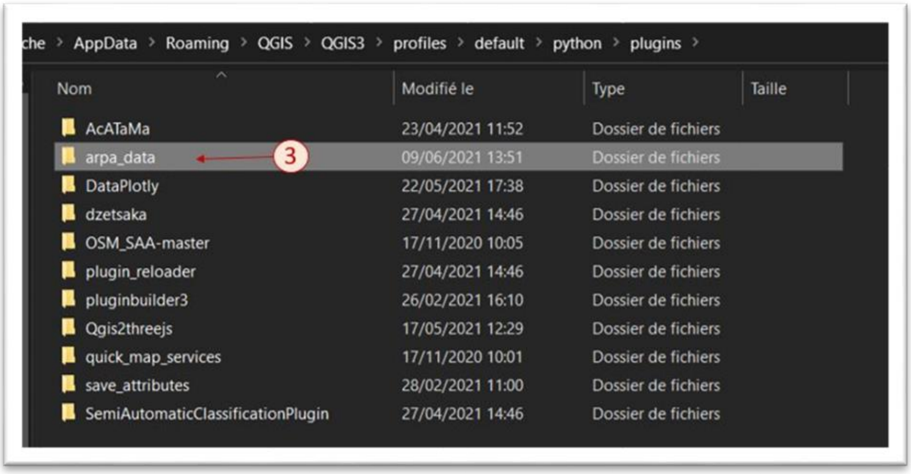
Enter into python subfolder



che › AppData › Roaming › QGIS › QGIS3 › profiles › default › python ›

Nom	Modifié le	Type	Taille
expressions	23/09/2019 17:53	Dossier de fichiers	
plugins	22/05/2021 17:38	Dossier de fichiers	

Enter into plugins subfolder



che › AppData › Roaming › QGIS › QGIS3 › profiles › default › python › plugins ›

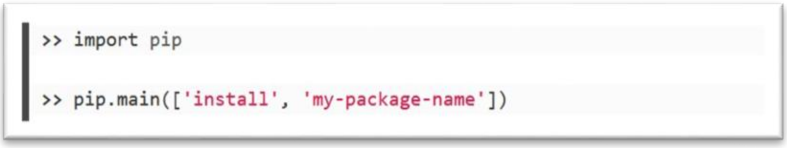
Nom	Modifié le	Type	Taille
AcATaMa	23/04/2021 11:52	Dossier de fichiers	
arpa_data	09/06/2021 13:51	Dossier de fichiers	
DataPlotly	22/05/2021 17:38	Dossier de fichiers	
dzetsaka	27/04/2021 14:46	Dossier de fichiers	
OSM_SAA-master	17/11/2020 10:05	Dossier de fichiers	
plugin_reloader	27/04/2021 14:46	Dossier de fichiers	
pluginbuilder3	26/02/2021 16:10	Dossier de fichiers	
Qgis2threejs	17/05/2021 12:29	Dossier de fichiers	
quick_map_services	17/11/2020 10:01	Dossier de fichiers	
save_attributes	28/02/2021 11:00	Dossier de fichiers	
SemiAutomaticClassificationPlugin	27/04/2021 14:46	Dossier de fichiers	

Paste the plugin directory

4. Then, the "*sodapy*" Python library must be installed to make the plugin work. This can be done in two different ways: inside QGIS or from the command line.

i. Inside QGIS

Open QGIS Python console (under Plugins » Python Console) and type the following lines (with ‘my-package name’ replaced by ’sodapy’).



```
>> import pip

>> pip.main(['install', 'my-package-name'])
```

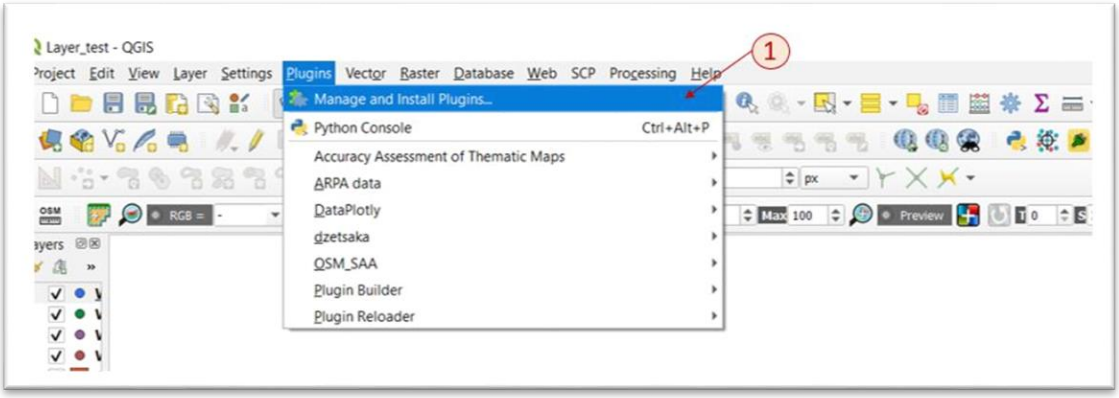
ii. From the command line

On Linux systems, QGIS use the main Python installation, so the only thing to do is to run `pip install sodapy` (pip must be installed) in the command line.

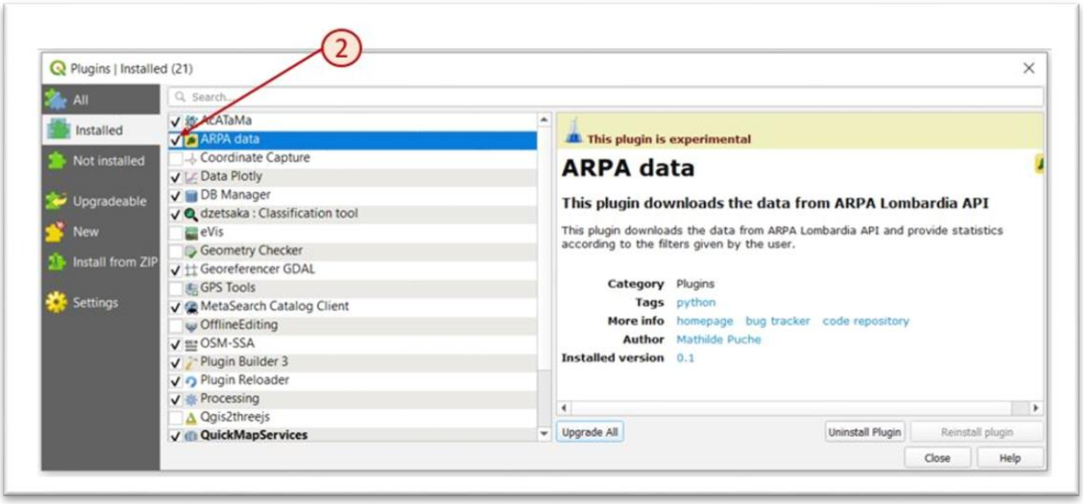
On Windows, QGIS has its own Python, so the library must be installed in the right one. Navigate to `C:\QGIS\apps\Python27\` or `C:\QGIS\apps\Python37`. Open command prompt or powershell here and type: `python -m pip install sodapy`.

More information can be found on the GitHub repository of sodapy: https://github.com/xmunoz/sodapy

5. Finally, the plugin should appear in the Plugins management box of QGIS, in installed plugins and only needs to be activated as a usual plugin.


Layer_test - QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web SCP Processing Help
Manage and Install Plugins...
Python Console Ctrl+Alt+P
Accuracy Assessment of Thematic Maps
ARPA data
DataPlotly
dzetsaka
OSM_SAA
Plugin Builder
Plugin Reloader
1

Go to the Plugins management box in QGIS



Plugins | Installed (21)

- ☑ AcATaMa
- ☑ ARPA data
- □ Coordinate Capture
- ☑ Data Plotly
- ☑ DB Manager
- ☑ dzetsaka : Classification tool
- □ eVis
- □ Geometry Checker
- ☑ Georeferencer GDAL
- □ GPS Tools
- ☑ MetaSearch Catalog Client
- □ OfflineEditing
- ☑ OSM-SSA
- ☑ Plugin Builder 3
- ☑ Plugin Reloader
- ☑ Processing
- □ Qgis2threejs
- ☑ **QuickMapServices**

This plugin is experimental

ARPA data

This plugin downloads the data from ARPA Lombardia API

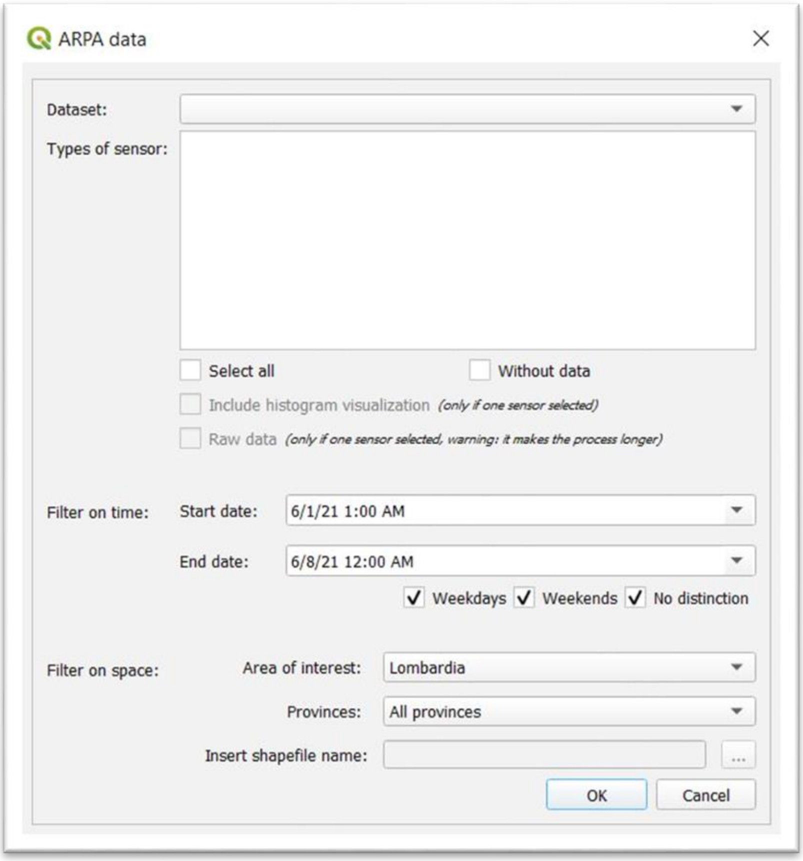
This plugin downloads the data from ARPA Lombardia API and provide statistics according to the filters given by the user.

Category	Plugins
Tags	python
More info	homepage bug tracker code repository
Author	Mathilde Puche
Installed version	0.1

Activate the Arpa plugin in the installed plugins

b. General functioning of the plugin

Once installed, the plugin is present in the menu “Plugin” of QGIS. When the user launches it, a dialog window opens (Fig.1), and a set of parameters must be provided.



ARPA data

Dataset:

Types of sensor:

□ Select all □ Without data

□ Include histogram visualization *(only if one sensor selected)*

□ Raw data *(only if one sensor selected, warning: it makes the process longer)*

Filter on time: Start date: 6/1/21 1:00 AM

End date: 6/8/21 12:00 AM

☑ Weekdays ☑ Weekends ☑ No distinction

Filter on space: Area of interest: Lombardia

Provinces: All provinces

Insert shapefile name:

Figure 1 – Entire window when the plugin starts

In the following, all the parameters and their meaning are detailed.

1. DATA FILTERS

   i. **The user chooses the dataset**: Weather or Air quality (1)

   ii. **The user chooses the type of sensors**: *(multiple selection)* (2)

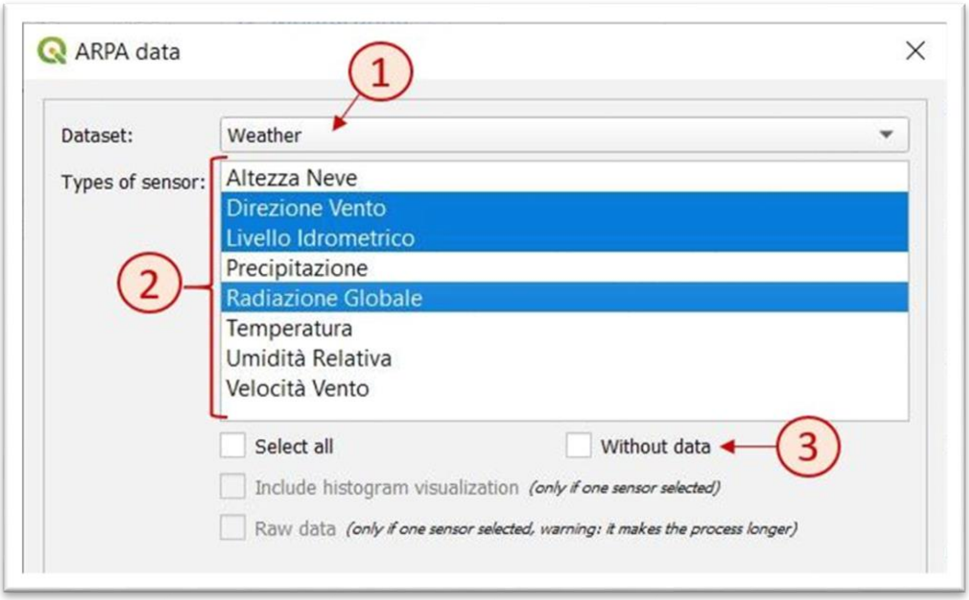
      a. <u>For Weather data:</u>
      Altezza Neve, Direzione Vento, Livello Idrometrico, Precipitazione, Radiazione Globale, Temperatura, Umidità Relativa, Velocità Vento.

      b. <u>For Air quality data:</u>
      Ammoniaca, Arsenico, Benzene, Benzo(a)pirene, Biossido di Azoto, Biossido di Zolfo, BlackCarbon, Cadmio, Monossido di Azoto, Monossido di Carbonio, Nikel, Ossidi di Azoto, Ozono, Particelle sospese PM2.5, Particolato Totale Sospeso, Piombo, PM10, PM10 (SM2005).

Note: If the user selects "No data" , the result is only the stations locations without any monitoring data. (3)

If only one type of sensor is selected the user has 2 more choices:

- <u>Include the visualization</u>: to have the final map with a visualization of the statistics per area of interest by means of histograms (4)
- <u>Raw data</u>: to get, as an additional layer, the raw data from the datasets. **<u>This option can greatly increase the data processing time</u>**, so it is only allowed when only one sensor type is selected. (5)



ARPA data

Dataset: Weather

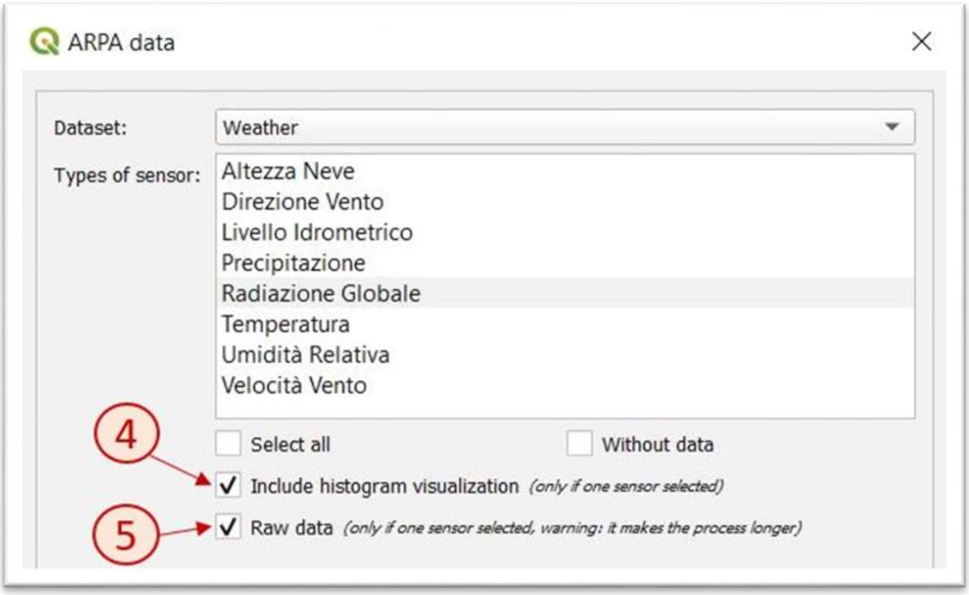
Types of sensor:
Altezza Neve
Direzione Vento
Livello Idrometrico
Precipitazione
Radiazione Globale
Temperatura
Umidità Relativa
Velocità Vento

□ Select all □ Without data

□ Include histogram visualization *(only if one sensor selected)*

□ Raw data *(only if one sensor selected, warning: it makes the process longer)*

Figure 2 – Data filters (several types of sensors selected)



ARPA data

Dataset: Weather

Types of sensor:
Altezza Neve
Direzione Vento
Livello Idrometrico
Precipitazione
Radiazione Globale
Temperatura
Umidità Relativa
Velocità Vento

□ Select all □ Without data

☑ Include histogram visualization *(only if one sensor selected)*

☑ Raw data *(only if one sensor selected, warning: it makes the process longer)*

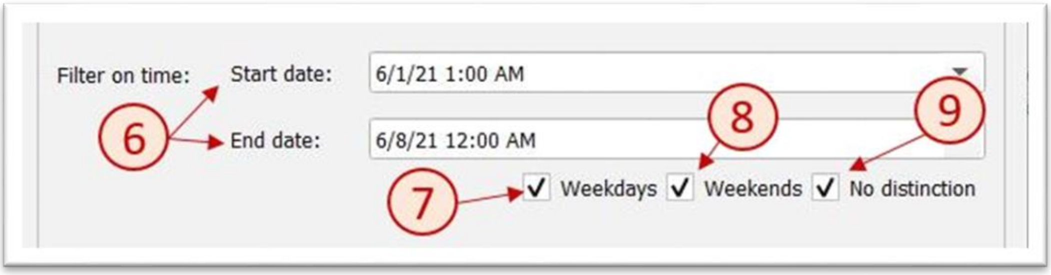
Figure 3 – Data filters (one type of sensors selected)

2. TIME FILTERS

   i. **The user chooses the dates** (start date and end date of the data collection) (6)

   ii. **The user chooses the day type**:

      a. Weekdays, (7)
      b. Weekends, (8)
      c. Both weekdays and weekends:
         - Without distinction: the statistics are computed on all selected days without distinction. In case, the raw data are asked, they are provided without distinction as well (Fig. 4, (9)).
         - With distinction: the statistics are computed for weekdays and for weekends with distinction. One result for each is provided. In case, the raw data are asked, they are provided with a distinction as well (Fig. 5, (9)).

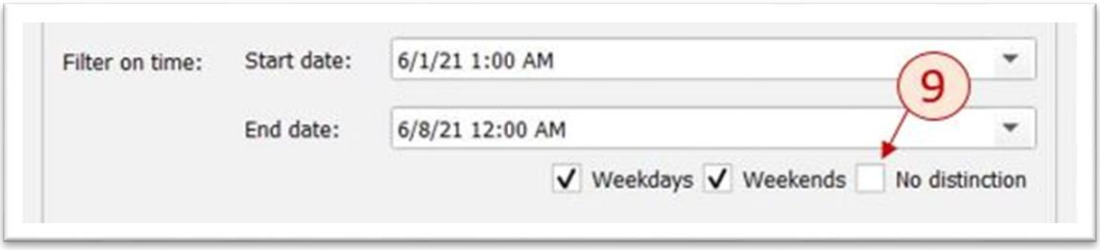


Filter on time: Start date: 6/1/21 1:00 AM

End date: 6/8/21 12:00 AM

☑ Weekdays ☑ Weekends ☑ No distinction

Figure 4 – Time filters (without distinction between weekdays and weekends)


Filter on time: Start date: 6/1/21 1:00 AM

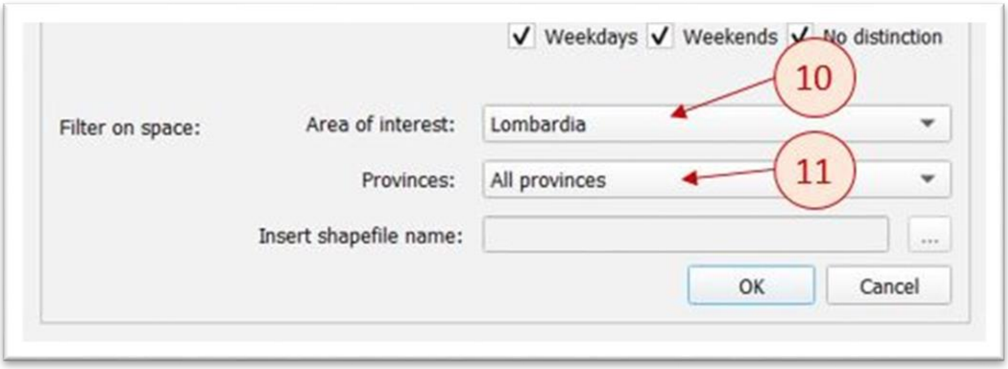
End date: 6/8/21 12:00 AM

☑ Weekdays ☑ Weekends □ No distinction

Figure 5 – Time filters (with distinction between weekdays and weekends)

3. SPACE FILTERS
   i. **The user selects the area(s) of interest**:
      a. Lombardia (all provinces or select one in particular) (10 and 11)
      b. Insert an area: browse a shapefile on his/her local device. (12 and 13) */!\ The shapefile must contain polygons! These polygons will be the areas of interest and the statistics are computed for each of the polygons. /!\ The CRS of the inserted shapefile must be WGS84!*

The selected areas of interest will be added to the resulted layers.

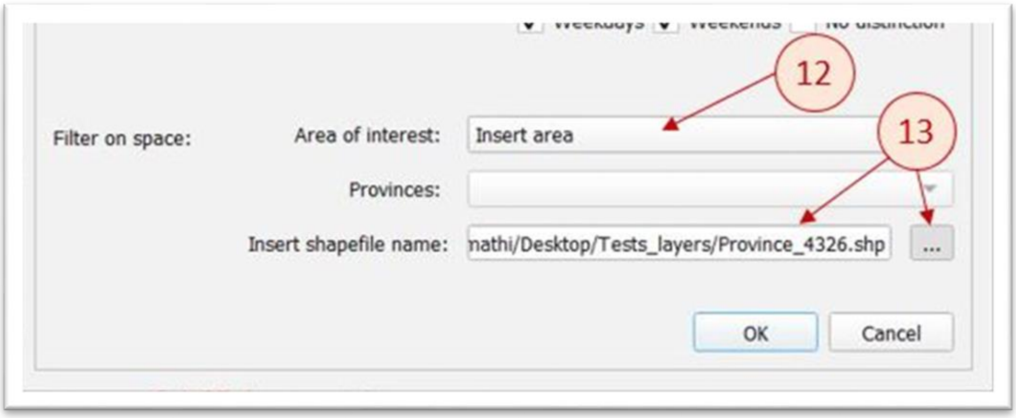

☑ Weekdays ☑ Weekends ☑ No distinction

Filter on space: Area of interest: Lombardia

Provinces: All provinces

Insert shapefile name:

Figure 6 – Space filters (Lombardy with all provinces)


Filter on space: Area of interest: Insert area

Provinces:

Insert shapefile name: nathi/Desktop/Tests_layers/Province_4326.shp

Figure 7 – Space filters (Inserted shapefile)

c. Results

The user presses OK and, after the processing, the maps are generated and added to the QGIS project.

1. Resulted maps

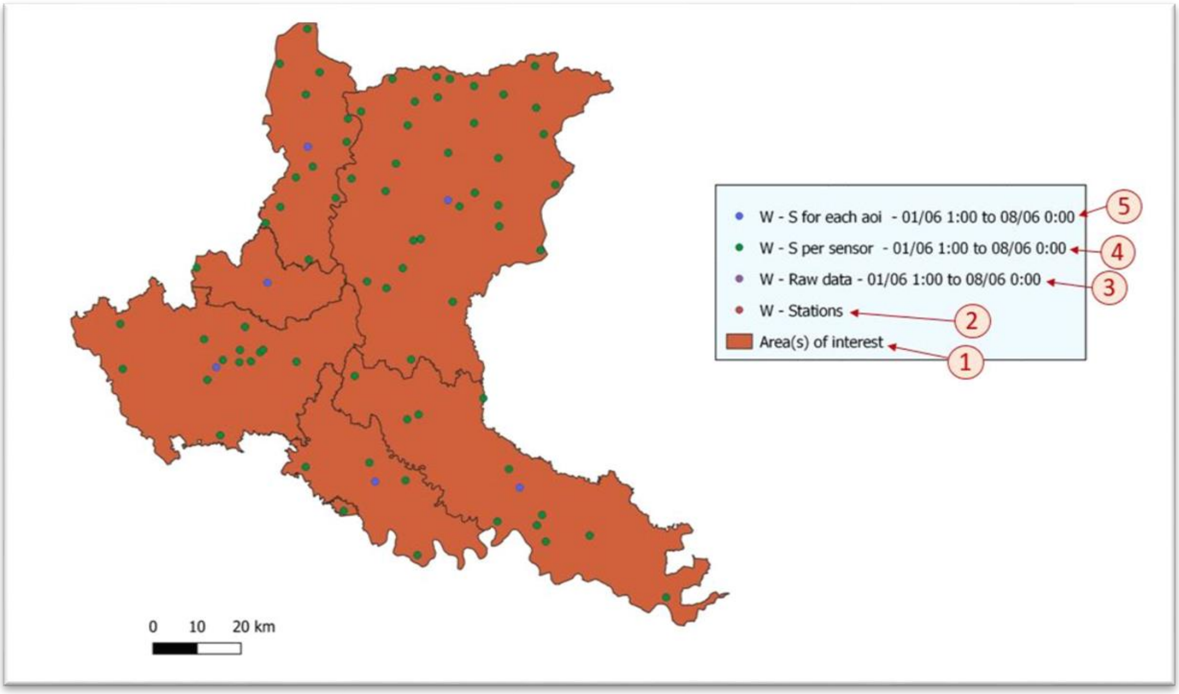
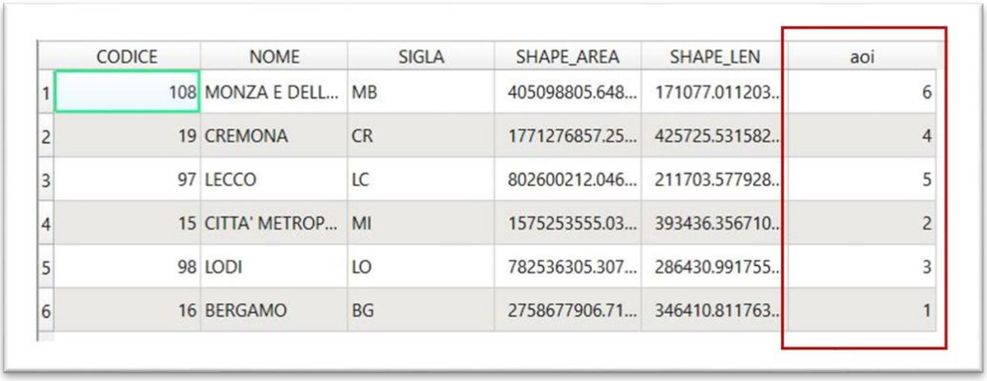

● W - S for each aoi - 01/06 1:00 to 08/06 0:00
● W - S per sensor - 01/06 1:00 to 08/06 0:00
● W - Raw data - 01/06 1:00 to 08/06 0:00
● W - Stations
▨ Area(s) of interest
5
4
3
2
1
0 10 20 km

Figure 8 – Resulted layers, all together

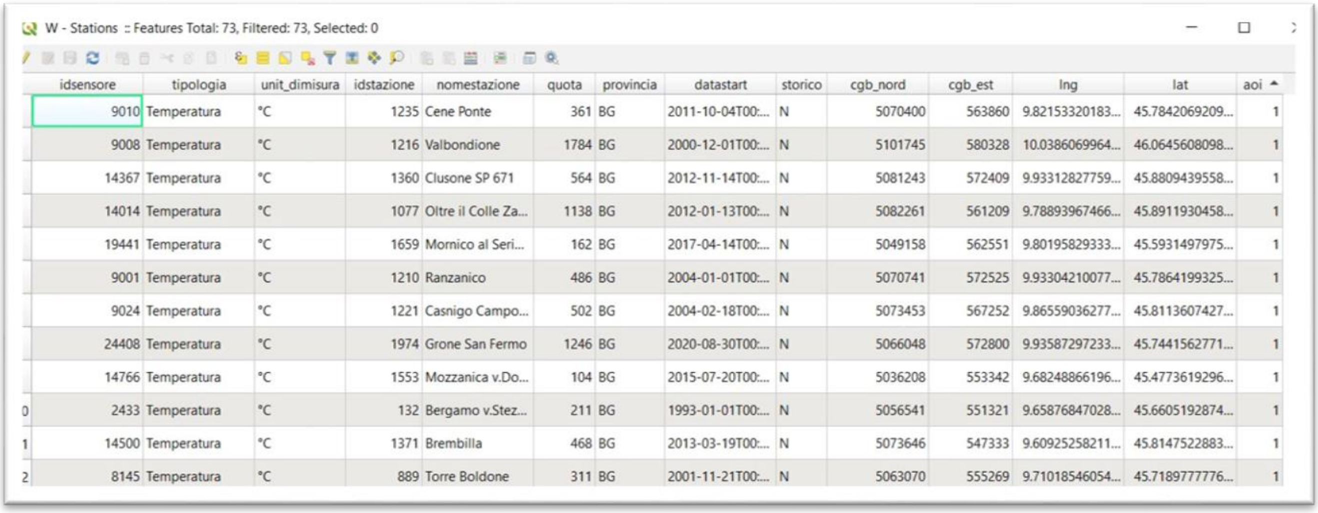
i. **One map of polygons representing the areas of interest**. (Fig. 8 (1)) If the user has entered his own file, it is the same layer. If the user has selected Lombardia or a particular province, the corresponding vector layer is added from a shapefile contained in the plugin directory. Each area of interest contains an identifier in the attribute named "aoi" which is found in all following layers. *Named "Area(s) of interest"*



	CODICE	NOME	SIGLA	SHAPE_AREA	SHAPE_LEN	aoi
1	108	MONZA E DELL...	MB	405098805.648...	171077.011203..	6
2	19	CREMONA	CR	1771276857.25...	425725.531582..	4
3	97	LECCO	LC	802600212.046...	211703.577928..	5
4	15	CITTA' METROP...	MI	1575253555.03...	393436.356710..	2
5	98	LODI	LO	782536305.307...	286430.991755..	3
6	16	BERGAMO	BG	2758677906.71...	346410.811763..	1

Figure 9 – Attribute table of Area(s) of interest layer

ii. **One map of points representing the stations concerned by the filters** (Fig. 8 (2)) and containing some metadata (name and id of the station, location, type of sensor, commune, province, start time of the station...)
Named "W - Stations" or "AQ - Stations"

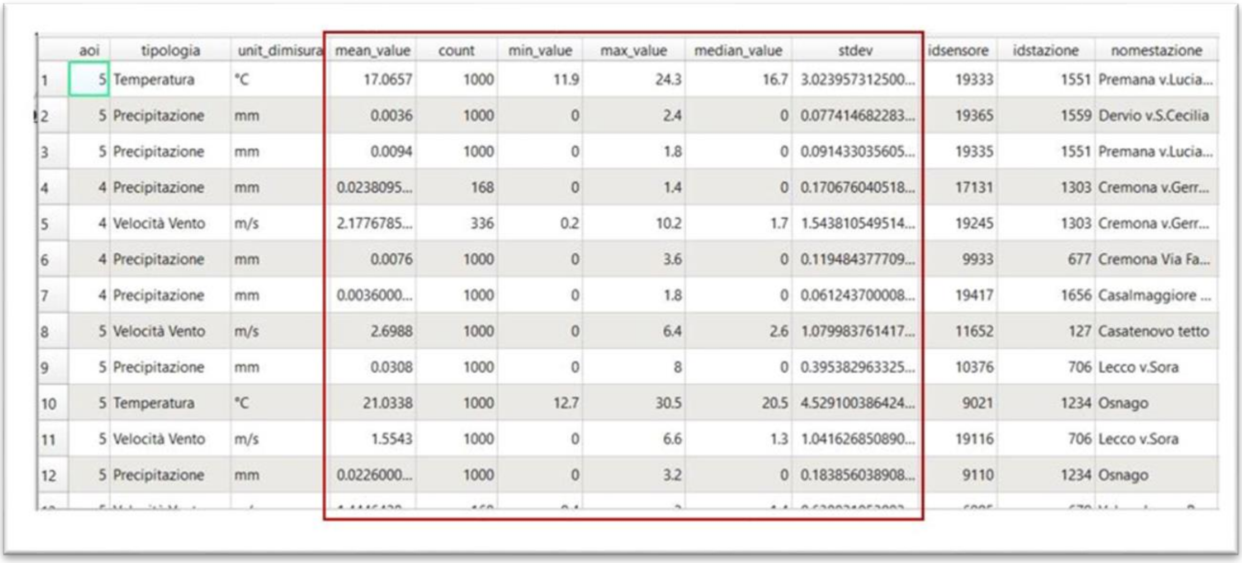


W - Stations :: Features Total: 73, Filtered: 73, Selected: 0

	idsensore	tipologia	unit_dimisura	idstazione	nomestazione	quota	provincia	datastart	storico	cgb_nord	cgb_est	lng	lat	aoi
	9010	Temperatura	°C	1235	Cene Ponte	361	BG	2011-10-04T00:...	N	5070400	563860	9.82153320183...	45.7842069209...	1
	9008	Temperatura	°C	1216	Valbondione	1784	BG	2000-12-01T00:...	N	5101745	580328	10.0386069964...	46.0645608098...	1
	14367	Temperatura	°C	1360	Clusone SP 671	564	BG	2012-11-14T00:...	N	5081243	572409	9.93312827759...	45.8809439558...	1
	14014	Temperatura	°C	1077	Oltre il Colle Za...	1138	BG	2012-01-13T00:...	N	5082261	561209	9.78893967466...	45.8911930458...	1
	19441	Temperatura	°C	1659	Mornico al Seri...	162	BG	2017-04-14T00:...	N	5049158	562551	9.80195829333...	45.5931497975...	1
	9001	Temperatura	°C	1210	Ranzanico	486	BG	2004-01-01T00:...	N	5070741	572525	9.93304210077...	45.7864199325...	1
	9024	Temperatura	°C	1221	Casnigo Campo...	502	BG	2004-02-18T00:...	N	5073453	567252	9.86559036277...	45.8113607427...	1
	24408	Temperatura	°C	1974	Grone San Fermo	1246	BG	2020-08-30T00:...	N	5066048	572800	9.93587297233...	45.7441562771...	1
	14766	Temperatura	°C	1553	Mozzanica v.Do...	104	BG	2015-07-20T00:...	N	5036208	553342	9.68248866196...	45.4773619296...	1
0	2433	Temperatura	°C	132	Bergamo v.Stez...	211	BG	1993-01-01T00:...	N	5056541	551321	9.65876847028...	45.6605192874...	1
1	14500	Temperatura	°C	1371	Brembilla	468	BG	2013-03-19T00:...	N	5073646	547333	9.60925258211...	45.8147522883...	1
2	8145	Temperatura	°C	889	Torre Boldone	311	BG	2001-11-21T00:...	N	5063070	555269	9.71018546054...	45.7189777776...	1

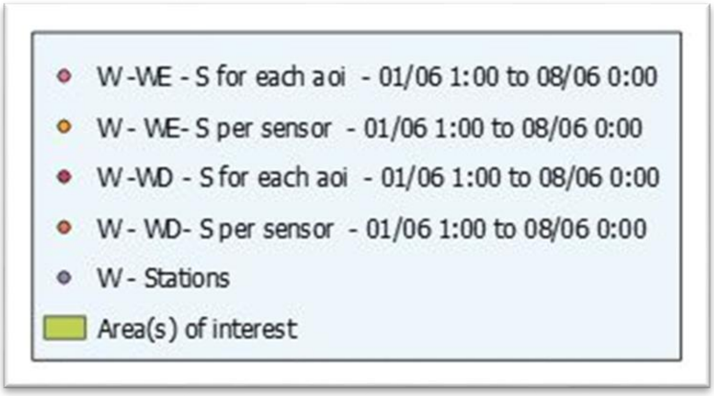
Figure 10 – Attribute table of "Stations" layer

iii. **One map of points representing the sensors concerned by the filters** (Fig. 8 (4)) **and containing the computed statistics for each sensor** (count, mean, min, max, median and standard deviation of the measurements). Here, if the user chose previously to have weekdays and weekends with a distinction, there are two different maps: one for weekdays and one for weekends (cf. Fig. 12).
Named "W - S per sensor - Dates" or "AQ - S per sensor - Dates "



	aoi	tipologia	unit_dimisura	mean_value	count	min_value	max_value	median_value	stdev	idsensore	idstazione	nomestazione
1	5	Temperatura	°C	17.0657	1000	11.9	24.3	16.7	3.023957312500...	19333	1551	Premana v.Lucia...
2	5	Precipitazione	mm	0.0036	1000	0	2.4	0	0.077414682283...	19365	1559	Dervio v.S.Cecilia
3	5	Precipitazione	mm	0.0094	1000	0	1.8	0	0.091433035605...	19335	1551	Premana v.Lucia...
4	4	Precipitazione	mm	0.0238095...	168	0	1.4	0	0.170676040518...	17131	1303	Cremona v.Gerr...
5	4	Velocità Vento	m/s	2.1776785...	336	0.2	10.2	1.7	1.543810549514...	19245	1303	Cremona v.Gerr...
6	4	Precipitazione	mm	0.0076	1000	0	3.6	0	0.119484377709...	9933	677	Cremona Via Fa...
7	4	Precipitazione	mm	0.0036000...	1000	0	1.8	0	0.061243700008...	19417	1656	Casalmaggiore ...
8	5	Velocità Vento	m/s	2.6988	1000	0	6.4	2.6	1.079983761417...	11652	127	Casatenovo tetto
9	5	Precipitazione	mm	0.0308	1000	0	8	0	0.395382963325...	10376	706	Lecco v.Sora
10	5	Temperatura	°C	21.0338	1000	12.7	30.5	20.5	4.529100386424...	9021	1234	Osnago
11	5	Velocità Vento	m/s	1.5543	1000	0	6.6	1.3	1.041626850890...	19116	706	Lecco v.Sora
12	5	Precipitazione	mm	0.0226000...	1000	0	3.2	0	0.183856038908...	9110	1234	Osnago

Figure 11 – Attribute table of "Statistics per sensor" layer

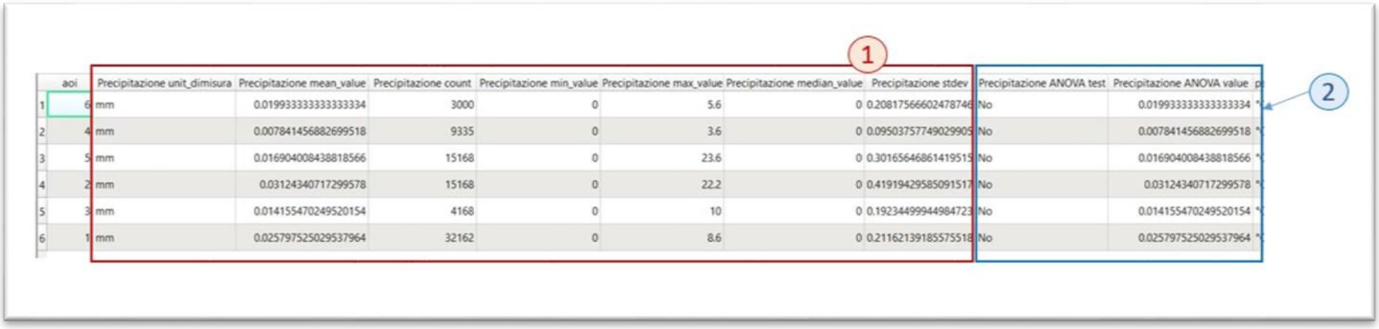


- ● W -WE - S for each aoi - 01/06 1:00 to 08/06 0:00
- ● W - WE- S per sensor - 01/06 1:00 to 08/06 0:00
- ● W -WD - S for each aoi - 01/06 1:00 to 08/06 0:00
- ● W - WD- S per sensor - 01/06 1:00 to 08/06 0:00
- ● W - Stations
- 🟩 Area(s) of interest

Figure 12 – Layers if distinction between Weekdays and Weekends

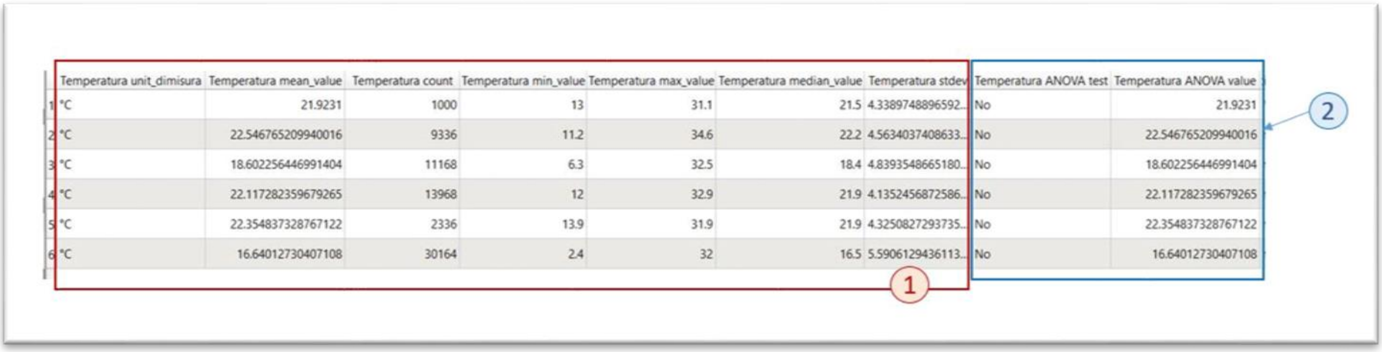
iv. **One map of points representing one point in the centre of each of the areas of interest** (Fig. 8 (5)). **Each point contains the statistics of the selected sensor types for the concerned area** (cf. Fig. 13 and 14 (1)). Moreover, the Fisher test result is present for each type of sensor (cf. Fig. 13 and 14 (2)). Here again, if the user chose to have weekdays and weekends with a distinction, there are two different maps. *Named "W - S for each aoi - Dates" or "AQ - S per each aoi - Dates"*

Note: If the areas of interest are a unique area, the final map is a unique point with the statistics of each sensor type for the whole area.



	aoi	Precipitazione unit_dimisura	Precipitazione mean_value	Precipitazione count	Precipitazione min_value	Precipitazione max_value	Precipitazione median_value	Precipitazione stdev	Precipitazione ANOVA test	Precipitazione ANOVA value
1	6	mm	0.019933333333333334	3000	0	5.6	0	0.2081756660247874€	No	0.019933333333333334
2	4	mm	0.007841456882699518	9335	0	3.6	0	0.09503757749029905	No	0.007841456882699518
3	5	mm	0.016904008438818566	15168	0	23.6	0	0.30165646861419515	No	0.016904008438818566
4	2	mm	0.03124340717299578	15168	0	22.2	0	0.41919429585091517	No	0.03124340717299578
5	3	mm	0.014155470249520154	4168	0	10	0	0.19234499944984723	No	0.014155470249520154
6	1	mm	0.025797525029537964	32162	0	8.6	0	0.21162139185575518	No	0.025797525029537964

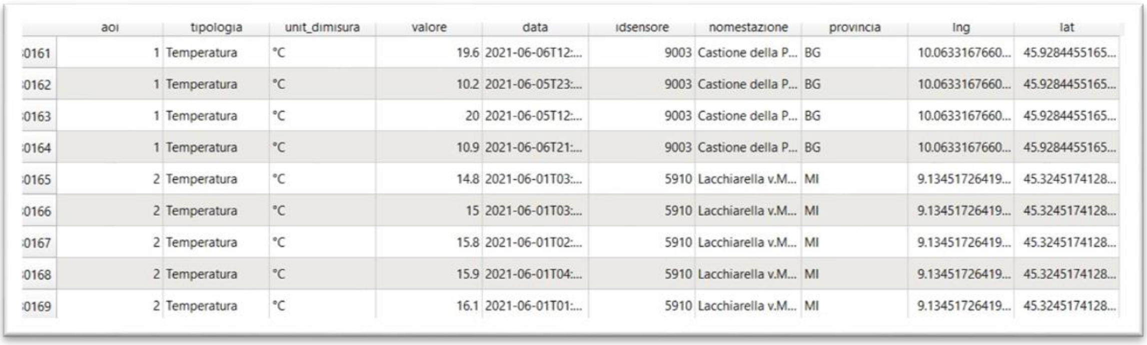
Figure 13 – Attribute table of "Statistics per aoi" layer (part 1)



	Temperatura unit_dimisura	Temperatura mean_value	Temperatura count	Temperatura min_value	Temperatura max_value	Temperatura median_value	Temperatura stdev	Temperatura ANOVA test	Temperatura ANOVA value
1	°C	21.9231	1000	13	31.1	21.5	4.3389748896592...	No	21.9231
2	°C	22.546765209940016	9336	11.2	34.6	22.2	4.5634037408633...	No	22.546765209940016
3	°C	18.602256446991404	11168	6.3	32.5	18.4	4.8393548665180...	No	18.602256446991404
4	°C	22.117282359679265	13968	12	32.9	21.9	4.1352456872586...	No	22.117282359679265
5	°C	22.354837328767122	2336	13.9	31.9	21.9	4.3250827293735...	No	22.354837328767122
6	°C	16.64012730407108	30164	2.4	32	16.5	5.5906129436113...	No	16.64012730407108

Figure 14 – Attribute table of "Statistics per aoi" layer (part 2)

v. If the raw data have been asked:
One map of points representing the sensors concerned by the filters and containing all their measurements in the period selected. (Fig. 8 (3)) Each row of the layer contains one measurement. Each measure is associated to the point on the map corresponding to the location of the sensor. *Named "W - Raw data - Dates" or "AQ - Raw data - Dates"*



	aoi	tipologia	unit_dimisura	valore	data	idsensore	nomestazione	provincia	lng	lat
0161	1	Temperatura	°C	19.6	2021-06-06T12:...	9003	Castione della P...	BG	10.0633167660...	45.9284455165...
0162	1	Temperatura	°C	10.2	2021-06-05T23:...	9003	Castione della P...	BG	10.0633167660...	45.9284455165...
0163	1	Temperatura	°C	20	2021-06-05T12:...	9003	Castione della P...	BG	10.0633167660...	45.9284455165...
0164	1	Temperatura	°C	10.9	2021-06-06T21:...	9003	Castione della P...	BG	10.0633167660...	45.9284455165...
0165	2	Temperatura	°C	14.8	2021-06-01T03:...	5910	Lacchiarella v.M...	MI	9.13451726419...	45.3245174128...
0166	2	Temperatura	°C	15	2021-06-01T03:...	5910	Lacchiarella v.M...	MI	9.13451726419...	45.3245174128...
0167	2	Temperatura	°C	15.8	2021-06-01T02:...	5910	Lacchiarella v.M...	MI	9.13451726419...	45.3245174128...
0168	2	Temperatura	°C	15.9	2021-06-01T04:...	5910	Lacchiarella v.M...	MI	9.13451726419...	45.3245174128...
0169	2	Temperatura	°C	16.1	2021-06-01T01:...	5910	Lacchiarella v.M...	MI	9.13451726419...	45.3245174128...

Figure 16 – Attribute table of "Raw data" layer

All these layers are temporary layers and need to be saved if the user wants to use them in future sessions. **<u>They must be saved as Geopackage</u>** in order to keep the attribute names as they are in the layers. If saved as shapefile, the layer will contain shortened attribute names and it prevents from knowing what they correspond to.

2. <u>Visualization</u>

If the user selected only one type of sensor and decided to ask the visualization of the results, histograms are automatically built for two layers:

- Layer of statistics per sensor (iii)
- Layer of statistics per area of interest (iv)

The histograms are placed on the points of the layer and represent, in order from left to right, the max value, the mean value, the median value, the min value and the standard deviation of the measurements.

For the layer of statistics per area of interest, the diagrams appear in a "small scale" ranging from 1:1300000 to 1:350000.

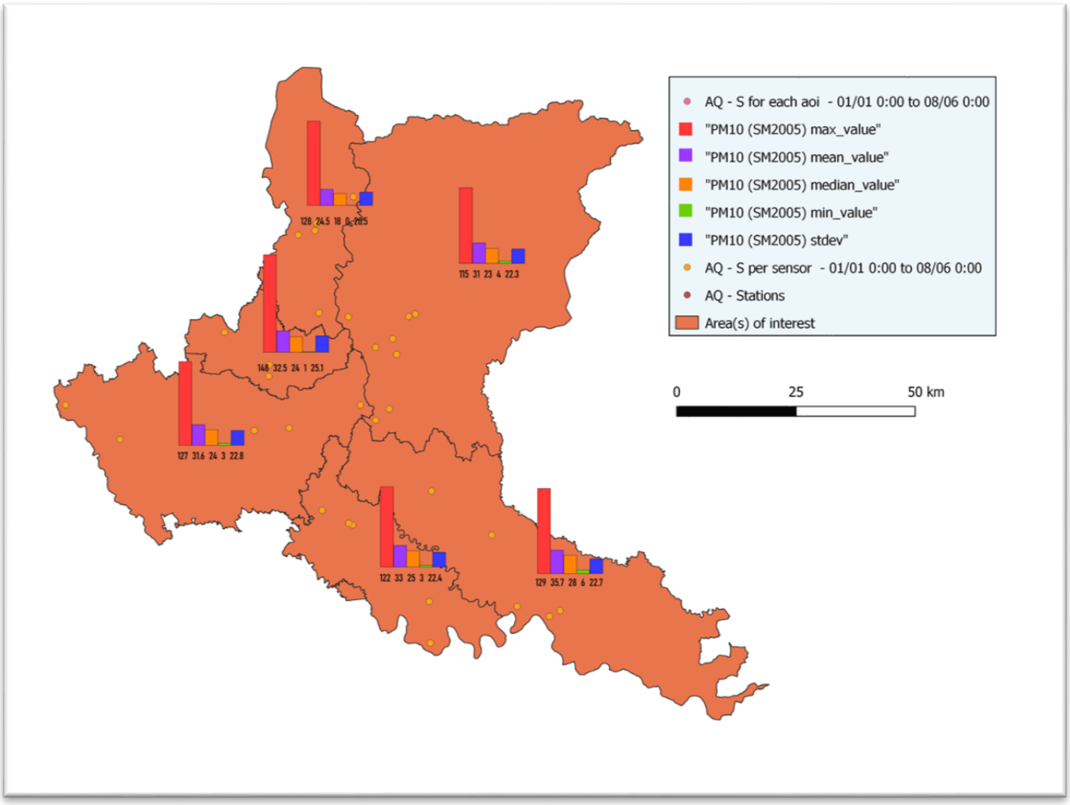

128 24.5 18 0 20.5
115 31 23 4 22.3
148 32.5 24 1 25.1
127 31.6 24 3 22.8
122 33 25 3 22.4
129 35.7 28 6 22.7
AQ - S for each aoi - 01/01 0:00 to 08/06 0:00
"PM10 (SM2005) max_value"
"PM10 (SM2005) mean_value"
"PM10 (SM2005) median_value"
"PM10 (SM2005) min_value"
"PM10 (SM2005) stdev"
AQ - S per sensor - 01/01 0:00 to 08/06 0:00
AQ - Stations
Area(s) of interest
0
25
50 km

Figure 17 – Visualization of statistics per aoi (small scale)

In contrast, for the layer of statistics per sensor, the diagram appears in a "large scale" starting from 1:350000.

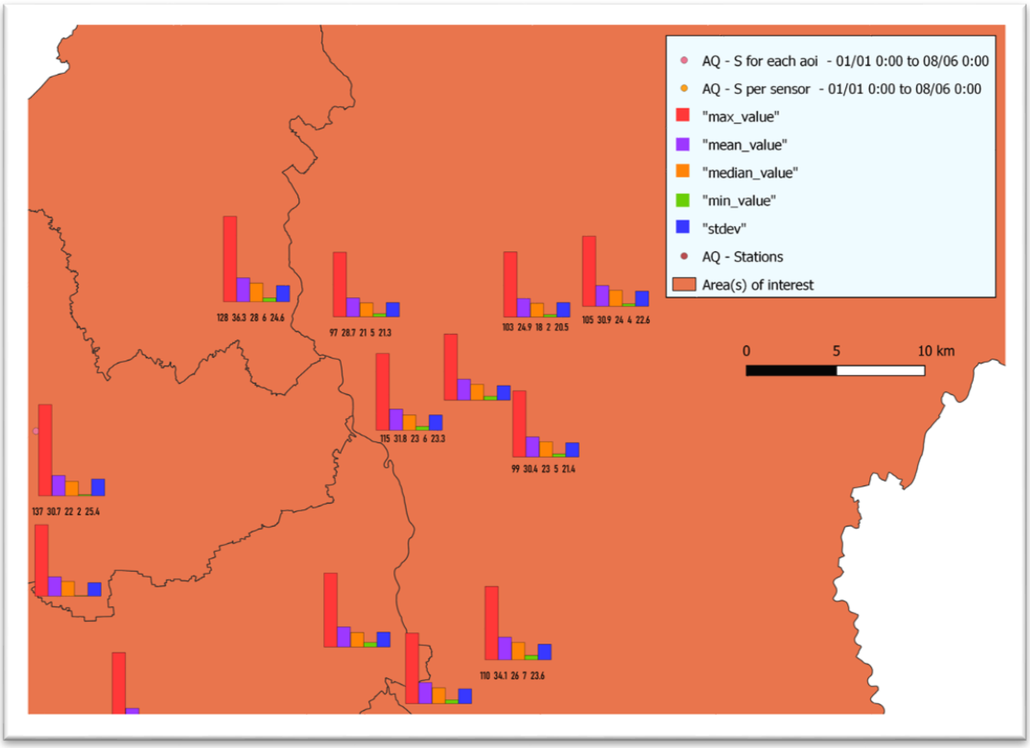


Figure 18 – Visualization of statistics per sensor (large scale)

Specific case: If the user chose to have weekdays and weekends with a distinction and also asked for the visualization, the resulted maps will contain the histograms for each of them. As the histogram for weekdays and the one for weekends are placed at the same position, they are overlapping. Therefore, they must be visualized one by one. One of the two layers (WE or WD) has to be hidden by the user to have a correct visualisation of the histograms.

d. Plugin Implementation

This plugin has been implemented in Python, using the python scripting in QGIS (PyQGIS) to interact with QGIS and QT Creator to design the interface. To set-up all the necessary files and the standard code for the plugin at the beginning, the QGIS Plugin Builder was firstly used.

1. Get the data from ARPA API

The open datasets of ARPA Lombardia are hosted by Socrata (Socrata Open Data API) that provides the built-in API. To use it with Python, the library “sodapy” has to be installed.

Each available dataset can be retrieved by sending a request to this API and the response is received in JSON format. Therefore, a first request is made to get the information related to the stations of the selected dataset (weather or air quality). Then, for each of the sensors returned, a second request is made to get its monitoring data. This means that, for N stations in the area of interest, the number of requests is N+1:

- 1 for station information
- N for the monitoring data (1 per sensor)

The responses are stored in a python dictionary to be added to a new layer into QGIS.

2. Create and add a layer into QGIS

In our case, the layers to add are all vector layers. Using PyQGIS, the creation of a vector layer only requires setting the geometry type (point, linestring, polygon, multipolygon...), the coordinate reference system (CRS), the name of the layer and the name of the data provider. For all the layers created in this project, we used the World Geodetic System 84 (EPSG:4326) because the data from ARPA is provided in this CRS.

Then, the fields for the created vector layer have to be set and the features to be added. This is done item by item, setting the attributes and the geometries one by one.

Finally, the vector layer is added to the current QGIS project (if none is opened, a new one is created). The resulted layer is added as temporary layer and has to be saved by the user manually if he/she wants to keep it in the long term.

This process is followed for all layers created by the plugin.

3. Add filters

The data was up till now requested without any filter: all the monitored data of all the types of sensors were requested for the selected dataset. Some filters had to be added to give more flexibility to the user and to prevent him/her from making a useless big request that takes more time to be served.

Filter on data type

Firstly, a filter on the type of sensor has been added. The user could choose to get data for all types of sensors or only for a subset of them. Depending on this choice the query to the API is modified and a condition is added on the type of sensor.

Then, we decided to give the possibility to the user to request only the information about stations without the monitoring data. This is especially useful to get only stations location. In that case, only one request is made, and the response time is very short.

Filter on time

Another important filter to be implemented was the one on time. The user could want to have the monitoring data for the all period of monitoring or only on a restricted period. Therefore, a start date and end date have been added. The query for each sensor is taking this range of time into consideration and only returns monitoring done during this period.

In addition, a comparison between weekday and weekend measures could be useful. For example, considering the air quality data, this would allow an analysis of the difference in C02 between working days and weekends. Therefore, we decided to add the possibility to distinguish the monitoring data of weekdays from the weekends ones through three checkboxes called ‘weekdays’, ‘weekends’ and ‘no distinction’. By default, they are all checked, which means that all the data from weekends and weekdays are requested without making any distinction. In contrast, if the "no distinction" box is not checked, weekday data will be provided to the user separately from weekend data. Here, the distinction is made after the request on the received JSON.

Filter on space

The data provided by the API concerns the whole region. However, the user could want to have only data from a specific part of Lombardy.

Thus, a first filter on space has been added: the filter on provinces. Indeed, the dataset containing information about stations provides the name of the province to which each of them belongs. Therefore, a condition is added to the query on stations in order to request only the information in the desired province. Nevertheless, the user has still the possibility to ask for data in the whole region by selecting “All provinces”.

In addition, the user could want to focus on a certain area that is not necessarily a province. In that case, he/she has the possibility to insert a shapefile that contains multi-polygons and has the WGS84 CRS. The different polygons contained in the file are extracted and define the list of areas of interest. For each of them, the N+1 requests on station information and monitoring data are made (1 for station information, and 1 per sensor for measurements).

In the end, the result provided to the user contains a distinction by zone. A layer called “Area(s) of interest” is added to the QGIS project and corresponds either to the file previously inserted by the user, or to a shapefile contained in the plugin directory. Each point of the produced maps has an attribute called “aoi” containing an id that matches one of the polygons of the "Area(s) of interest" layer.

4. Provide statistics

The data requested are monitoring data from sensors that are periodically measuring. As explained in the Data Analysis section, the weather sensors are measuring every 10 minutes and the air quality ones every hour. Therefore, the amount of data available is huge and adding all the measurements of all sensors in all the areas of interest could take a lot of time. For this reason, it has been decided to provide statistics rather than raw data.

Firstly, for each sensor, a list containing all measurements over the selected period of time is built. Then, from this list, the following statistics are computed: maximum, minimum and median value, mean, count and standard deviation. They are rendered in one layer representing each of the sensors. One point (i.e one row) corresponds to one sensor with its statistics.

Secondly, for each area of interest, several types of sensors can be present, but the statistics must be computed type by type. Therefore, for each area, one list per sensor type is built and the same parameters are computed: maximum, minimum and median value, mean, count and standard deviation. They are rendered in one layer representing each area of interest. One point (i.e one row) corresponds to one area of interest and contains all the statistics for each type of sensor.

However, the user could want to have the raw data and not the statistics. For this reason, the possibility to ask for the raw data has been added. This functionality is limited to the case when the user selects only one type of sensor in order to prevent the process from being too long. The measurements are directly added to a new layer without any processing. In that layer, one point (i.e one row) corresponds to one measure of one sensor. Depending on the selected period of time, this layer can contain a huge number of rows.

5. Implement the Fisher test

The results provided by the plugin could be used to build models of weather or air quality in Lombardy. For example, one could want to build a model of the temperature in the region. The idea is to determine whether a relationship exists between the means of temperature in the areas. If so, a unique general mean value could be used to model the temperature in the areas. This is the Analysis of variance (ANOVA) method and can be used for each type of sensors.

Therefore, at the end of the collection of data and the computation of statistics, the ANOVA test is made. The within areas source of variation is computed as well as the between areas source of variation. Using the Fisher F-Test, a comparison with the theoretical value corresponding to 5 % probability is made and determines if the test passes or not.

The result is stored in the layer containing the statistics of each area of interest (*layer n° iv*). For each area and each type of sensor, two attributes are set:

- ANOVA test: YES if the test passed; NO if it didn't pass.
- ANOVA values: Mean value of the mean of each area if it passed; Mean value of the area if it didn't pass.

Special case: When the within area source of variation is equal to zero, there is no internal variability in each region. The value for each area is deterministically always the same. The variable is therefore not random or the sample too small to apply the Fisher test. In that case, the test is not applicable.

6. Provide an understandable symbology

The final step was to provide a visualization of the statistics to the user. As several types of sensors can be selected, having a visualization mixing them would not make sense (the mean temperature cannot be compared to the mean velocity of wind). Therefore, visualizing the resulted statistics is possible only when the user selected one single type of sensor.

The representation of the statistics is made by means of histograms with five bars (maximum value, mean, median, minimum value and standard deviation). They are built for the map of statistics per area and for the one of statistics per sensor. To prevent the diagrams from overlapping each other, a scale dependent visibility is set. In a small scale, the histograms of areas are visible while in a large scale, the visibility is for histograms of each sensor.

Moreover, the values are displayed below the graphs as labels to have an idea of the quantity represented by each bar. We can notice that some labels are not appearing when the diagrams are too close from one another to prevent any overlapping. Also, some labels seem to be misaligned with the diagrams. This is because, when diagrams are too close from one another, they are automatically displaced while labels keep the original position. In any case, when zooming in, the diagram takes its original placement, and the alignment is good again.

IV. Limitations and improvements

a. Limitations on data

Several limitations on the requested data are present. First, as explained in the data analysis section, only a subset of the monitoring data is available through the API (current month for the weather and current year for the air quality).

Secondly, some of the air quality sensors seem not to provide any monitoring data anymore. For example, the requests on the Arsenico, Benzo(a)pirene, Cadmio, Nikel, Particolato Totale Sospeso, Piombo and PM10 in June don't provide any answer.

Finally, sometimes the data of the last week are not providen. Therfore, when requesting the data for the last week nothing appear in the response.

b. Possible improvements of the plugin

The plugin could be improved by adding the possibility to request more data from the ARPA API. Indeed, the plugin is focusing on the air quality and weather data, but other datasets are available: flow data, water quality and waste collection.

Moreover, the region of Lombardy performs some analysis and produces maps with aggregated data about air quality. The regional aggregated data is not available but there is an alternative: the communal estimations. The dataset called "stime comunali" contains aggregated municipal values calculated from the results of regional-scale simulations. They are not measurements but estimates that also use data from the ARPA air quality network and are available through the API. Therefore, the plugin could also add the possibility to add these data.

Finally, if one day the older monitoring data are made available by ARPA through the API, the plugin should be improved to be able to request them.

V. **Tools used in the project**

- QGIS (https://qgis.org)
- ARPA Lombardia API (https://tinyurl.com/yyw6rgoq)
- Pandas (https://pandas.pydata.org)
- QGIS Processing Script (https://tinyurl.com/yxe7ww69)
- QT Creator (https://www.qt.io/offline-installers)
- PyQGIS (https://docs.qgis.org/3.16/en/docs/pyqgis_developer_cookbook/index.html)