Exploring Udine and its Municipalities

Project of Geospatial Analysis and Representation for Big Data

Daniele Passabì, 221229

Contents

1 Introduction

The project focuses on Udine and its municipalities. It is mainly aimed at students interested in studying and moving to the city.

The project code is public and can be explored from the following Github repository. Inside there is a READ_ME file, which contains all the information needed to run the code on your machine.

Alternatively, it is possible to visit this webpage and explore the notebooks online. The site also displays the key results obtained in a more user-friendly form.

The project is divided into three main parts:

1. *Static map of Udine, with universities and points of interest*

   It provides the main information about the city.

2. *Interactive map of Udine with real walking and cycling routes*

   The map is designed for those who want to train and test themselves.

3. *Geospatial analysis of average house prices and rents in the province of Udine*

   Interesting for those planning to move to Udine or its vicinity. It provides useful hints and considerations about the influence of each municipality on prices in neighbouring municipalities.

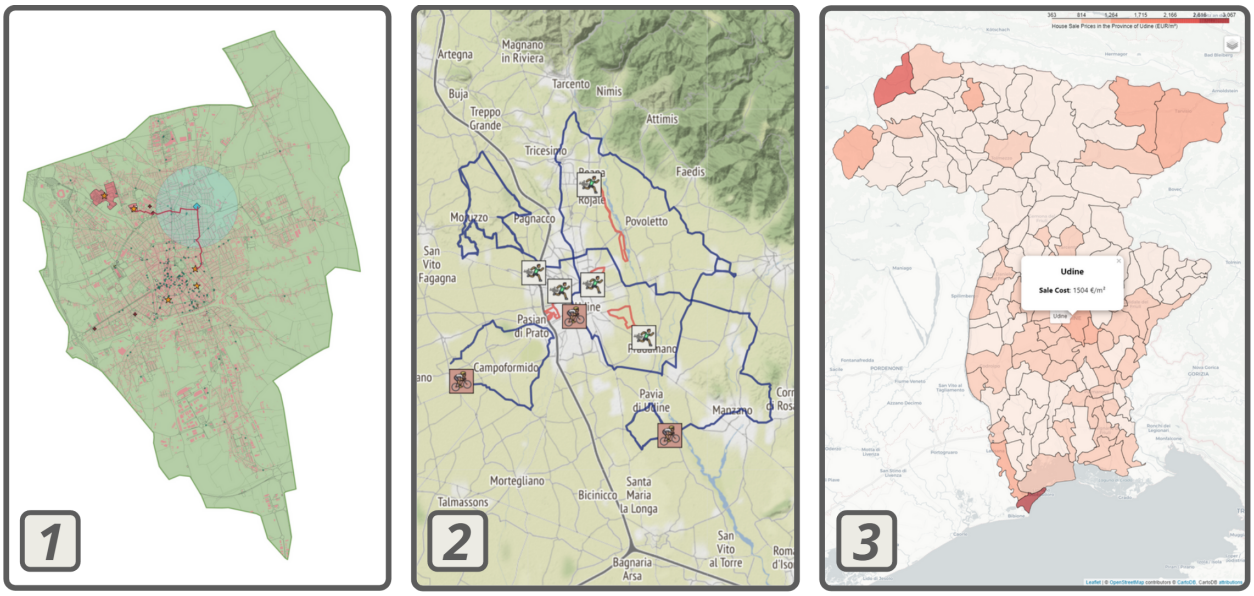


Figure 1: Preview of the 3 main parts of the project

In the following paragraphs we will analyse these components in depth, focusing on the data needed, the methodologies and technologies used, and showing the final result.

2 Udine Map with Universities and Points of Interest

2.1 Objective

The map aims to provide the user with useful information about the city of Udine, starting from the address entered by the student. The map is able to deliver insight about universities and points of interest generally useful to students, such as eating places, supermarkets, bus stations, bicycle and car rental, and hospitals.

The student can use the map to compare the location of different flats (or places to live) and choose the one that is most convenient for him/her.

2.2 Data

The following data are used to create the map:

- shapefile of the Italian municipalities, provided by the professor during the course;
- `.pbf` file of the city of Udine, obtained using the website Wikimedia Italy.

Wikimedia Italy offers a free service to extract data about Italy from the OpenStreetMap database. As we can read on the OpenStreetMap Foundation's FAQ page:

> *"The most important thing about the licence is that you do not have to pay anybody anything to use the data. There are NO copyright, licence, usage or other fees. You may use the data for personal, community, educational, commercial, government or any other use that you can think of."*

It is only required to attribute OpenStreetMap and to make public any changes we make to the data. For this reason, the Github repository where the complete project is located has been made public.

2.3 Methodologies and Technologies

The map is generated via the `plot_udine_map()` function. A brief explanation of how it works and the technologies used follows.

- First the buildings and streets data are extracted using `pyrosm`, a library that *"makes it easy to extract various datasets from OpenStreetMap pbf-dumps"* (documentation).
- This information is then added to the map of Udine, created through the `geopandas` library, an open source project to easily work with geospatial data.
- Afterwards, all the universities in Udine are extracted from OSM. The ones requested by the user are filtered and shown on the map with a star symbol. The geometry of the building is also displayed, if available.

- A similar logic is applied to points of interest. Based on those required, their information is extracted and plotted on the map with a customised symbol.
- If an address has been provided, it is geocoded through arcGIS to obtain its coordinates on the map.

  Special care was taken not to save the geocodes obtained, as required by arcGIS policy.

  Once the coordinates of the address have been obtained, the application checks that they are indeed within Udine and then shows the position of the address on the map, together with the 1km area around it, if requested.
- Finally the function uses the `osmnx` library to analyze Udine's street networks and obtain the following information:
  - The fastest routes to universities, together with the distances in meters.
  - For each type of point of interest the function returns the name and distance of the nearest location, together with the total number of said locations within one km of the address given.

  The information is stored in a dictionary, but using the `extract_info_from_dict()` function it can be displayed in tabular form.

2.4 Final Result

Figure 2 shows an example of possible input for the function, the map that is generated and the information that the function returns in tabular form.

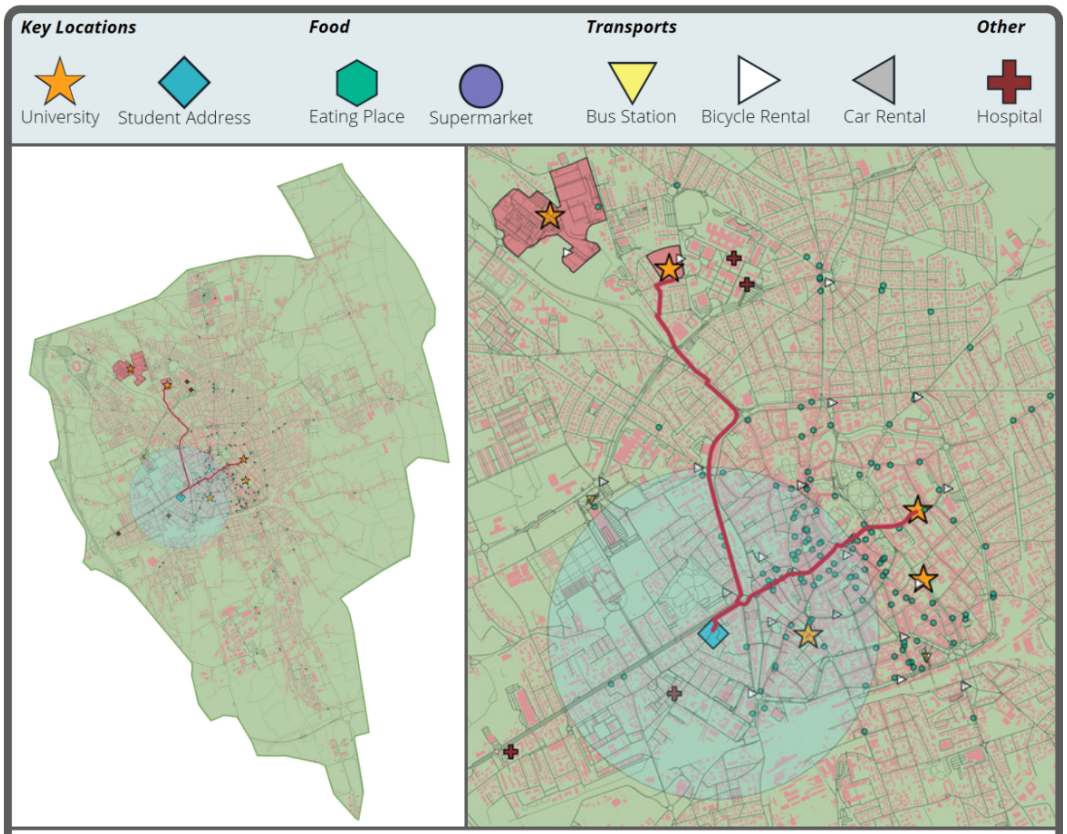
INPUT

Required Universities
- *Dipartimento di Area medica*
- *Dipartimento di Scienze Giuridiche*

***Show Uni Routes**: True*
***Show 1km Range**: True*

Required Locations
- *University*
- *Hospital*
- *Eating Place*
- *Bicycle Rental*
- *Bus Station*

OUTPUT



University	Distance (m)
Dipartimento di Scienze Giuridiche	1540.99
Dipartimento di Area medica	2605.93

Location	Closest (name)	Closest (distance)	Total in 1km Area
hospital	Policl. Città di Udine Polo 2	530.00	1
eating_place	Osteria Da Artico	393.95	68
bicycle_rental	Moretti	345.93	7
bus_station	Terminal Studenti	1337.19	0

Figure 2: Example of Input and Output of Udine's Map with Universities and Points of Interest

3 Interactive Udine Map with Strava Data

3.1 Objective

This map is aimed at sports people. It shows routes for running or cycling, based on real data collected by Strava, American internet service for tracking human exercise. For each route, details are available such as total distance, total travel time, minimum and maximum altitude, start and end point.

The map also shows the location of the fitness centres.

3.2 Data

There are two main types of data used:

- *Routes Data*

  The route data was downloaded through the tool provided by StravaToGPX. This allows the data to be obtained with the travel time of the routes, which is useful information that will be provided to the end user.

- *Sport Centers Data*

  The data of the gyms were obtained through `pyrosm` in a similar way as for the points of interest in the map shown in Section 2.

3.3 Methodologies and Technologies

The map is generated using `folium`, a library for creating interactive maps, based on the `leaflet.js` library.

The main function is called `create_folium_map()` and operates as follows:

- First the base map is created, starting with the given latitude and longitude; the correct level of zoom is set.
- Three groups are then created, one for running routes, one for cycling routes and one for gyms. The latter is a `MarkerCluster`, useful to manage the high number of gyms, which if displayed normally would have "polluted" the map.
- Three possible layers are added:
  - The first layer is the *OpenStreetMap layer*, useful for orientation within the city.
  - The second layer, called *Stamen Terrain*, provides a visualisation of the terrain. The user can thus understand at a glance where the running or cycling activity takes place and choose according to his or her personal preferences.
  - The last layer, called *cartodbpositron*, allows the user to focus on the routes thanks to its simple and clear theme.

- Routes are then added.
  - For each route the function takes care of retrieving from the data information on distance, travel time, altitude and starting and finishing locations.

    Note: start and end point details are reversely geolocalised using `Nominatim`.
  - This information is carefully formatted in HTML code, so that it can then be inserted into the markers `Popup` in the form of an `IFrame`.
  - Each route is assigned to its own group (*run* or *bike*)
- The information about the gyms is added to the sport centres group.
- Each group is finally added to the base map, together with the control centre layer.

3.4 Final Result

Figure 3 shows the final result. However, as it is an interactive map, it is advisable to view it in a website to better enjoy it.

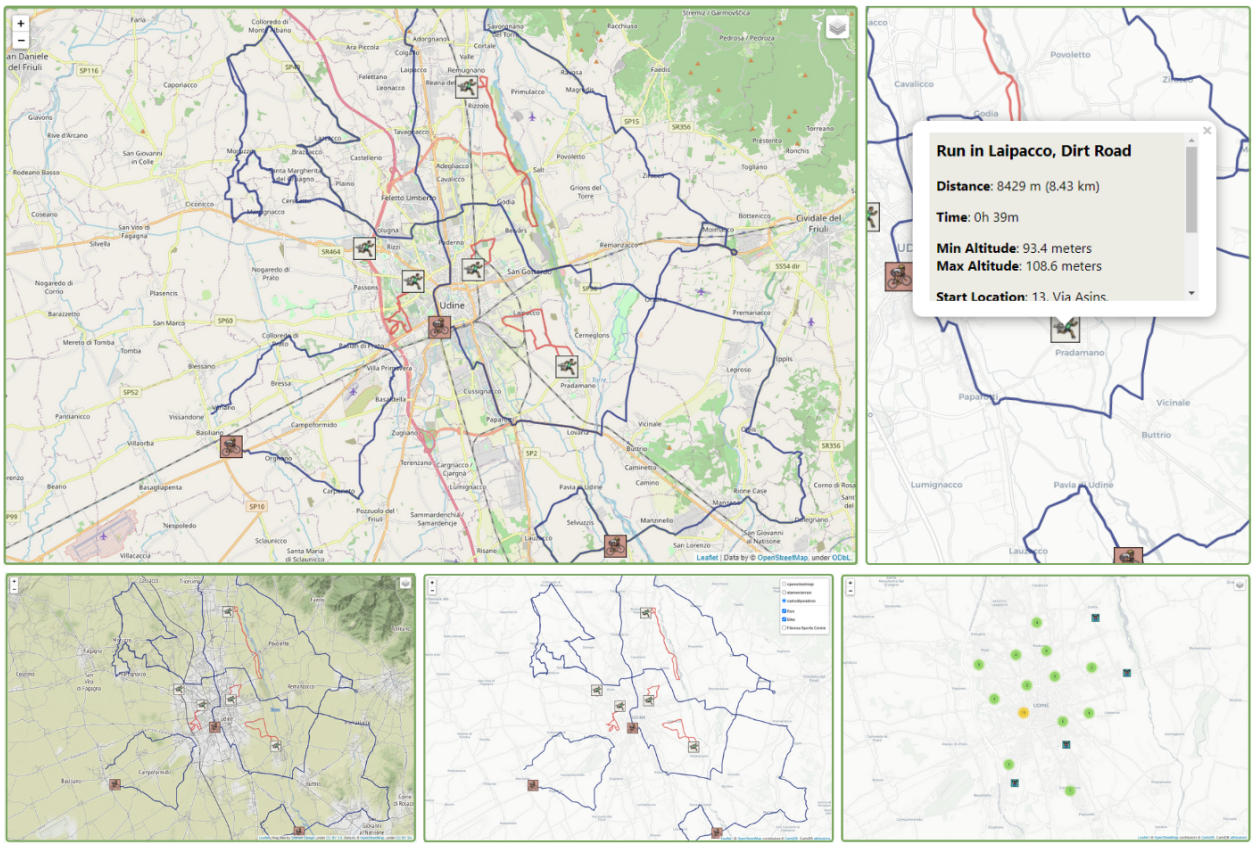


Figure 3: Interactive Map with Strava Data

4 Analysis of House Sale and Rent Cost in Prov of UD

4.1 Objective

The aim of this section is to perform descriptive spatial statistical analyses, both from a global and local perspective, on the price of real estate in the province of Udine.

The objective is to understand if there is an autocorrelation for the price of houses for sale and for rent.

4.2 Data

The data used for the analysis, updated to December 2021, are the property of *Immoboliare.it* and can be found here.

It should be stressed that these data are **only used for this analysis and specific project**, but should not be distributed in any way. As we read in the Terms of Service (in Italian) of Immobiliare.it, in fact:

> *The Holder is the legitimate owner of the Website and the App, the pages they include, the information or elements contained in the texts, documents, photographs, drawings, graphics, databases, software, logos, trademarks, trade names or other signs protected by intellectual or industrial property rights.*
>
> *Unless expressly authorised by the Holder or third parties holding the corresponding rights, or unless legally authorised, the User may not reproduce, transform, modify, decode, distribute, rent, make available to anyone or allow access through any form of communication to the public of any of the elements mentioned in the previous point.*

4.3 Methodologies and Technologies

The first part of the analysis, of an exploratory nature, was carried out in Python. Choropleth maps were created, both in static and interactive versions using `folium`. In this way it is possible for the user to explore the territory of the province of Udine and see how the price of houses and rents changes.

The second part of the analysis, of a statistical nature, was performed using R. Two types of analysis were executed: global and local. However, both required a common preparation, which is briefly outlined below:

- After importing the data, the centroid of each one of the municipalities was identified.
- Then, the different neighbourhood possibilities were explored, i.e. *k-nearest neighbours*, *critical cut-off neighbourhood* and *contiguity-based neighbourhood*.

- Spatial weights were defined, one for each neighbours list previously created.

After this data preparation, a *Moran's I test* of spatial autocorrelation (global analysis) and a local analysis through the *Moran scatterplot* and the *Local Moran's I* were performed.

Each topic and method was explained before its use, and each result was discussed.

4.4 Final Result

Figure 4 shows the results obtained in the first part of the analysis. It is recommended to view the interactive maps on the website for a more enjoyable use of the content.

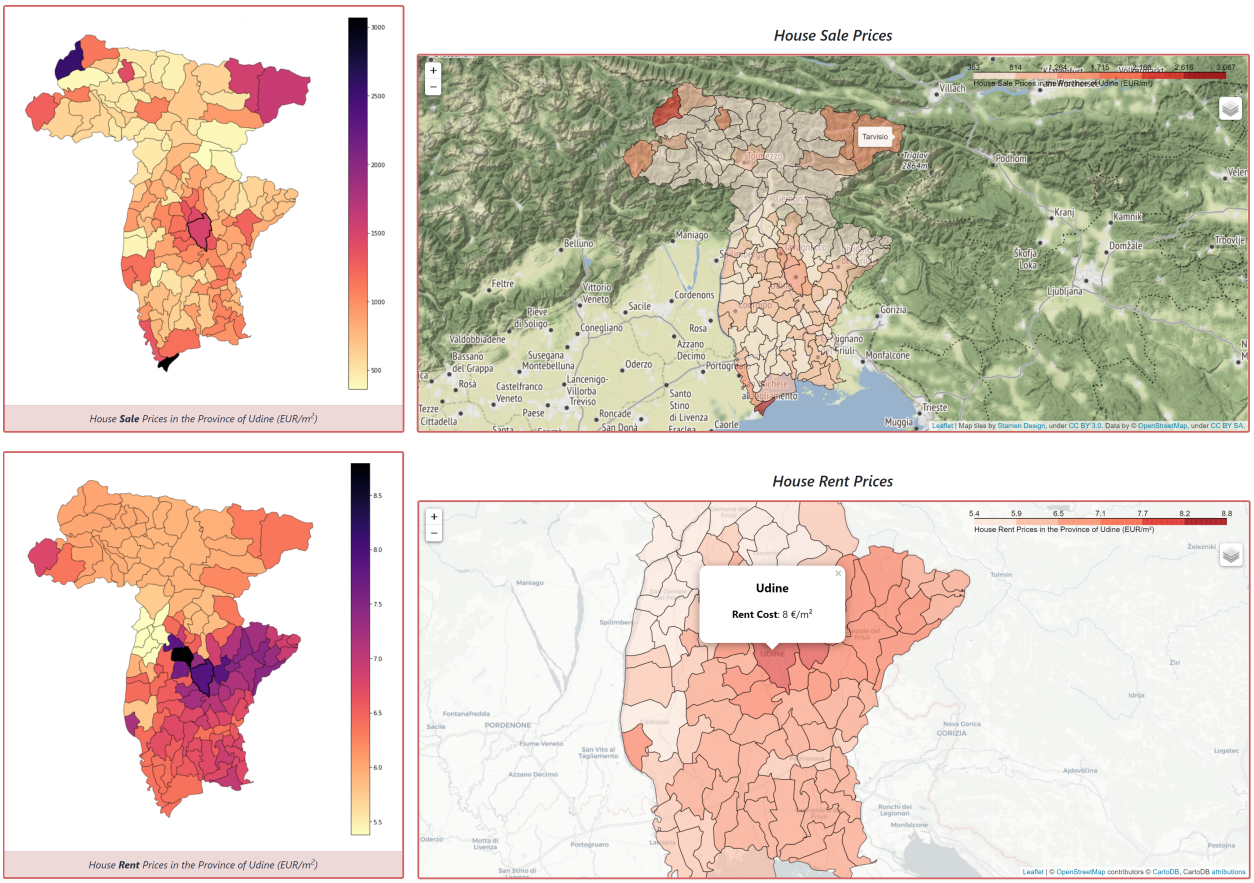


Figure 4: Interactive Map with Strava Data

Regarding the analysis in R, a positive (global) spatial autocorrelation, although not too strong, was found for house prices. For rents, on the other hand, the positive spatial autocorrelation detected was strong.

It is recommended to explore the notebook to follow the analysis step by step and to examine other results.