Explore the world by bike

Paweł Kryzia

24 January 2021

INTRODUCTION

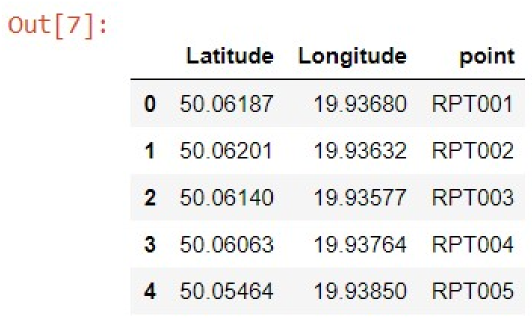
My project is aimed at people who like to explore the world by bike (also for car tourists). The idea is that on the basis of the pre-prepared route, the notebook will find (using Foursquare) tourist attractions, accommodation or restaurants located on or near the route (depending on whether you are traveling by bike or by car you will be able to increase or decrease the radius of the search)
In my project, for example, I will take into account the route from Krakow in Poland to the Italian capital Rome (I would like to visit Rome).
In the second part, I will use Foursquare to explore the monuments of Rome. Using k-mean clustering, I will group the found monuments, taking into account their location (coordinates) to as many groups as many days as I want to visit Rome.
Of course, the project will be versatile enough to change input data to explore other routes and cities.

OBJECTIVES

- finding the nearest monuments, food and accommodation for any point of the route
- grouping the monuments found in Rome to as many clusters as we choose days, based on their location

DATA

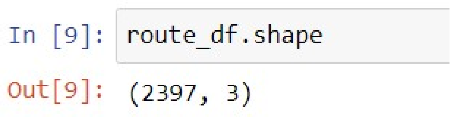
The data used in the project is firstly the coordinates of the waypoints downloaded from a GPX file prepared in one of the services that allow you to create and download such routes. The GPX file was previously converted to CSV format and uploaded to the notebook in this form. The file contains only waypoints and their coordinates (latitude and longitude), so we can easily convert it to a dataframe. Of course, the second source of data will be Foursquare.



Out[7]:

	Latitude	Longitude	point
0	50.06187	19.93680	RPT001
1	50.06201	19.93632	RPT002
2	50.06140	19.93577	RPT003
3	50.06063	19.93764	RPT004
4	50.05464	19.93850	RPT005

There are as many as 2,397 route points

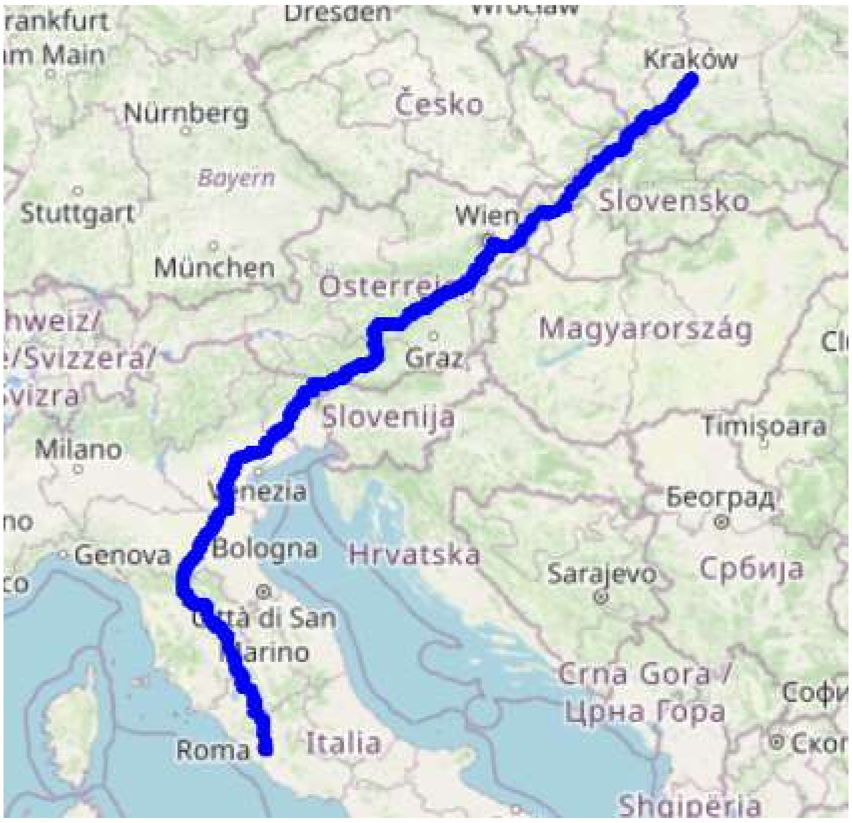


```
In [9]: route_df.shape

Out[9]: (2397, 3)
```

EXPLORING A ROUTE

At the beginning, using the folium, we draw the entire route on the map of Europe based on a GPX route converted to a CSV file



Next, for the selected point of the route, using Foursquare, I search for monuments, food and accommodation. Using the Geolocator module, I can also search for objects for specific places along the route or next to the route. Selecting coordinates:

```
n = 1750 #point number on the route
latitude = route_df['Latitude'][n]
longitude = route_df['Longitude'][n]
```

or

```
address = 'Gaio, Italy'
geolocator = Nominatim(user_agent="cracow_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
```

I use foursquare explore with categoryId parameter. I found the categories that interest me at: https://developer.foursquare.com/docs/build-with-foursquare/categories/

- **Historic Site**
  4deefb944765f83613cdba6e
- **Museum**
  4bf58dd8d48988d181941735
- **Food**
  4d4b7105d754a06374d81259
- **Hotel**
  4bf58dd8d48988d1fa931735

Search results for the sample 1750 waypoint:

- monuments

Out[56]:

	name	categories	lat	lng
0	Palazzo Del Vignola	Historic Site	44.604799	11.348963
1	La Grotta 1570	Historic Site	44.547189	11.351965
2	Rocca Isolani	Historic Site	44.623295	11.490487

- food

Out[58]:

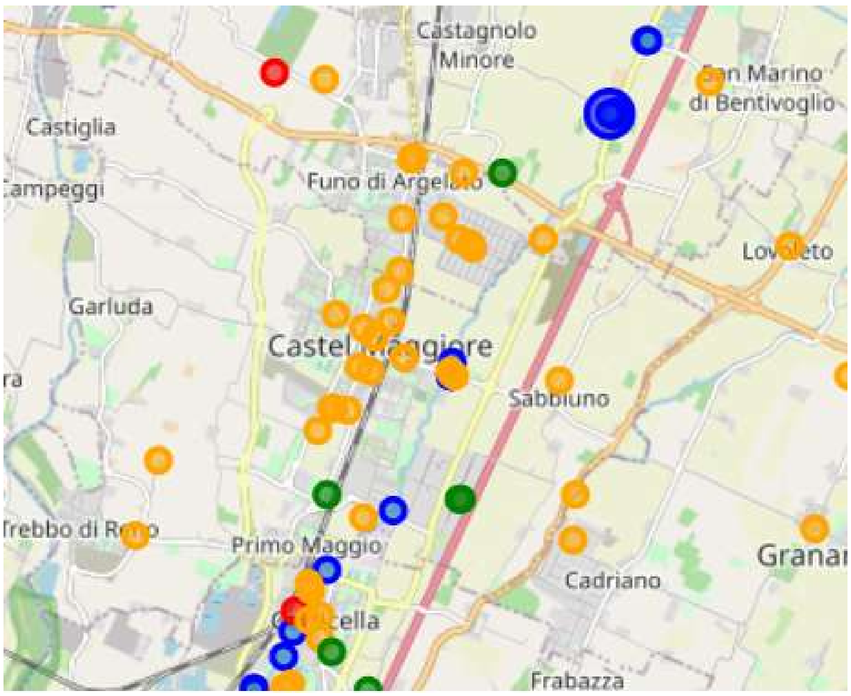
	name	categories	lat	lng
0	La Scuderia	Italian Restaurant	44.572880	11.375412
1	Trattoria del Gallo	Italian Restaurant	44.559714	11.394200
2	Villa La Torre	Italian Restaurant	44.604116	11.356408
3	Piadina sbarazzina	Fast Food Restaurant	44.589166	11.368166
4	Villa Orsi	Italian Restaurant	44.589528	11.374251

- accommodation

Out[60]:

	name	categories	lat	lng
0	Ramada Encore Hotel Bologna Fiera	Hotel	44.533210	11.367620
1	Imperial Hotel Bologna	Hotel	44.533344	11.370654
2	Hotel Marconi	Hotel	44.594178	11.383175
3	B&B Hotel Bologna	Hotel	44.559101	11.376876
4	Hotel Nettuno	Hotel	44.559086	11.376791

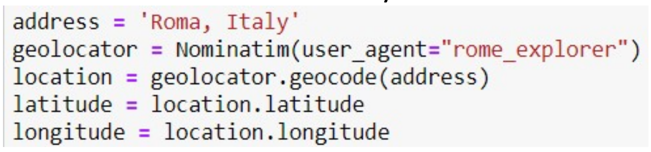
Now I can draw a map with the found places marked. Monuments marked in red, food in orange, and accommodation in green. Blue are waypoints.



SIGHTSEEING IN ROME

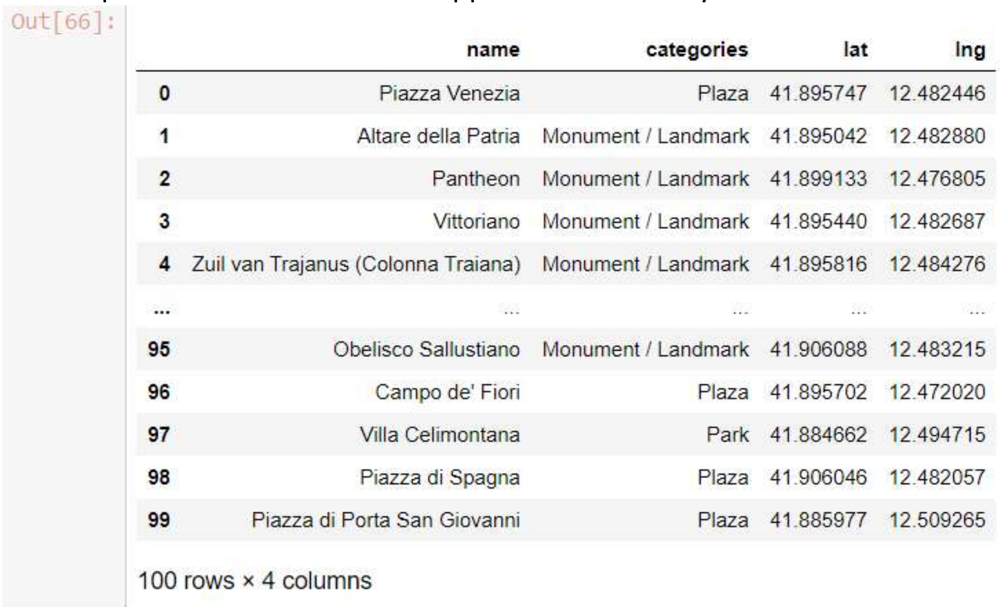
The second part of the project is not directly related to the first. The idea is to use Foursquare to find monuments in the chosen city (in my example it is Rome). Foursquare limits the results to 100, so we have 100 places to visit. Assuming that we want to explore the city for 6 days we will group (based on the coordinates) these 100 places into 6 groups using k-means Clustering.

First, we specify the coordinates of the selected city:



```
address = 'Roma, Italy'
geolocator = Nominatim(user_agent="rome_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
```

Next, we specify the parameters for searching places by specifying the limit, radius and query (in this case, we select 'monuments'). Then we compose the appropriate URL and retrieve the results.
In the next step I filter the results so its appear in a clear way and I have a table:

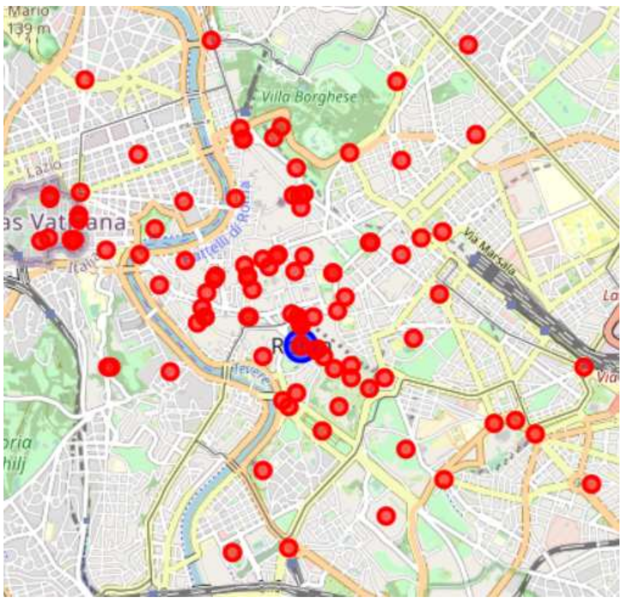


Out[66]:

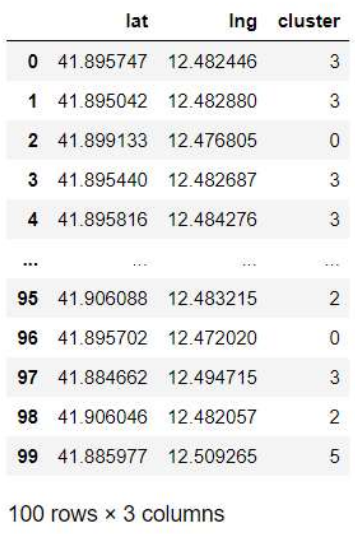
	name	categories	lat	lng
0	Piazza Venezia	Plaza	41.895747	12.482446
1	Altare della Patria	Monument / Landmark	41.895042	12.482880
2	Pantheon	Monument / Landmark	41.899133	12.476805
3	Vittoriano	Monument / Landmark	41.895440	12.482687
4	Zuil van Trajanus (Colonna Traiana)	Monument / Landmark	41.895816	12.484276
...	...	...	...	...
95	Obelisco Sallustiano	Monument / Landmark	41.906088	12.483215
96	Campo de' Fiori	Plaza	41.895702	12.472020
97	Villa Celimontana	Park	41.884662	12.494715
98	Piazza di Spagna	Plaza	41.906046	12.482057
99	Piazza di Porta San Giovanni	Plaza	41.885977	12.509265

100 rows × 4 columns

and map:

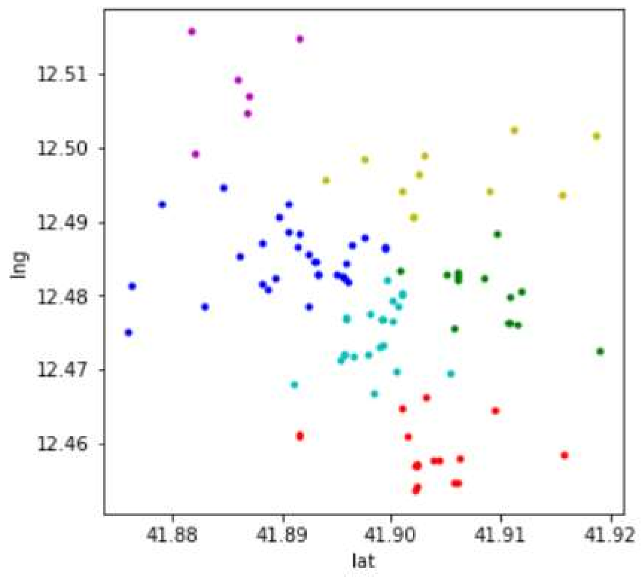


The last important step will be to group the results into a selected quantity (in my example, 6) by combining them into groups based on location. I will use for this k-means Clustering.
After preparing the dataframe and specifying the parameters, we get the result.
Visualization:

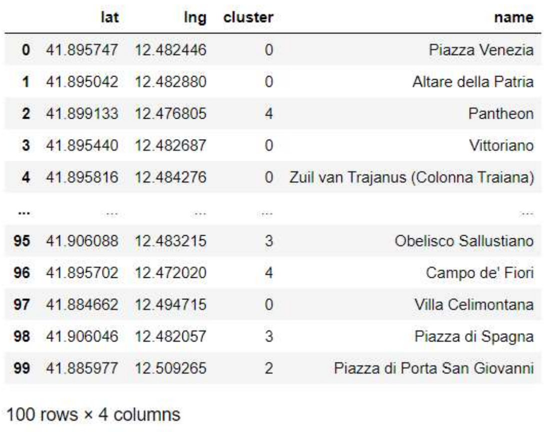


	lat	lng	cluster
0	41.895747	12.482446	3
1	41.895042	12.482880	3
2	41.899133	12.476805	0
3	41.895440	12.482687	3
4	41.895816	12.484276	3
...	...	...	...
95	41.906088	12.483215	2
96	41.895702	12.472020	0
97	41.884662	12.494715	3
98	41.906046	12.482057	2
99	41.885977	12.509265	5

100 rows × 3 columns

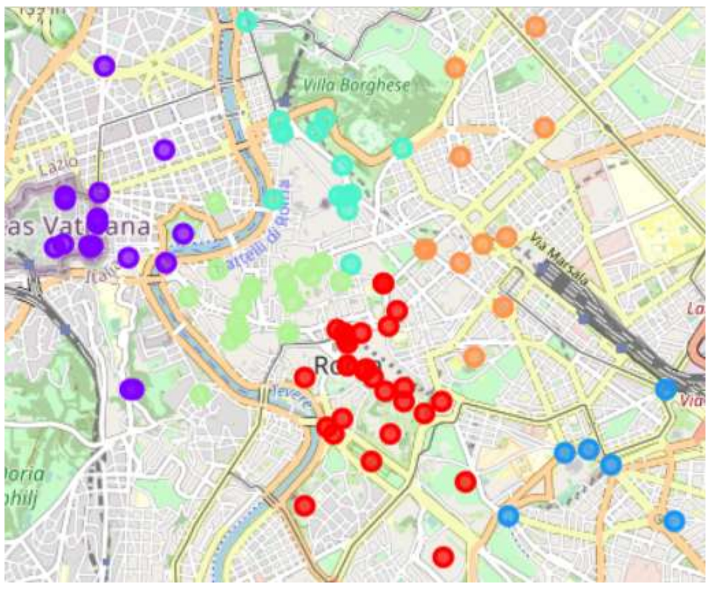


Then, after re-linking the results to the object names, we draw a map showing the color division into clusters:

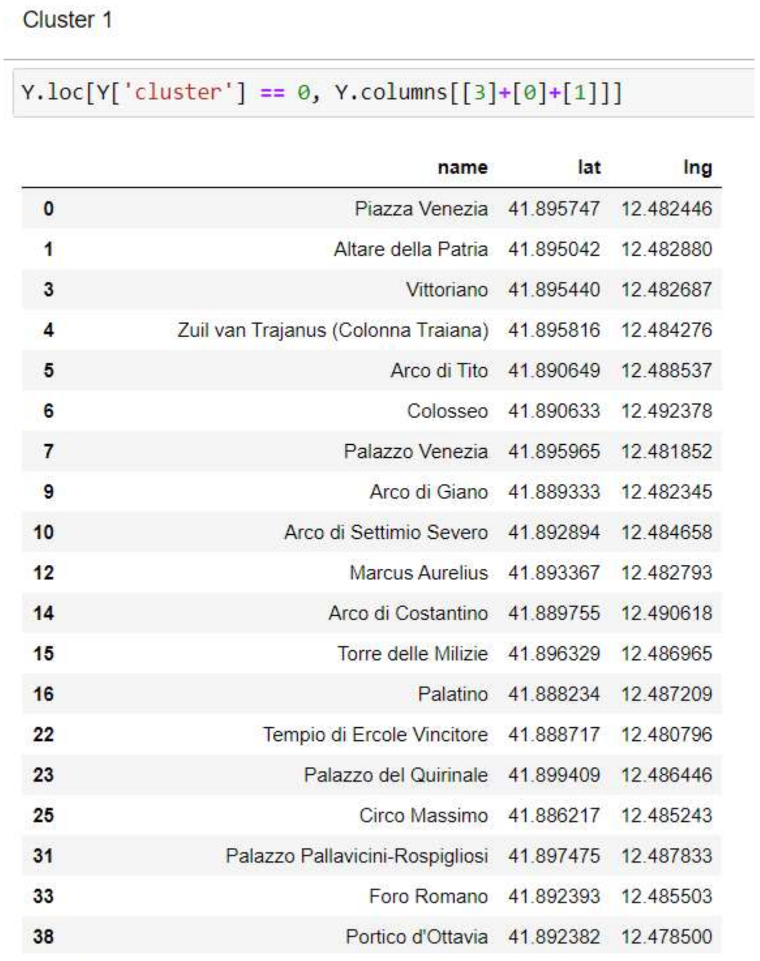


	lat	lng	cluster	name
0	41.895747	12.482446	0	Piazza Venezia
1	41.895042	12.482880	0	Altare della Patria
2	41.899133	12.476805	4	Pantheon
3	41.895440	12.482687	0	Vittoriano
4	41.895816	12.484276	0	Zuil van Trajanus (Colonna Traiana)
...	...	...	...	...
95	41.906088	12.483215	3	Obelisco Sallustiano
96	41.895702	12.472020	4	Campo de' Fiori
97	41.884662	12.494715	0	Villa Celimontana
98	41.906046	12.482057	3	Piazza di Spagna
99	41.885977	12.509265	2	Piazza di Porta San Giovanni

100 rows × 4 columns



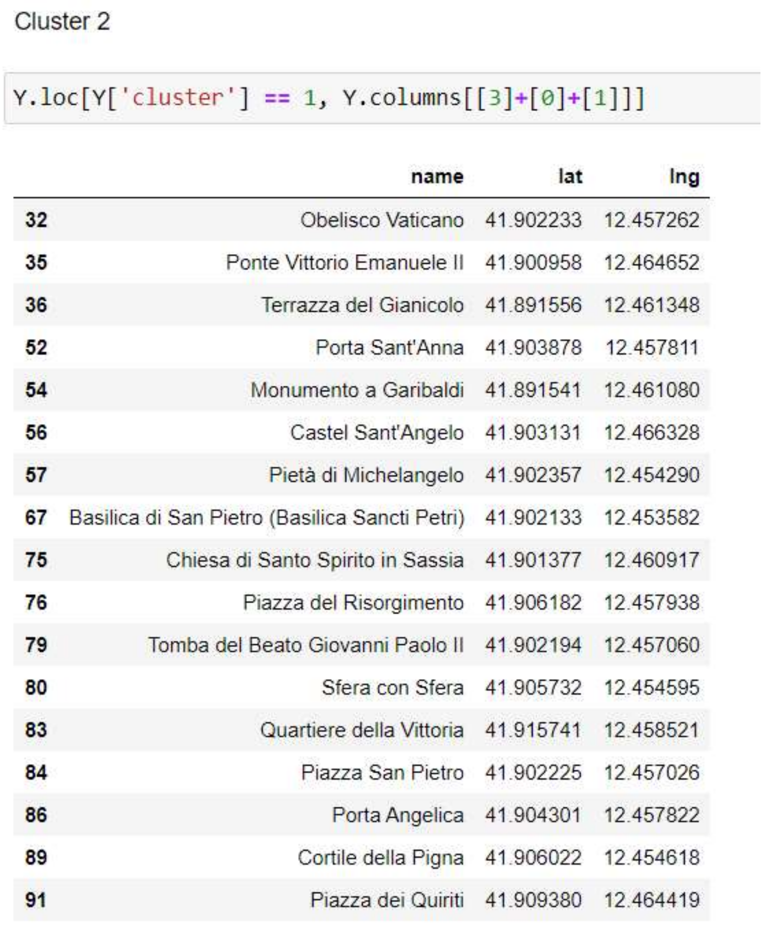
Finally, we print out the individual clusters:



Cluster 1

```
Y.loc[Y['cluster'] == 0, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
0	Piazza Venezia	41.895747	12.482446
1	Altare della Patria	41.895042	12.482880
3	Vittoriano	41.895440	12.482687
4	Zuil van Trajanus (Colonna Traiana)	41.895816	12.484276
5	Arco di Tito	41.890649	12.488537
6	Colosseo	41.890633	12.492378
7	Palazzo Venezia	41.895965	12.481852
9	Arco di Giano	41.889333	12.482345
10	Arco di Settimio Severo	41.892894	12.484658
12	Marcus Aurelius	41.893367	12.482793
14	Arco di Costantino	41.889755	12.490618
15	Torre delle Milizie	41.896329	12.486965
16	Palatino	41.888234	12.487209
22	Tempio di Ercole Vincitore	41.888717	12.480796
23	Palazzo del Quirinale	41.899409	12.486446
25	Circo Massimo	41.886217	12.485243
31	Palazzo Pallavicini-Rospigliosi	41.897475	12.487833
33	Foro Romano	41.892393	12.485503
38	Portico d'Ottavia	41.892382	12.478500



Cluster 2

```
Y.loc[Y['cluster'] == 1, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
32	Obelisco Vaticano	41.902233	12.457262
35	Ponte Vittorio Emanuele II	41.900958	12.464652
36	Terrazza del Gianicolo	41.891556	12.461348
52	Porta Sant'Anna	41.903878	12.457811
54	Monumento a Garibaldi	41.891541	12.461080
56	Castel Sant'Angelo	41.903131	12.466328
57	Pietà di Michelangelo	41.902357	12.454290
67	Basilica di San Pietro (Basilica Sancti Petri)	41.902133	12.453582
75	Chiesa di Santo Spirito in Sassia	41.901377	12.460917
76	Piazza del Risorgimento	41.906182	12.457938
79	Tomba del Beato Giovanni Paolo II	41.902194	12.457060
80	Sfera con Sfera	41.905732	12.454595
83	Quartiere della Vittoria	41.915741	12.458521
84	Piazza San Pietro	41.902225	12.457026
86	Porta Angelica	41.904301	12.457822
89	Cortile della Pigna	41.906022	12.454618
91	Piazza dei Quiriti	41.909380	12.464419

Cluster 3

```
Y.loc[Y['cluster'] == 2, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
51	Obelisco Lateranense	41.886876	12.504656
66	Scala Santa	41.887042	12.507036
68	Porta Maggiore	41.891530	12.514794
88	Piazza dei Re di Roma	41.881732	12.515683
92	Piazzale Metronio	41.882147	12.499062
99	Piazza di Porta San Giovanni	41.885977	12.509265

Cluster 4

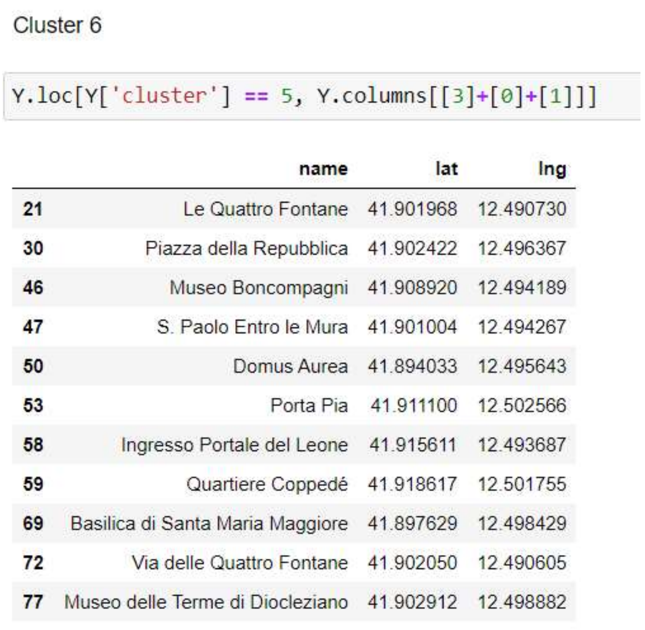
```
Y.loc[Y['cluster'] == 3, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
8	Trevi-fontein (Fontana di Trevi)	41.900844	12.483252
24	Colonna dell'Immacolata	41.905000	12.482881
26	Scalinata di Trinità dei Monti	41.905974	12.482647
27	Museo dell'Ara Pacis	41.905744	12.475521
28	Obelisco Flaminio	41.910751	12.476385
41	Villa Medici - Accademia di Francia a Roma	41.908346	12.482405
43	Porta del Popolo	41.911541	12.476027
78	Piazza del Popolo	41.910683	12.476342
81	Viale delle Belle Arti	41.919000	12.472664
82	Porta Pinciana	41.909572	12.488381
85	Obelisco Pincio	41.910853	12.479761
90	Idrocronometro	41.911763	12.480692
95	Obelisco Sallustiano	41.906088	12.483215
98	Piazza di Spagna	41.906046	12.482057

Cluster 5

```
Y.loc[Y['cluster'] == 4, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
2	Pantheon	41.899133	12.476805
11	Elefantino e Obelisco della Minerva	41.898041	12.477461
13	Colonna di Marco Aurelio	41.900930	12.480185
17	Largo di Torre Argentina	41.895797	12.476852
18	Piazza Navona	41.899239	12.473184
19	Tempio di Adriano	41.900029	12.479352
20	Piazza Colonna	41.901028	12.480237
29	Piazza della Maddalena	41.900130	12.476583
34	Obelisco di Monte Citorio	41.900611	12.478559
37	Santa Maria della Scala	41.891161	12.468144
39	Piazza Cavour	41.905422	12.469573
40	Obelisco Agonalis	41.898899	12.473165
42	Via dei Coronari	41.900488	12.469740
60	Via dei Banchi Vecchi	41.898436	12.466881
62	Piazza Farnese	41.895261	12.471209


Cluster 6

```
Y.loc[Y['cluster'] == 5, Y.columns[[3]+[0]+[1]]]
```

	name	lat	lng
21	Le Quattro Fontane	41.901968	12.490730
30	Piazza della Repubblica	41.902422	12.496367
46	Museo Boncompagni	41.908920	12.494189
47	S. Paolo Entro le Mura	41.901004	12.494267
50	Domus Aurea	41.894033	12.495643
53	Porta Pia	41.911100	12.502566
58	Ingresso Portale del Leone	41.915611	12.493687
59	Quartiere Coppedè	41.918617	12.501755
69	Basilica di Santa Maria Maggiore	41.897629	12.498429
72	Via delle Quattro Fontane	41.902050	12.490605
77	Museo delle Terme di Diocleziano	41.902912	12.498882

SUMMARY AND CONCLUSIONS

One of the two main goals of this project was to facilitate travel and at the same time to explore places that we do not know and do not know what to see, where to stay for food or accommodation. The first part of the project solves this problem with the help of Foursquare. Provides information about such locations for the selected route point or city and visualizes them on the map.
The second part was to help in visiting the monuments of the selected city. K-means clustering helped to divide the monuments found by Foursquare into clusters based on their location. The number of clusters corresponds to the number of days we have for visiting. Some clusters contain too many objects, but we can choose the most interesting ones ourselves.