Introduction and install

Alexandre Neto

January 16, 2026

Introduction and installs

What software we will be using

- Miniforge (Python package and environment manager)
- Python 3.10+ (installed from miniforge)
- VS Code (the IDE/text editor)
- PostgreSQL (with the PostGIS and pgRouting extensions)
- QGIS (mostly for data inspection)
- Postman (to test APIs)

We will be installing most of this during this session and make sure everything is working as expected for the next classes.

Recommended VS code extensions

- Python extension (ms-python.python)
- Jupyter extension (ms-toolsai)

What Python packages to install

Open your command line.

- If you are on windows, open conda command prompt.
- If you are on a mac, open terminal.
- If you are on linux, you should know how to do this!

Start by creating a new environment to isolate the installation of new packages:

```
conda create -n gps_python_intro
```

Activate the environment:

```
conda activate gps_python_intro
```

Now, install the python packages that we will use

```
conda install ipykernel -c conda-forge
```

This will install `ipykernel` package that will allow you to see and work with Jupyter notebooks inside VS Code. Note that Jupyter notebooks **ARE NOT** programming tools, but rather an exploration and note taking tool.

Now, let's try to install other python packages:

```
conda install pandas geopandas psycopg2 matplotlib -c conda-forge
```

If you get any problem raise your hand (or mail me).

Group Project

In a group up to 3 persons, you will develop a coding project. The project thematic is open in nature, but your project must include all of the **following components**:

- An Extract Transform and Load (ETL) module;
- A relational database model prepared for Create, Read, Update, Delete (CRUD) operations;
- An application programming interface (API) module;

Minimum requirements:

- The ETL should run in an independent way, that is, one single command that imports or updates new data into your database.
- The database model should cover at least 5 entities(main tables) and at least one must store spatial data.
- The API should have at least 5 endpoints to comunicate with the database, one of which should be for a POST method to insert data.

You **must use** the following technologies:

- Python 3 (ETL and API)
- PostgreSQL/PostGIS (Spatial Database)

You can use external technology, like extra packages, APIs, etc… as long as they are open source. In case of doubt, ask me.

The quality of your code will be evaluated using the following:

- Package dependencies are stated and available in a yaml file;
- Code is organized in modules and packages;
- Functions and methods have descriptive docstrings;
- Code is correctly formated and stylistically consistent;
- Code have comments explaining the steps logic, when clarification is needed;
- Modules have a README.md file explaining the purpose and use of the module.

Important dates

Before **Session 9**, groups will need to send me by email their project specification and objectives (preferably, this information can be already in the README.md file of the group code repository). This includes:

- Identifying the group members (up to 3 persons);
- Describing the project's main idea and workflow (ETL + Database + API);
- Origin of the dataset that will be utilised (use exclusively open data);

I will reply with an **OK**, if there are no relevant observations, or a set of recommendations, otherwise.

At the begining of **Session 10**, groups will have to quickly present their project specification and objectives to rest of the class.

I strongly advice you to have the all project code working by the end of **Session 18**, giving you time for code review and clean up, improving the README and preparing the final presentation

During **Session 20**, the group will do a 15 minutes presentation of the final results of their project, including a live demonstration, followed by a 5 minutes period for questions and answers from professors and their peers.

Code repository

During project development, to facilitate code sharing, communication and project assessement, **all the project's code should continualy be available on a online code repository** like Github or Gitlab. For instructions on how to work with github, I advise you to check the following URL https://github.com/skills/introduction-to-github.

The README.md file on your repository will also serve as your Final Report. It should contain:

- Introductions and description of your project
- Instructions on how to install and run your applications

- Results and conclusions
- Next steps

How are you going to be evaluated

Your grade is the weighted average of the following factors:

- Group project's grade 80% (16 points)
- Project presentation's grade 10% (2 points)
- Individual performance in course 10% (2 points)

As for the Group project's grade (up to 14 points) it will be distributed like this:

- An Extract Transform and Load (ETL) module (30%);
- A relational database Create, Read, Update, Delete (CRUD) module (30%);
- An application programming interface (API) module (30%);

The remained 10% are points reserved for groups that are able to explore the technologies and bring elements that were not explicitly utilised during the teaching sessions. (e.g. html, javascript, leaftlet, . . .)