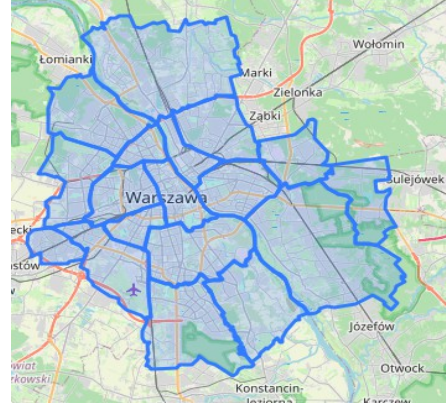
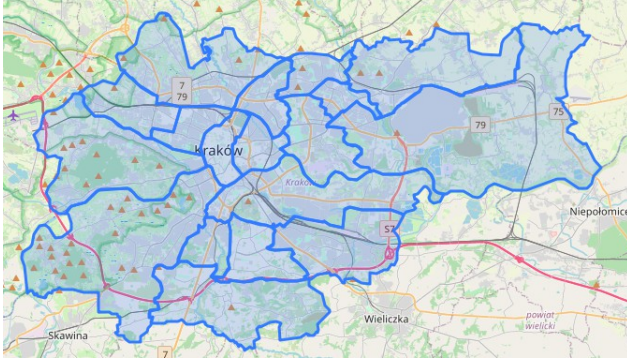


## Warsaw and Krakow -districts analysis.



# 1. Introduction.

## 1.1. Project description and main goals.

In this project, I would like to explore two polish cities' districts -Warsaw and Krakow. Warsaw is the capital and largest city of Poland. The metropolis stands on the Vistula River in east-central Poland and its population is officially estimated at 1.8 million residents within a greater metropolitan area of 3.1 million residents, which makes Warsaw the 7th most-populous capital city in the European Union. City area is 517.24 km<sup>2</sup> (199.71 sq mi).

Krakow, is the second largest and one of the oldest cities. Its population is estimated at 779 115 residents, and city area is 326.8 km<sup>2</sup> (126.2 sq mi).

In the first place, I would like to create choroplets maps with most popular types of venues for both cities. This type of maps could be useful to quickly find similar districts, for example by tourists or investors.

Then, I would like to group districts into few cluster (no assumptions about the number), taking into consideration what kind of venues could be find there. This kind of data can be useful if, for example, someone is running business in one place, and wants find similar environment in different city. Also, this information could be used, for example, by travel agency to suggest clients new destination, similar to places he/she visited previously.

As an extra task, I will try to find correlation between different types of venues in districts (is number of Italian restaurants correlated with number of coffee shops?). Maybe I will find some interesting and surprising conclusions.

## 1.2. Data description.

Following data will be used to perform analysis:

- a) Shapefiles of Warsaw and Krakow. The shapefile format is a geospatial vector data format for geographic information system (GIS) software. These files can be easily downloaded from web:
  - [https://gis-support.pl/wp-content/uploads/dzielnice\\_Warszawy.zip](https://gis-support.pl/wp-content/uploads/dzielnice_Warszawy.zip)
  - [https://gis-support.pl/wp-content/uploads/dzielnice\\_Krakowa.zip](https://gis-support.pl/wp-content/uploads/dzielnice_Krakowa.zip)
- b) From shapefiles, with help of QGIS software, I will create GeoJSONs for both cities. GeoJSON is an open standard format designed for representing simple geographical features, along with their non-spatial attributes. It is based on the JSON format. When transforming .shp to geojson we intend to use with Folium, it is important to choose correct coordinate system: WGS 84. Generally speaking, GeoJSONs for both cities have different structure, but I will use mostly two informations: district name and shape of district (given by polygon coordinates).

| Example of Warsaw geojson:   | Example of Cracow geojson:  |
|--|---|
| <pre> {   "type": "FeatureCollection",   "name": "warsaw_districts",   "crs": {     "type": "name",     "properties": {       "name": "urn:ogc:def:crs:OGC:1.3:CRS84"     }   },   "features": [     {       "type": "Feature",       "properties": {         "nazwa_dzie": "Żoliborz",         "style": {           "weight": 1,           "opacity": 0.9,           "color": "black",           "fillOpacity": 0.7,           "fillColor": "#d9f0a3"         },         "highlight": {}       },       "geometry": {         "type": "MultiPolygon",         "coordinates": [           [             [               [                 20.957550244360345,                 52.266927972075955               ],               [                 20.957595033280743,                 52.26712068328835               ],               .....             ]           ]         ]       }     }   ] }</pre> | <pre> {   "type": "FeatureCollection",   "name": "cracow_districts",   "crs": {     "type": "name",     "properties": {       "name": "urn:ogc:def:crs:OGC:1.3:CRS84"     }   },   "features": [     {       "type": "Feature",       "properties": {         "objectid": 1.0,         "mslink": 1.0,         "mapid": 3013.0,         "id_dzielni": 1.0,         "nr_dzielni": "I",         "powierzchn": 5567646.0,         "nazwa": "Stare Miasto",         "nazwa_peln": "Dzielnica I Stare Miasto",         "opis": "UCHWALA NR XCIX/1495/14 RADY MIASTA KRAKOWA",         "data_aktua": "2014/04/01",         "st_area(sh)": 5567646.1253,         "st_length(": 11730.268840426876,         "style": {},         "highlight": {}       },       "geometry": {         "type": "MultiPolygon",         "coordinates": [           [             [               [                 19.92376505386849,                 50.062551412466526               ],               [                 19.923769137430885,                 50.06260611760511               ],               .....             ]           ]         ]       }     }   ] }</pre> |

From these GeoJSON files, the most important information was district geometry, defined as type of figure (MultiPolygon), and coordinates of its vertices. These data were used to:

- defining centers of districts
  - calculating radius for defining searching area in Foursquare request
  - verification if point with given coordinates is inside or outside district
- c) For creating choropleth maps, data about districts areas were required. This type of data can be easily find in Wikipedia.
- d) I used Foursquare API to get information about venues in different districts.

## 2. Methodology.

### 2.1. Creating choropleths maps

With help of previously mentioned data, I created dataframe and dictionaries with districts centers, radiuses (to use in Foursquare request), and geometries -to verify later if venues returned from Foursquare are inside district.

| Dataframe with districts centers:                | <table> <tr> <th colspan="2">District_center</th></tr> <tr> <td><b>Żoliborz</b></td><td>[52.2688536216614, 20.985135391429377]</td></tr> <tr> <td><b>Praga-Południe</b></td><td>[52.235168690871, 21.071138918572952]</td></tr> <tr> <td><b>Mokotów</b></td><td>[52.18866196405862, 21.052814696946914]</td></tr> <tr> <td><b>Wola</b></td><td>[52.22969762335711, 20.94634273758449]</td></tr> <tr> <td><b>Wilanów</b></td><td>[52.15030830502675, 21.091139416474352]</td></tr> </table> | District_center |  | <b>Żoliborz</b> | [52.2688536216614, 20.985135391429377] | <b>Praga-Południe</b> | [52.235168690871, 21.071138918572952] | <b>Mokotów</b> | [52.18866196405862, 21.052814696946914] | <b>Wola</b> | [52.22969762335711, 20.94634273758449] | <b>Wilanów</b> | [52.15030830502675, 21.091139416474352] |
|--|--|-----------------|--|-----------------|--|-----------------------|---------------------------------------|----------------|---|-------------|--|----------------|---|
| District_center                                  |  |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| <b>Żoliborz</b>                                  | [52.2688536216614, 20.985135391429377]   |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| <b>Praga-Południe</b>                            | [52.235168690871, 21.071138918572952]  |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| <b>Mokotów</b>                                   | [52.18866196405862, 21.052814696946914]  |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| <b>Wola</b>                                      | [52.22969762335711, 20.94634273758449]   |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| <b>Wilanów</b>                                   | [52.15030830502675, 21.091139416474352]  |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| Dictionary with districts polygons:              | <pre>{'Żoliborz': [[20.957550244360345, 52.266927972075955], [20.957595033280743, 52.26712068328835], [20.957577637116344, 52.26723607840504], [20.957537312295642, 52.26727740707875], [20.957761257322645, 52.267475395339645], [20.957815144778245, 52.267498512850246],</pre>  |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |
| Districts radiuses to be used in Foursquare api: | <pre>{'Żoliborz': 2735, 'Praga-Południe': 4228, 'Mokotów': 4866, 'Wola': 3791, 'Wilanów': 5010, 'Wesoła': 3668, 'Wawer': 7006, 'Włochy': 4599, 'Ursynów': 4896, 'Śródmieście': 4339, 'Praga-Północ': 3133, 'Ursus': 2402, 'Targówek': 3617, 'Rembertów': 3461, 'Ochota': 2640, 'Bielany': 4385, 'Białołęka': 7513, 'Bemowo': 3478}</pre>   |                 |  |                 |  |                       |                                       |                |   |             |  |                |   |

With help of these informations, I prepared function for sending request to Foursquare API. Returned informations were used for creation of Dataframes for both cities with all venues in all districts:

|   | District | Name                           | Category            | Lat       | Lon       | Venueld                  | Inside |
|---|----------|--------------------------------|---------------------|-----------|-----------|--------------------------|--------|
| 0 | Żoliborz | Park Żeromskiego               | Park                | 52.268377 | 20.988747 | 4baf7aa5f964a52031033ce3 | True   |
| 1 | Żoliborz | Galeria Wypieków               | Bakery              | 52.268523 | 20.986111 | 55508b67498e2dcf9038f190 | True   |
| 2 | Żoliborz | Plac zabaw w Parku Żeromskiego | Playground          | 52.267248 | 20.988827 | 4db2f35a4b226b343d6d0581 | True   |
| 3 | Żoliborz | Kino Wisła                     | Indie Movie Theater | 52.269609 | 20.986743 | 4c14d3afa9c220a11e18589d | True   |
| 4 | Żoliborz | Plac Wilsona                   | Plaza               | 52.268914 | 20.985587 | 4bb771276edc76b0a92e321c | True   |

...

Then, I grouped venues by category and prepared top 5 categories for Warsaw and Krakow:

warsaw\_top5

|                    | District | Name | Lat | Lon | Venueld |
|--------------------|----------|------|-----|-----|---------|
| Category           |          |      |     |     |         |
| Café               | 179      | 179  | 179 | 179 | 179     |
| Park               | 140      | 140  | 140 | 140 | 140     |
| Coffee Shop        | 138      | 138  | 138 | 138 | 138     |
| Italian Restaurant | 117      | 117  | 117 | 117 | 117     |
| Supermarket        | 84       | 84   | 84  | 84  | 84      |

cracow\_top5

|                    | District | Name | Lat | Lon | Venueld | Inside |
|--------------------|----------|------|-----|-----|---------|--------|
| Category           |          |      |     |     |         |        |
| Hotel              | 52       | 52   | 52  | 52  | 52      | 52     |
| Supermarket        | 35       | 35   | 35  | 35  | 35      | 35     |
| Italian Restaurant | 31       | 31   | 31  | 31  | 31      | 31     |
| Café               | 26       | 26   | 26  | 26  | 26      | 26     |
| Park               | 24       | 24   | 24  | 24  | 24      | 24     |

Then I summed up top categories, to finally receive top 5 categories for further analysis:

['Café', 'Coffee Shop', 'Hotel', 'Italian Restaurant', 'Park']

Last step before creation of choroplets were counting venues of given categories in districts, and calculating densities [number of venues/km2]:

|           | Name               | Lat | Lon | Venueld |
|-----------|--------------------|-----|-----|---------|
| District  | Category           |     |     |         |
| Bemowo    | Café               | 5   | 5   | 5       |
|           | Coffee Shop        | 3   | 3   | 3       |
|           | Italian Restaurant | 4   | 4   | 4       |
|           | Park               | 8   | 8   | 8       |
| Białoleka | Café               | 8   | 8   | 8       |

|   | Category           | District  | Category_density |
|---|--------------------|-----------|------------------|
| 0 | Café               | Bemowo    | 0.200401         |
| 1 | Coffee Shop        | Bemowo    | 0.120240         |
| 2 | Italian Restaurant | Bemowo    | 0.160321         |
| 3 | Park               | Bemowo    | 0.320641         |
| 4 | Café               | Białoleka | 0.109529         |
| 5 | Coffee Shop        | Białoleka | 0.068456         |
| 6 | Hotel              | Białoleka | 0.027382         |
| 7 | Italian Restaurant | Białoleka | 0.027382         |
| 8 | Park               | Białoleka | 0.136911         |
| 9 | Café               | Bielany   | 0.278293         |

Having above data prepared, creating choropleth maps with Folium is straightforward. Results are available in results section.

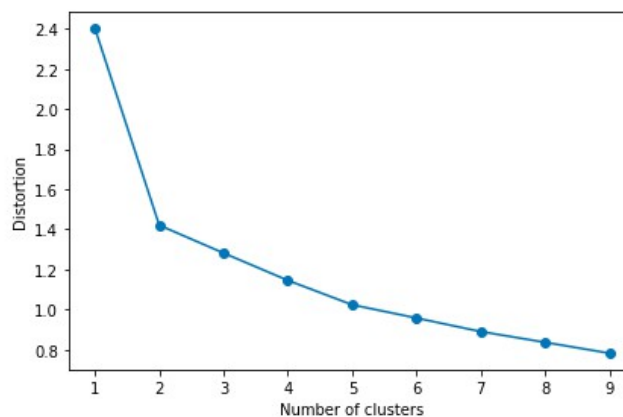
## 2.2. Clustering districts.

For further analysis, both cities venues dataframes were concatenated into one dataframe. Then I found top 10 categories for each district:

|   | District          | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| 0 | Bemowo            | Supermarket           | Park                  | Italian Restaurant    | Playground            | Coffee Shop           | Café                  | Shopping Mall         | Grocery Store         | Ski Area              | Mobile Phone Shop      |
| 1 | Białoleka         | Supermarket           | Shopping Mall         | Hotel                 | Sporting Goods Shop   | Bar                   | Multiplex             | Neighborhood          | Fast Food Restaurant  | Farmers Market        | Tennis Court           |
| 2 | Bielany           | Café                  | Grocery Store         | Gym / Fitness Center  | Park                  | Coffee Shop           | Supermarket           | Gym                   | Pizza Place           | Indian Restaurant     | Bookstore              |
| 3 | Bieńczyce         | Burger Joint          | Lake                  | Soccer Field          | Supermarket           | Food & Drink Shop     | Shopping Mall         | Fast Food Restaurant  | Farmers Market        | Market                | Park                   |
| 4 | Bieżanów-Prokocim | Bus Station           | Pizza Place           | Bakery                | Field                 | Supermarket           | Shoe Store            | Go Kart Track         | Park                  | Gym Pool              | Tram Station           |

...

Then k-means algorithm was used to group districts into clusters. I used so called 'elbow method' to choose optimal number of clusters:



In this plot, elbow is located at k=2. Nevertheless, when algorithm was executed with k=2, almost all districts were classified to one cluster:

```
: districts_venues_sorted.groupby('Cluster Labels').count()
```

```
:
```

|                | District | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|----------------|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| Cluster Labels |          |                       |                       |                       |                       |                       |                       |                       |                       |                       |                        |
| 0              | 35       | 35                    | 35                    | 35                    | 35                    | 35                    | 35                    | 35                    | 35                    | 35                    | 35                     |
| 1              | 1        | 1                     | 1                     | 1                     | 1                     | 1                     | 1                     | 1                     | 1                     | 1                     | 1                      |

I tested algorithm with different k, and most reasonable results were obtained for k=3:

```
districts_venues_sorted.groupby('Cluster Labels').count()
```

|                   | District | 1st Most<br>Common<br>Venue | 2nd Most<br>Common<br>Venue | 3rd Most<br>Common<br>Venue | 4th Most<br>Common<br>Venue | 5th Most<br>Common<br>Venue | 6th Most<br>Common<br>Venue | 7th Most<br>Common<br>Venue | 8th Most<br>Common<br>Venue | 9th Most<br>Common<br>Venue | 10th Most<br>Common<br>Venue |
|-------------------|----------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|------------------------------|
| Cluster<br>Labels |          |                             |                             |                             |                             |                             |                             |                             |                             |                             |                              |
| 0                 | 1        | 1                           | 1                           | 1                           | 1                           | 1                           | 1                           | 1                           | 1                           | 1                           | 1                            |
| 1                 | 11       | 11                          | 11                          | 11                          | 11                          | 11                          | 11                          | 11                          | 11                          | 11                          | 11                           |
| 2                 | 24       | 24                          | 24                          | 24                          | 24                          | 24                          | 24                          | 24                          | 24                          | 24                          | 24                           |

Only one district was labeled as cluster 0, so I will not perform its further analysis. For clusters 1 and 2, from top 10 types of venues in each district, I listed most popular. Below are most popular types of venues in clusters 1 and 2 and number of its appearances in top 10:

| Cluster 1            |    | Cluster 2          |    |
|----------------------|----|--------------------|----|
| Supermarket          | 11 | Park               | 16 |
| Fast Food Restaurant | 7  | Italian Restaurant | 15 |
| Park                 | 6  | Café               | 13 |
| Shopping Mall        | 6  | Supermarket        | 11 |
| Pizza Place          | 5  | Pizza Place        | 10 |

## 2.3. Correlations between different types of venues.

My last aim was investigation of correlations between different types of venues. This can be easily done by creating correlation matrix -in Pandas it can be done by calling `corr()` method on dataframe. As a result, desired matrix is obtained. It can be visualized -for example by so called heatmap -available in results section.

Although heatmaps can be great visualization tool, in this specific case it is not really useful -correlation matrix is too big (235x235), and it is almost impossible to find most correlated types of venues. Nevertheless, this matrix can be still useful for finding correlations. For example, here are places with highest correlation coefficient with Cafe:

|                               |          |
|-------------------------------|----------|
| Dessert Shop                  | 0.546974 |
| Indian Restaurant             | 0.452301 |
| Beer Bar                      | 0.414315 |
| Ramen Restaurant              | 0.409367 |
| Beach                         | 0.404221 |
| Vegetarian / Vegan Restaurant | 0.392186 |
| Food & Drink Shop             | 0.390660 |
| Liquor Store                  | 0.373259 |
| Wine Bar                      | 0.357540 |



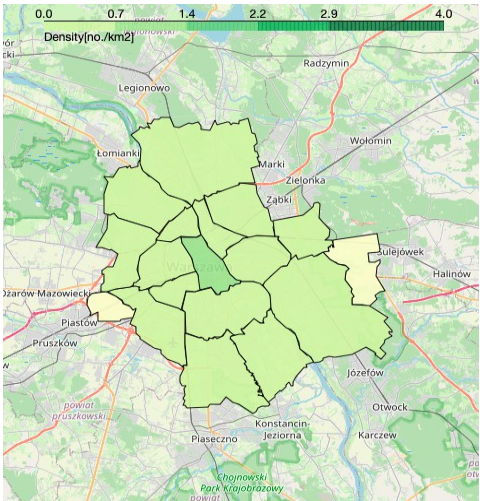
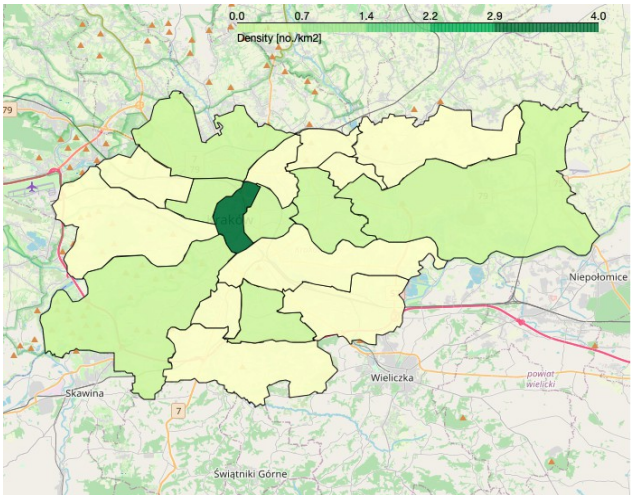
3. Results.

3.1. Choroplet maps.

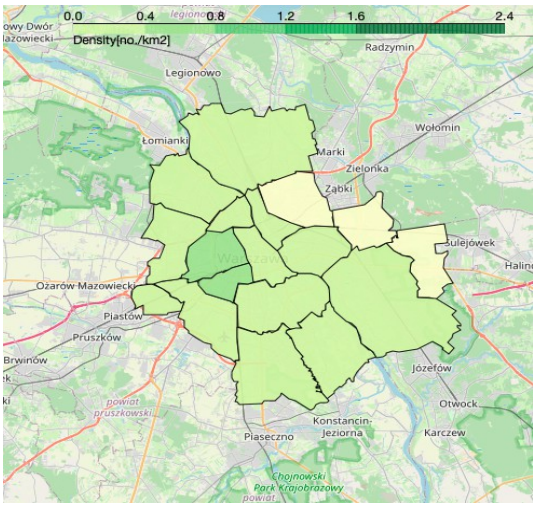
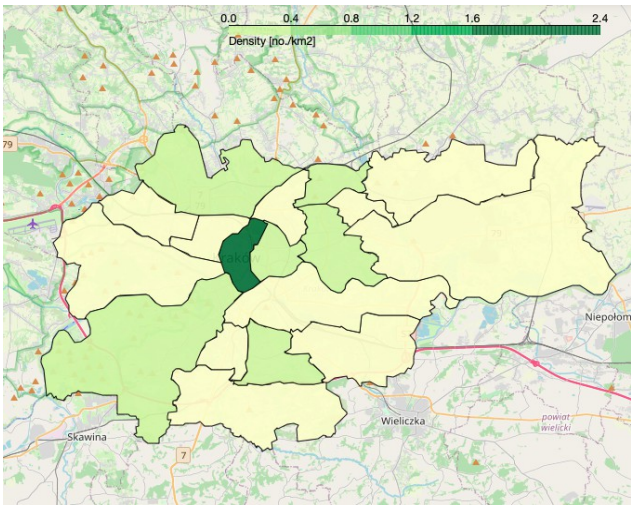
Krakov

Warsaw

Cafe

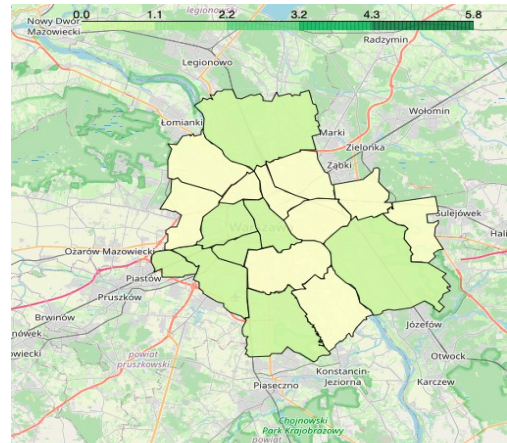
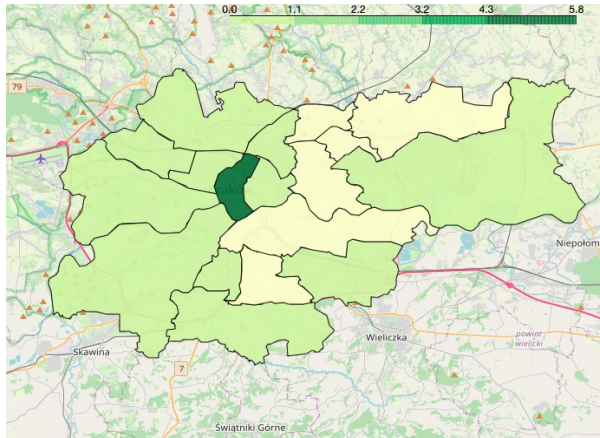


Coffee shops

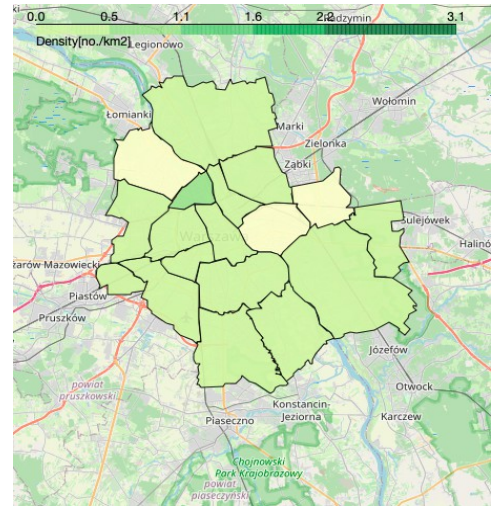
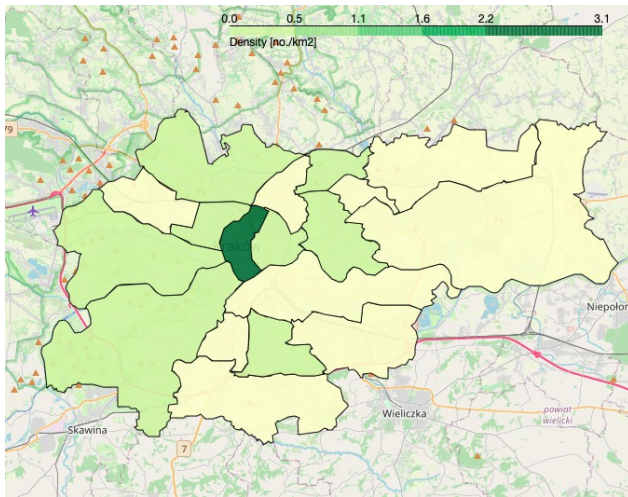




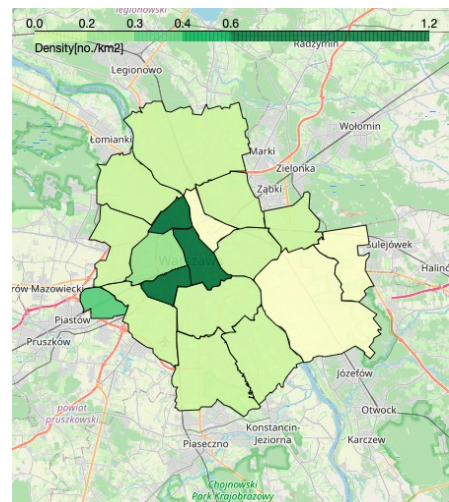
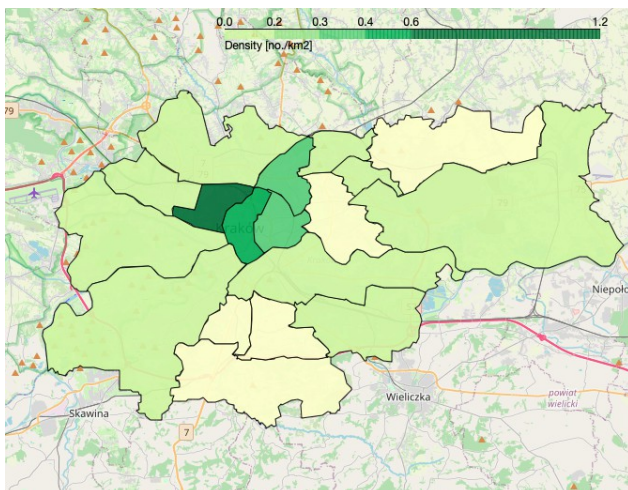
## Hotels



## Italian restaurants

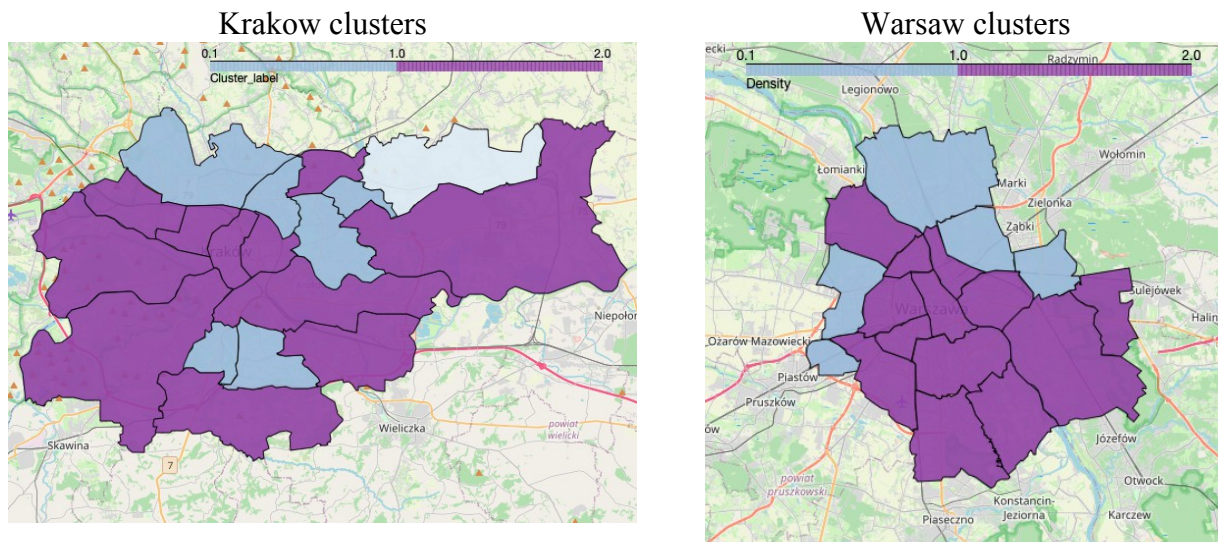


## Parks



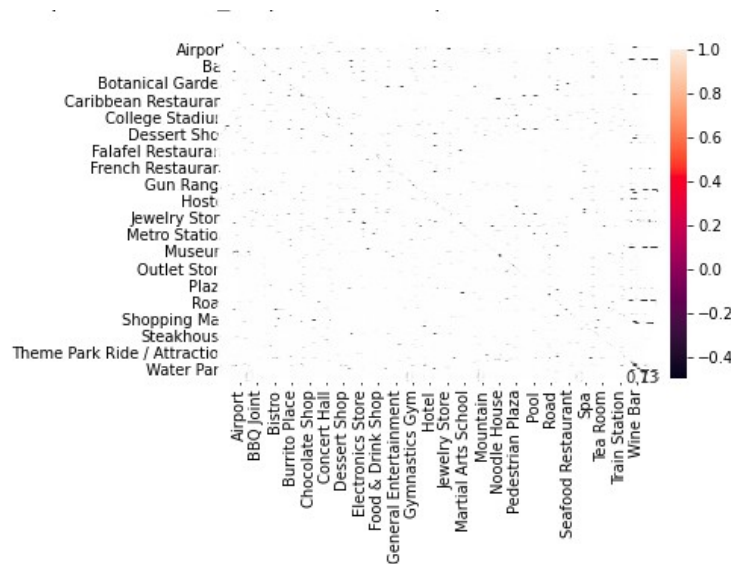
### 3.2. Clustering.

Here are results of performed clustering -maps with labels:



### 3.3. Correlations.

Heatmap of correlations matrix:



## 4. Discussion.

Finally, all my goals were achieved. Below I present short discussion about result.

### 4.1. Choropleth maps.

Although received choropleth maps look attractive, they are not as informative as I expected. On most of them, there are only three groups of density values used. Possible reason of that

problem is dividing scale into equal intervals from zero to maximum. Usually only one district fall into maximum interval, and all others into first and second minimum interval. Possible remedies for that could be different division of density scale. Second possible reason is fact, that Krakow centre is much more popular among english tourist than other districts of Krakow and Warsaw. Because of that, there is much more venues added in this area in Foursquare app.

## **4.2. Clustering.**

So called elbow method was used to determine optimal number of clusters, but results were not satisfactory -almost all districts were cumulated in one cluster. The best results, in terms of distribution of districts in cluster, were achieved for  $k=3$ . Nevertheless, clusters do not seems to be internally coherent. This could be caused by small number of venues received from Foursquare. To verify this hypothesis, some other service could be used to compare numbers of venues.

## **4.3. Correlations.**

Correlation matrix can be really useful for finding what kinds of places appear simultaneously in given area. This may be a hint of where to locate own business. But again, because of small number of venues in many districts, results should be compared with other service of similar functionality.

## **5. Conclusions**

Finally, all my goals were achieved. Nevertheless, comparing two cities is harder than I expected -completely different districts sizes and different amount of data available for Warsaw and Krakow made it challenging. What I miss in my report, is comparison with some other source of data (eg. google maps). What was most surprising for me, is amount of venues available in Foursquare for Krakow centre is so far away from other districts.