Introduction to GIS

Raphaëlle ROFFO

Sciences Po - GETEC Masters
Fall Semester 2021

Session 3

Working with vector data: the attribute table

Today's plan

1. Tutorial debriefing
2. Last week recap
3. Understanding the attribute table
4. Tutorial

Session 2 tutorial debriefing

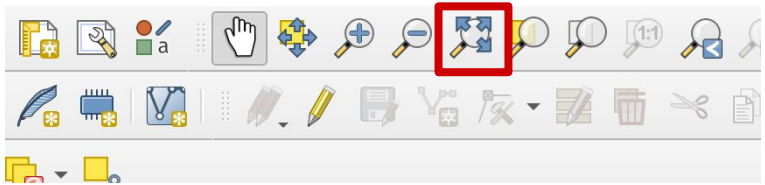
Tutorial objectives

The goal of this tutorial was to walk you through:

- The process of **downloading data** from a data portal
- The difference between a **QGIS project, layers,** and **data sources**
- What ***.qgz** project files are
- The **import** of data into a QGIS project
- The **export** of data, including in different formats / CRS
- The use of **geopackages** as the best practice for storing vector data and even QGIS project files.

Remarks

- **Projections:** most recent versions of QGIS reproject your layers “on the fly”, meaning it “translates” layers that are in a different CRS so that they actually fit your project CRS. See what happens when you use a wrong CRS (demo). For more details, read sections 10.1 to 10.3 of this page.
- When you’ve “lost your layers” or don’t know where you are on your map canvas, this magnifier is your friend:



Questions?

Common issues include:

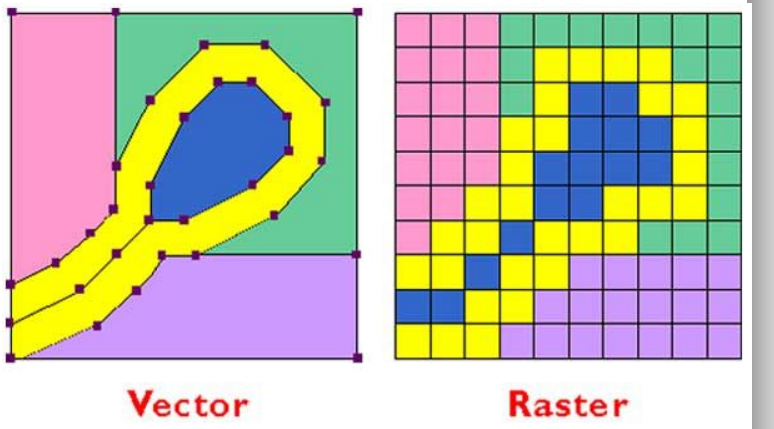
- opening shapefiles,
- setting up the correct CRS,
- locating data after some files were moved to a different folder
- etc...

Last week recap

Vector vs Raster

Spatial data can be represented in two ways:

- **Vector** = Object view = Discrete = geometries: Point, Line, Polygon
- **Raster** = Field view = Continuous = matrix of cells/pixel that each hold a numeric or categorical value



Source: David DiBiase et al, The Nature of Geographic Information

The importance of metadata

Always ensure you work with data from **reputable sources**, and check for *completeness, currency, consistency, accuracy* etc to make sure the data is **relevant** and **fit for purpose**.

To download data, explore **open data portals** at EU, country, city level as well as specialised agencies (environmental agencies, statistics offices, etc.). You can also explore OpenStreetMaps for geographic features such as roads, rivers, buildings etc.

The GIS workflow

1. Define a **research question** / gather requirements for your end user
2. Download some **data**
3. Load it in QGIS or other software to **explore the data**; check the quality and content of the data, and to refine your research question
4. You may need to go back to step 2 here, and **iterate** a few times through steps 2 and 3 to **refine your research question** and determine the exact datasets you want to work with
5. The **analysis**
6. **Symbology** and visualisation

The attribute table

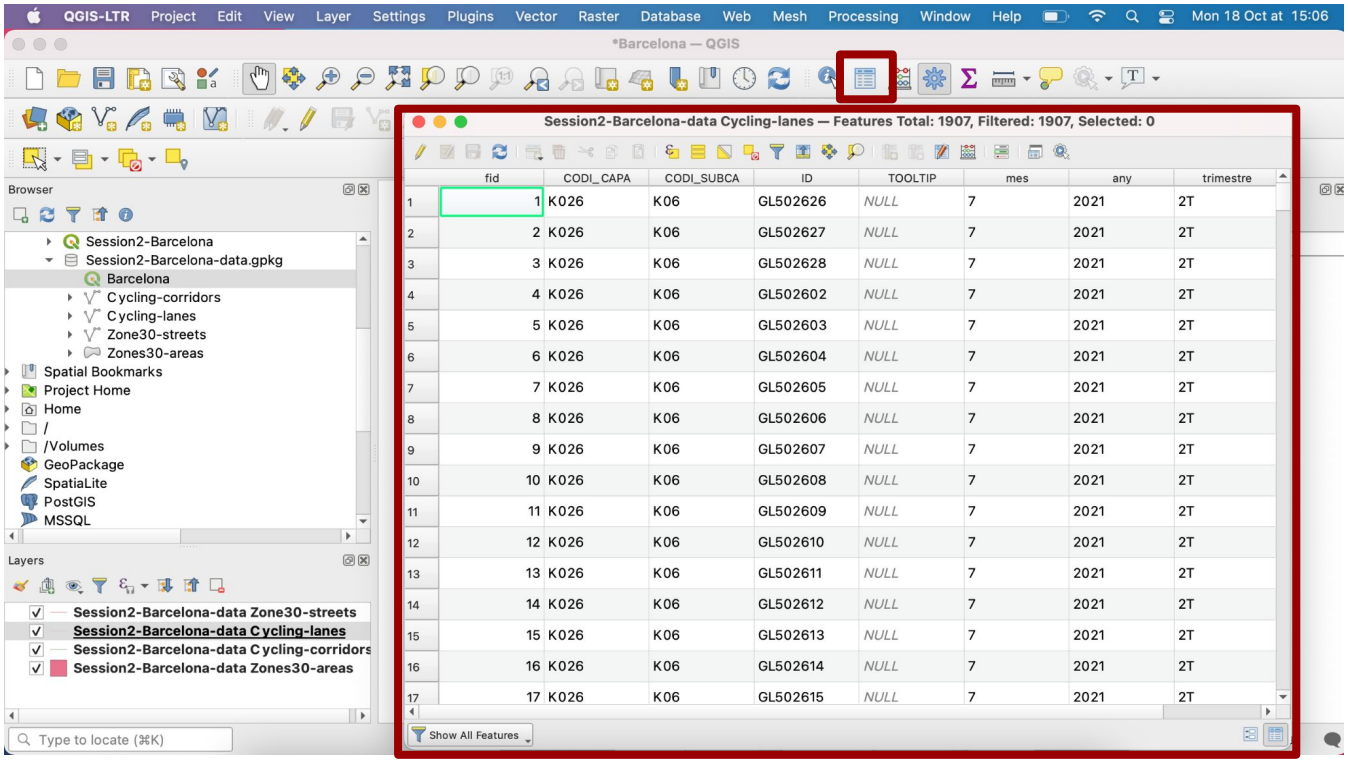
The Attribute Table

GIS maps are not just pictures; the reason GIS is such a powerful tool is that **the objects displayed on the map also hold attributes**. Your map not only represents where objects are located, but it usually also carries information about these objects.

In the case of **vector data**, this information is contained in the **attribute table**.

(in the case of raster data: it's the grid itself and the values held by each raster cell)

The Attribute Table



	fid	CODI_CAPA	CODI_SUBCA	ID	TOOLTIP	mes	any	trimestre
1	1	K026	K06	GL502626	NULL	7	2021	2T
2	2	K026	K06	GL502627	NULL	7	2021	2T
3	3	K026	K06	GL502628	NULL	7	2021	2T
4	4	K026	K06	GL502602	NULL	7	2021	2T
5	5	K026	K06	GL502603	NULL	7	2021	2T
6	6	K026	K06	GL502604	NULL	7	2021	2T
7	7	K026	K06	GL502605	NULL	7	2021	2T
8	8	K026	K06	GL502606	NULL	7	2021	2T
9	9	K026	K06	GL502607	NULL	7	2021	2T
10	10	K026	K06	GL502608	NULL	7	2021	2T
11	11	K026	K06	GL502609	NULL	7	2021	2T
12	12	K026	K06	GL502610	NULL	7	2021	2T
13	13	K026	K06	GL502611	NULL	7	2021	2T
14	14	K026	K06	GL502612	NULL	7	2021	2T
15	15	K026	K06	GL502613	NULL	7	2021	2T
16	16	K026	K06	GL502614	NULL	7	2021	2T
17	17	K026	K06	GL502615	NULL	7	2021	2T

Records, features, fields, attribute values

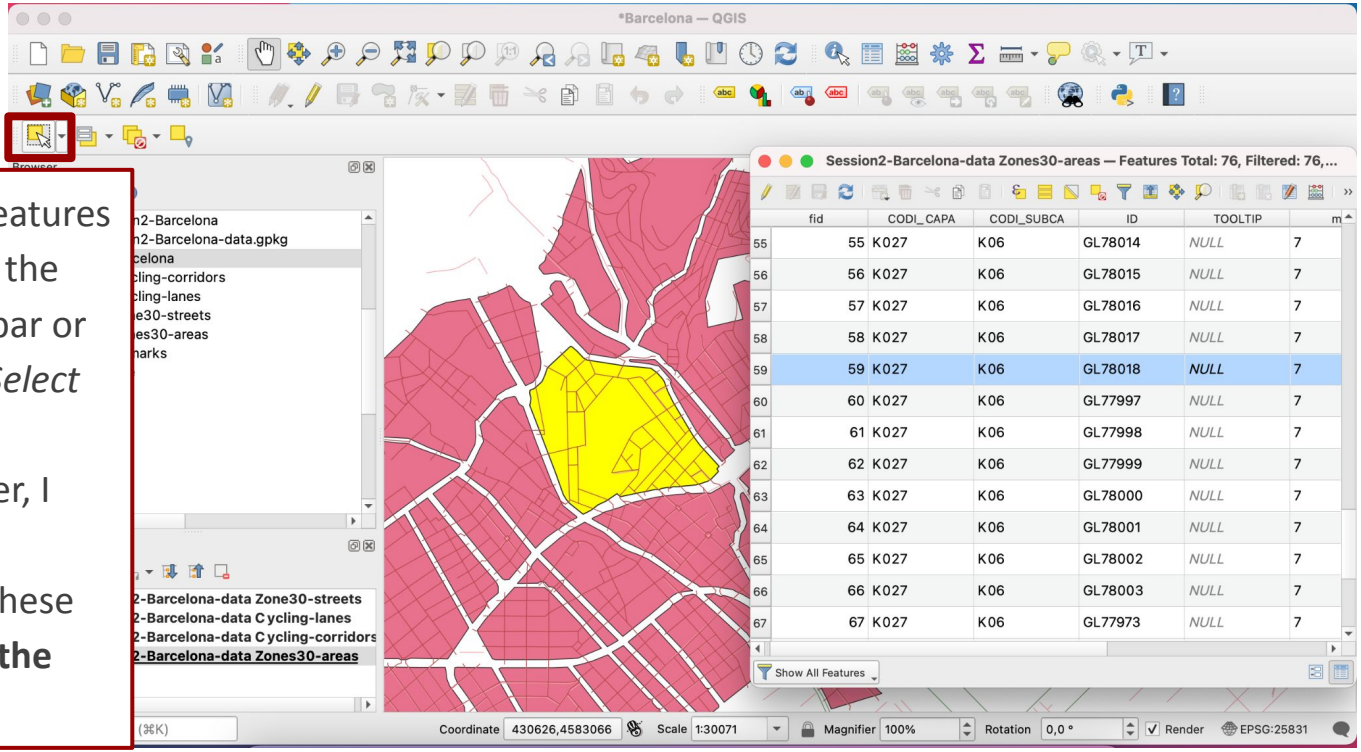
In your attribute table, we call a *row* a **record**. This record contains information about one **feature** (a *single polygon, line or point*). If your cycling lane layer contains 1907 line features, the attribute table will also contain 1907 rows.

The *columns* are called **fields**; their name usually aptly describes the information they contain.

The *value in a cell* is called an **attribute value**.

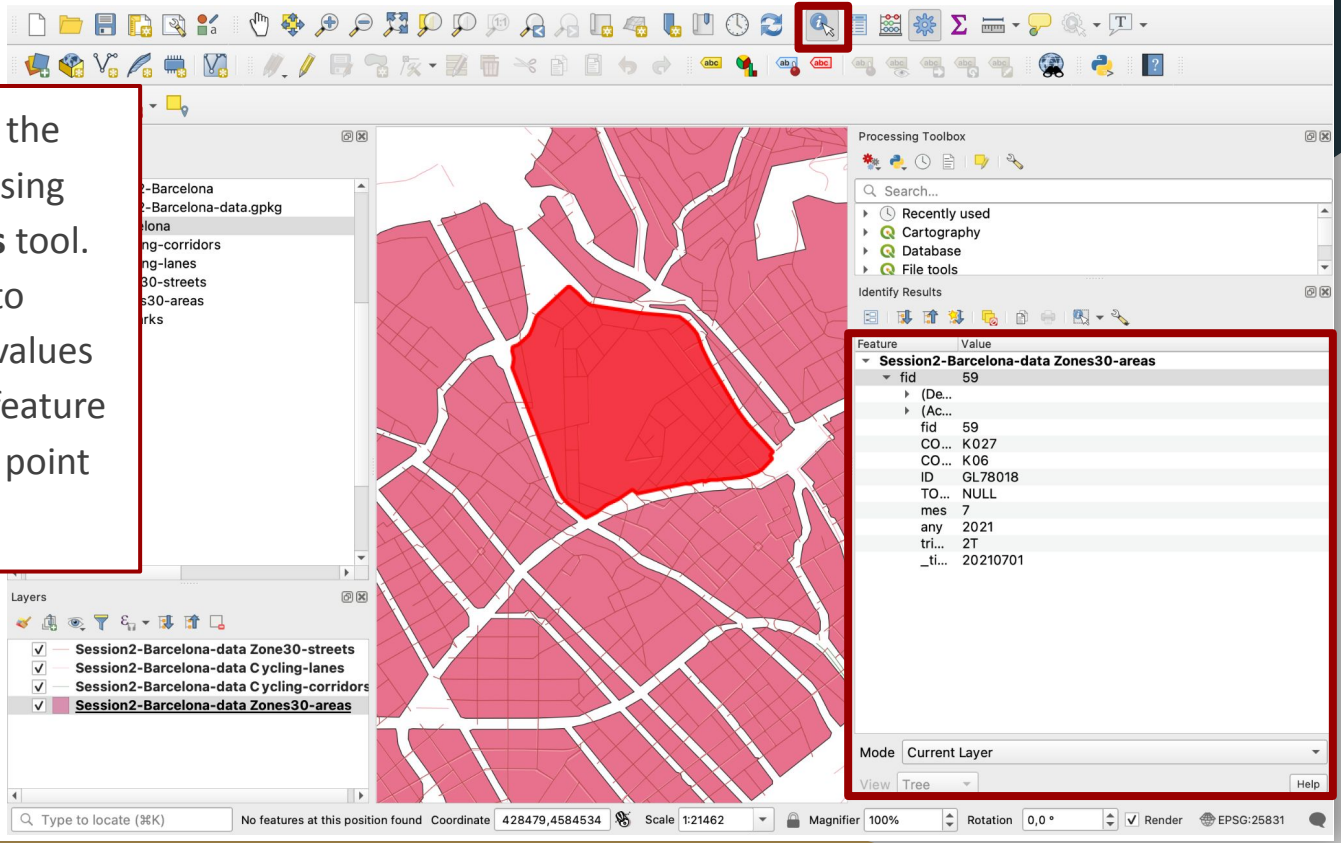
Accessing a feature's record

If I select one or several features on the map canvas (using the **selection icon** in the toolbar or the menu *Edit > Select > Select Features*), then open the attribute table of that layer, I will be able to see one or several highlighted row. These are **the records linked to the selected features.**



Identifying features

You can also retrieve the **record** of a **feature** using the **Identify Features** tool. This tool allows you to access the attribute values stored for a specific feature (one polygon, line of point that you click on).



Thinking like SQL

To understand how you can best interact with the attribute table, you need to know that QGIS is built in C++ and Python, and it interacts with data using SQLite and Spatialite, with the **SQL (Structured Query Language)** syntax.

What this means in practice is that QGIS is very good at **querying data**. You can build queries to go find the rows in your table that match certain constraints, a bit like setting your own **filters** to only return rows you're interested in (for example *"Select all the rows WHERE condition A is true and condition B is true").*

→ In my buildings table, select all the rows where **"Height" > 20 and "Material" = 'Glass'.** This will return the rows that match this condition and therefore their linked features too.

Dealing with delimited text files (*.csv etc.)

In QGIS, you can load spatial and *non-spatial tables*. **Vector layers are spatial tables**; each vector layer contains a geometry and an attribute table. But you can also load **delimited text files** like *.csv in your QGIS projects. Two configurations are possible:

- The layer contains s**ome coordinates / geometries** (e.g. two fields for latitude and longitude). In that case you can turn that table into a vector layer.
- The layer contains **text only**: in that case you can't turn it into a vector layer. However, **maybe one of the fields can be linked to an existing geometry** using an **attribute-based join** (e.g. census data linked to a polygon layer of IRIS zone/ LSOA/ census block). For this, it's crucial that you work with unique records.

Joining layers by attribute

When you join two layers by attribute, you take two input layers and the output will be a version of your **Input layer 1** (with the same geometry, all rows and fields from your attribute table), but additional fields from **Input layer 2** are now extending the attribute table.

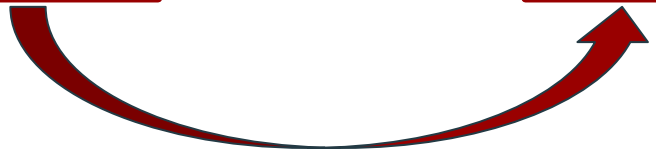
In order to do so, **Input layer 1** and **Input layer 2** must have a common field, so QGIS knows which rows to connect to each other. In SQL, this corresponds to a left join.

→ I have a **vector layer 1** that contains **country boundaries**, and a **non-spatial layer 2** that is a **table of the population count** for each country. Fortunately, attribute table 1 contains the **name of each country** and so does attribute table 2. I can join these two layers using the country name as join field, to **"attach" the population count to the country polygons**.

Joining layers by attribute

Admin boundaries (polygons)	
Unique ID	Geometry
CE0123	polygon1
CE0456	polygon2
CE0789	polygon3

Census data (non-spatial table)		
Census block ID	Household count	Unemployment %
CE0123	10120	6.3
CE0456	9800	7.9
CE0789	11300	5.4



Joining layers by attribute

Output layer: Admin boundaries (polygons) with extra attributes from the census			
Unique ID	**Geometry**	**Household count**	**Unemployment %**
CE0123	polygon1	10120	6.3
CE0456	polygon2	9800	7.9
CE0789	polygon3	11300	5.4

Tutorial

Homework

1. Do the QGIS tutorial on the attribute table
2. Use Slack if you have questions (#help).