Elisa Paolazzi [MAT. 224424]
elisa.paolazzi-2@studenti.unitn.it
MSc Data Science

Discovering Dolomiti peaks

Geospatial Analysis and Representation for Data Science course final project report

Introduction and project description

The project aims to explore the Dolomiti area through the methodologies discussed in the *Geospatial Analysis and Representation for Data Science* course. Through the various analyses performed, we attempt to apply the lectures' arguments and other related insights in order to find useful findings about the area. For the latter, in fact, geospatial data are not often investigated.
The project is divided into five sections: the first four sections are represented by Python notebooks and aim at investigating the area through the analysis of geographical data in relation to administrative borders, data concerning infrastructures and facilities (such as ski slopes and hotels) and data capable of representing the real surface of the territory (such as raster). On the other hand, the last section addresses the spatial statistics analysis conducted on an interesting portion of our data (i.e. ski pass prices) using R.
The scripts and the data for the whole project are available in the GitHub repository; the main analyses can also be visualised (without the need to download the repository's folder) at this address.

Data source and licences

The data involved in the project and their licences are described in the following points:

- `I_nove_Sistemi_delle_Dolomiti_UNESCO.kml`: georeferenced data of the nine Dolomiti systems. These data were downloaded from the integrated map (built with Google My Maps) of the official Dolomiti website. However, the licence applied to this data is not clear as it is not specified in section *Amministrazione trasparente | Altri contenuti – Accessibilità e Catalogo di dati, metadati e banche dati*.
  Still, the Geo guidelines section of the Google Brand Resources Center specifies that all uses of Google Maps, Google Earth and Street View content must provide attribution to Google and, where applicable, data providers.

- `Limiti01012021_g`: Confini delle unità amministrative a fini statistici, collection of georeferenced datasets concerning the administrative boundaries of Italian regions, provinces and municipalities. The columns of the datasets contain information regarding denominations, ISTAT codes, area and geometry of each administrative unit.

Creative Commons CC Attribuzione 3.0 Italia (CC BY 3.0 IT)[1]

- `Ski_slopes_data`: Registro provinciale delle piste da sci, georeferenced dataset containing information about geometry, id, area and name of individual province of Bolzano ski runs.
  Creative Commons CC0 1.0 Universale - Public Domain Dedication (CC0 1.0)[2]

- `Ski_slopes_data`: Registro provinciale degli impianti di risalita, georeferenced dataset containing information about geometry, id, length and name of individual province of Bolzano ski lifts.
  Creative Commons CC0 1.0 Universale - Public Domain Dedication (CC0 1.0)

- `alto_adige.pbf`: data containing OpenStreetMap data extracted by Wikimedia Italia.
  Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)[3]

- `Raser_apline_region`: Ombreggiatura Alpi (DTM 20m), Alpine region's Elevation Digital Terrain Model Hillshade.
  Creative Commons CC0 1.0 Universale - Public Domain Dedication (CC0 1.0)

- `Castelrotto_raster`: Modello Digitale del Terreno dell'Alto Adige(DTM 2,5m), cutted province's Digital Terrain Model (DTM).
  Creative Commons CC0 1.0 Universale - Public Domain Dedication (CC0 1.0)

Section 1: Dolomiti data exploration

In the file `1_dolomiti_intro.ipynb` the main character of the analysis, the Dolomiti territory, is firstly presented and briefly described. In addition to that the partition of this area, which is divided into nine mountain systems, is also explained.
Afterwards, using the geopandas library, we import as GeoDataFrame the KML file containing the shape of the Dolomiti area. In particular, our GeoDataFrame has three columns: Name, which contains the names of the Dolomiti systems; Description, which contains metadata such as the link to the system's web page; and, finally, the column containing the geometries of the systems.
We then check the CRS (Coordinate Reference Systems) data: it is WGS84 (World Geodetic System 1984) latitude-longitude projection, which has the degree as unit.
After ensuring the validity of the geometry we plot the data and decide to add a column containing the area of each system, which will be useful for future analysis.

---

[1] *Creative Commons CC Attribuzione 3.0 Italia (CC BY 3.0 IT)*: it is possible to share, reproduce, distribute, communicate to the public, publicly display, perform and act out this material by any means and in any format; to modify, remix, transform the material and rely on it for personal works and for any purpose, including commercial purposes.
[2] *Creative Commons CC0 1.0 Universale - Public Domain Dedication (CC0 1.0)*: With this licence, the work is dedicated to the public domain by waiving all copyrights worldwide as provided by copyright law. It is possible to modify, distribute and use the work, even for commercial purposes, without asking any permission and without citing any author
[3] *Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*: it is possible to share, reproduce, distribute, communicate to the public, publicly display, perform and act out this material by any means and in any format; to modify, remix, transform the material and rely on it for personal works and for any purpose, including commercial purposes, distributing the contributions under the same licence as the original material.

Before the calculation it is necessary to momentarily convert the GeoDataFrame into the EPSG which has metres (32632) as its unit using the geopandas `to_crs()` function. It is then possible to calculate the area using the geopandas' `area()`, which returns square metres values.
We calculate the total area of the whole Dolomiti territory by converting the result into square kilometres (dividing by 10^6).
Some visualisations are then created in order to show the shape and position of the nine systems:

- the first is a static map created using the `plot()` function and adding a contextly basemap, useful to better understand the surface characteristics of the area.
- The second is an interactive map built using folium in which it is possible to better explore the area by zooming and moving the focal point. To centre the position of the initial map, the centroids of the systems are calculated and the x and y coordinates are averaged (these values will represent the centring of our map).
- The last visualisation of this section is an interactive map constructed using the geopandas `explore()` function. This function is also built on top of folium, clearly simplifying its syntax but limiting its functionality. The map shows once again the Dolomiti area. Clicking on the individual systems a pop-up box containing all the information available on each system (name, description, area) is also displayed.

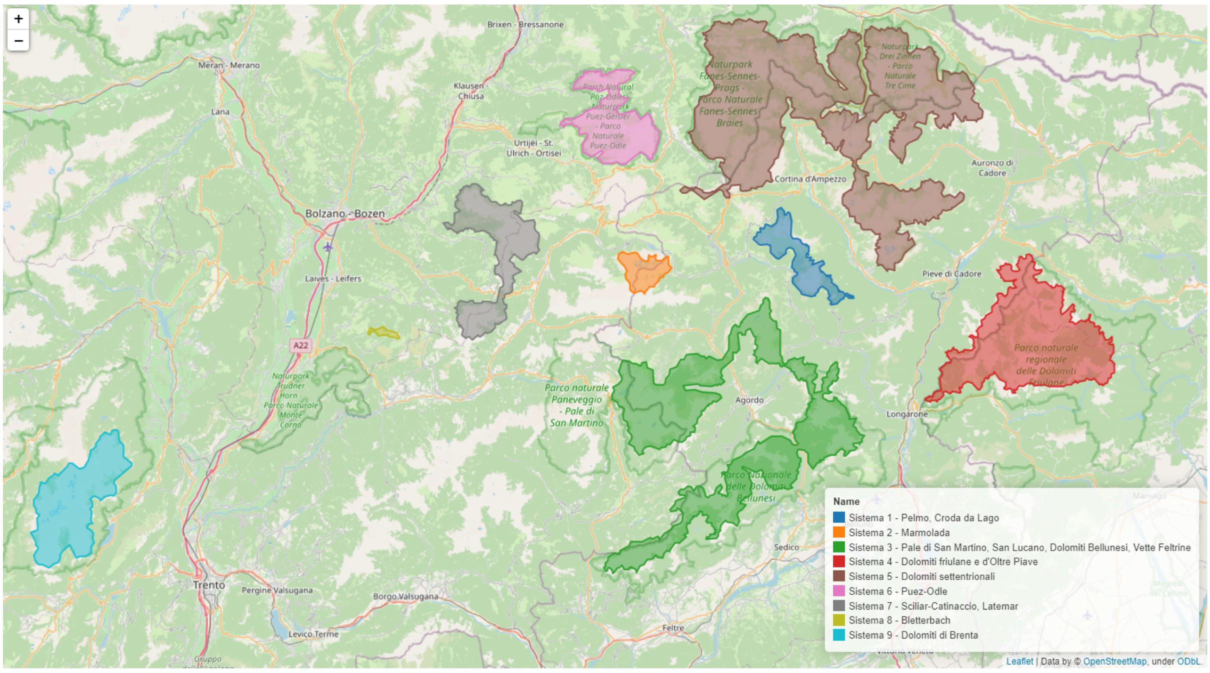

Figure 1: *The nine Dolomiti system's map*

The last point in this section consists of saving the obtained GeoDataFrame (containing the useful column "area" and "centroid"). We then attempt to save it in the most straightforward way, using geopandas' `to_file()` with the desired driver. However, the cell returns a TypeError. Investigations on the error type reveals that the geopandas `to_file()` function only supports one geometry per feature and our GeoDataFrame have both Polygon and Multipolygon as geometries (this means that some systems are composed of only one shape, while others of more than one). For this reason we simply save the GeoDataFrame in CSV format, so that the file can be imported and converted as a GeoDataFrame in the next notebooks.

Section 2: Dolomiti regions, provinces and municipalities analysis and ranking

In this section's notebook (`2_dolomiti_reg_prov_mun_analysis.ipynb`) we will build three interesting ranking lists indicating the percentage of the Dolomiti's peaks contained in the regions, provinces and municipalities involved in the area.

First of all, it is necessary to import the data on the Dolomiti systems (previously saved in CSV format) with the pandas' `read_csv()` function. The DataFrame is then converted into GeoDataFrame thanks to the geopandas' `GeoDataFrame()` function in which the column containing the geometry and the EPSG has to be specified.

It is now essential to import data about regions, provinces and municipalities: we use Limiti01012021_g, the ISTAT data used also during the course lectures.

A number of visualisations are then shown to help the contextualisation of the Dolomiti area on the regions that contain it. In particular, the custom function `regions_systems_percentage()` is used to calculate the percentage of Dolomiti's territory in each region that includes them, in particular the function:

- takes as input a region name and the Dolomiti's GeoDataFrame;
- uses geopandas `clip()` to cut the shape of the Dolomiti on the single region shape;
- recalculates the area of the portion of the Dolomiti included in the region shape;
- outputs the percentage on the total by dividing the value obtained by the total area of the Dolomiti and multiplying by 100.
  Note that the total Dolomiti's area is obtained by using geopandas' `dissolve()` function on the GeoDataFrame and recalculating the geometry's area.

The function is then applied to each region using a for loop and the results are saved in a sorted DataFrame.

In a similar way, the percentages of Dolomiti's territory for each province are also obtained and saved in a DataFrame. This time we also investigate the percentage of each of the 9 systems in each province. The custom function `province_systems_percentage()`:

- takes as input a province and a system's name;
- uses geopandas `clip()` to cut the shape of the single system on the shapes of the single province;
- recalculates the area of the portion of the system included in the province shape;
- outputs the percentage on the total by dividing the value obtained by the total area of the single system and multiplying by 100.

In the analysis of each province, the municipalities that contain Dolomiti area are then investigated. The function `mun_systems_area()` allows to obtain the amount of Dolomiti's area contained in each municipality and the percentage with respect to the total area:

- it takes as input the municipality's name and the the Dolomiti's GeoDataframe;
- uses geopandas `clip()` to cut the shape of the Dolomiti territory on the considered municipality shape;
- recalculates the area of the portion of the Dolomiti included in the municipality shape;
- outputs the Dolomiti area contained (in squared km) and the percentage on the total by dividing the value obtained by the total area of the single system and multiplying by 100.

As the previous, these values are stored in a DataFrame.

Using the three DataFrames obtained (investigating regions, provinces and municipalities of the Dolomiti) it is finally possible to construct a rank list showing the places with the highest concentration of Dolomiti mountains.
Several static maps are also displayed in this notebook. These maps are obtained using the `plot()` function and contextly's basemaps. In two cases the maps present some labels:

- the first one shows the Dolomiti and the regions containing them. Here the region labels are placed on the representative points of the individual regions whose geometry is calculated by the custom function `representative_points_for_labels()` (this function uses geopandas' `representative_point()`).
- The second shows the Dolomiti and the provinces containing them. The coordinates on which the label of the provincial capitals are placed are obtained by the custom function `city_coordinates_label()`, which uses reverse geocoding (in particular using geopandas' `tools.geocode()` with ArcGIS as provider).

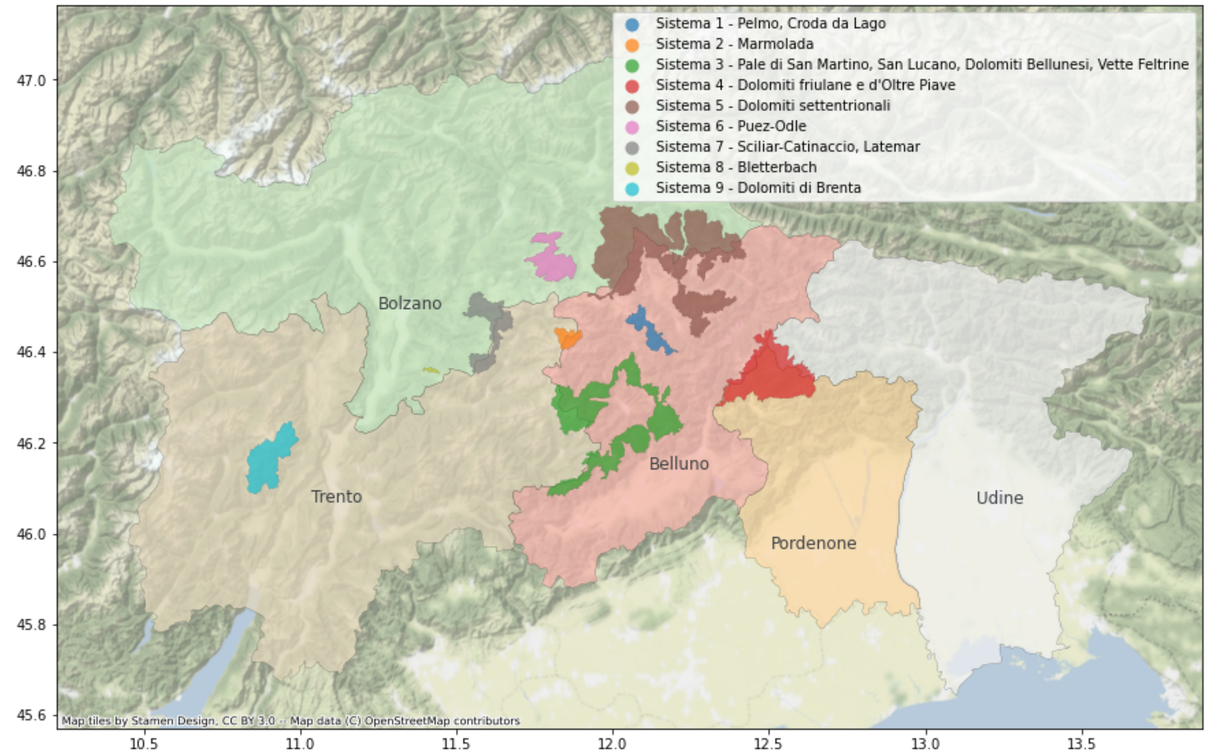

Figure 2: *Dolomiti provinces map*

Section 3: Alto Adige/Südtirol ski area exploration and analysis

This section aims to explore the ski area of the Dolomiti in the province of Bolzano.
First, the data of the Dolomiti systems and administrative borders (provinces and municipalities) are re-imposed. Beside this data we import and statically plot the new GeoDataFrames of the ski slopes and ski lifts, which can be downloaded from the province's Open Data Portal .
Once these data have been imported it is possible to calculate the surface of ski slopes that the province contains. We then clip the shapes of the ski slopes (Polygon) and ski lifts (Linestring) on the shape of the province's Dolomiti area). By doing this it is possible to focus on the ski areas in that territory and calculate their impact on the total number of slopes in South Tyrol. By recalculating the

area of the slopes contained in the Dolomiti's territory we can in fact understand what percentage of the province's slopes are included in this area.

After this first exploration onto the ski resorts, we obtain the PBF file of the province in order to extract the tourist accomodation structures (hotels, apartments, guest houses...). The aim now is to understand how the structures are placed with respect to the ski areas.

In particular, we calculate the starting points of each given ski lift Linestring in order to extract the starting points of each cable way (from which users can begin their skiing activities). After this step we compose two cluster visualisations using folium: the first maps the ski lift starting point clusters in the area, the second maps tourist accommodation clusters in the area.

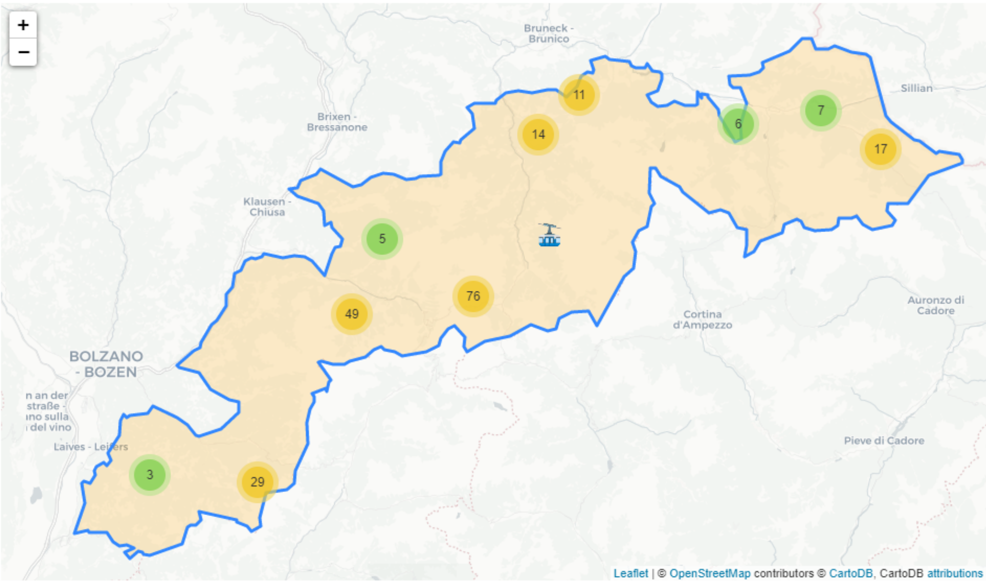

Figure 3: *Ski lift starting points cluster map visualisation*

After displaying the data we obtain OpenStreetMap's nodes and edges of the area from the PBF file using the `get_network()` function. We then obtain the `networkx` street network using the `to_graph()` function and adding the edge speeds and travel times.

Finally we are able to build a function that is capable of finding the closest tourist accommodation to a chosen ski starting point. The `nearest_accomodation_path()` function:

- takes as input the chosen starting point OBJECTID and the chosen type of tourist accommodation (hotel, alpine hut, apartment, ...);
- by means of the `near()` inner function obtains the nearest tourist accommodation (of the chosen type) to the chosen starting point using shapely's `nearest_points()`;
- uses the osmnx `get_nearest_node()` function to find the nearest network points to the chosen starting point and to the nearest tourist accommodation;
- calculates the shortest path between the two points with osmnx `shortest_path()` function;
- calculates the route duration using the `timedelta()` function from the datetime library
- builds a starting folium map with the route;
- adds ski lift starting point and nearest tourist accommodation markers to the map;
- returns the map and the ruote duration.

Figure 4: `nearest_accomodation_path()` function output

Note that we could have had the same result by directly calculating the shortest paths between the chosen ski lift starting point and all tourist accommodations (of the chosen type) and taking the shortest route. Extracting the nearest network points for each tourist accommodation (using osmnx's `get_nearest_node()`) and calculating all shortest paths with the starting point (using `shortest_path()`) is in fact resource demanding. For this reason, in the construction of the function, it was decided to first find the closest tourist accommodation to the chosen starting point and to compute only the shortest path between them.

Section 4: Alto Adige/Südtirol raster data analysis

This section (`4_dolomiti_alto_adige_raster.ipynb`) is dedicated to the overview and analysis of the raster data made available by the Geocatalog of the Province of Bolzano. A first approach to the analysis of these data allows us not only to better visualise the area we are considering and its landforms, but also to investigate in depth the third dimension, namely the altitude.

To begin with, we use Web Map Services (province's geoservices) and explore the data made available through them. Among this data, it is possible to identify orthophotos of different years. We decided to display the most recent one using `MemoryFile()` and `show()` from the `rasterio` library. Unsatisfied with the completeness of the orthophoto, we then decide to display the less recent ones as well, in order to get an idea of what the area we are investigating actually appears. We then plot the image in the different RGB colours using the rasterio `show()` function and appropriate *cmap*.

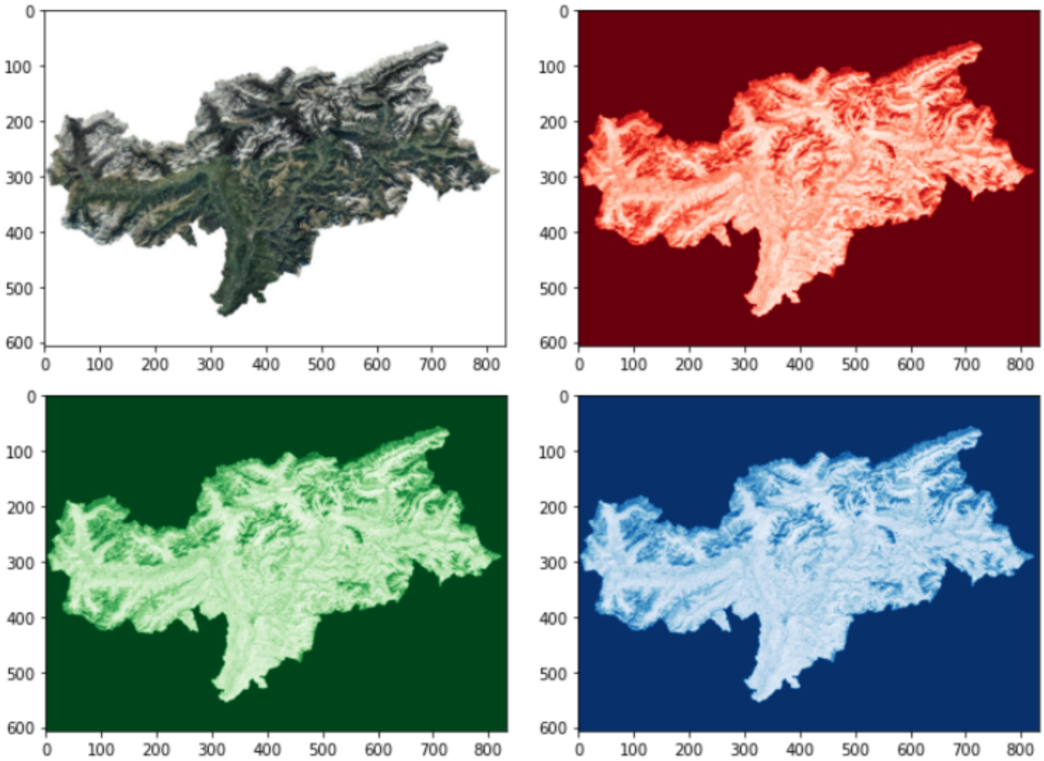

Figure 5: *Province of Bolzano orthophoto RGB plots*

In order to start the analysis part, we download the data concerning the Elevation Digital Terrain Model Hillshade of the Alpine region with the aim of cutting it on the shape of the area of our interest (the Dolomiti area of the South Tyrol). After downloading these data we need to read them with the rasterio `open()` function, subsequently checking the metadata (including crs, height and width of the image). However, the GeoDataFrame as it is, cannot be passed on to the rasterio function for cutting the DTM on it. A fundamental step is then to extract the features, transforming them, in a sense, into Geojson format. In order to do this we use the custom function seen in class `getFeatures()` which, taking the GeoDataFrame as input, uses the `json` library to extract features from its geometry. Finally, we can crop the area of interest from the DTM of the Alpine region, using the rasterio function `mask()`, to which we give the DTM raster and the features extracted from the GeoDataFrame as input. We can then update the metadata of the new raster and save everything in a TIF file and plot it.

The second part of the notebook aims instead to implement the third dimension (altitude) extraction to investigate the height difference of the longest ski lift in the area. First of all, the cableway location within the municipalities is identified. Understanding in which municipality the cableway is located helps us to avoid downloading and dealing with large DTMs: this information allows us to download only the portion of DTM of the municipality (Castelrotto in this case). The Castelrotto's DTM TIF file is then opened with rasterio and its metadata are checked. We can finally extract the altitude values and match them to the coordinates corresponding to the cableway Linestring. Finally, the cable car altitude values are saved in a DataFrame and then visualised in a representative line plot.

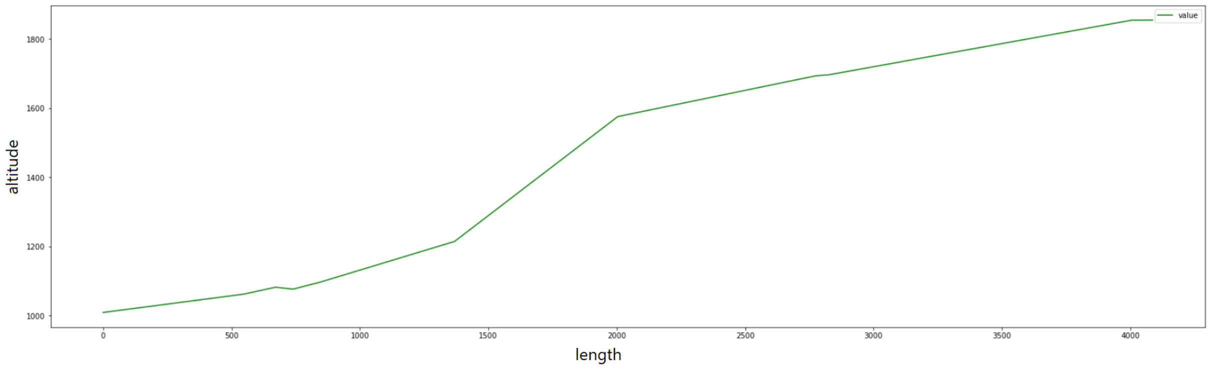

Figure 6: *Alpe di Siusi cableway high difference plot*

Section 5: Geospatial Statistics

This R notebook `5_spatial_statistics_analysis` aims to perform a spatial statistical analysis concerning ski pass prices in the area (the Dolomiti area in the province of Bolzano), investigating whether there are any relationships between them and their position. Since we are unable to find a complete resource containing the costs of the ski pass for the Dolomiti area, the data used in this notebook was created in the file `ski_area_cost.ipynb` by consulting the price lists on the websites of the various resorts.

We start our spatial statistics analysis defining the neighbours by means of the K-Nearest neighbours and Critical cut-off neighbourhood criteria.

Once the neighbourhood relationships between observations have been defined, the spatial weights matrix can be created. After this procedure we perform Moran's I test of spatial autocorrelation with reference to the variable corresponding to the price of ski passes. The test is then performed under both the normality, and randomisation assumption and under permutation bootstrap.

The second part concerns spatial descriptive statistics for areal data, i.e. local spillover analysis. Here we investigate the local spatial autocorrelation by the means of the Moran scatterplot and the local Moran's I, finding slightly more significant results compared to the global analysis.