Capstone Project

"The Battle of Athens's Neighborhoods, Greece"

Aman Kumar

12/24/2018

TABLE OF CONTENTS

INTRODUCTION/BUSINESS PROBLEM

A Pharmacist is looking to set up new “Pharmacy” store in a neighborhood of Athens, Greece.

He needs his pharmacy’s location to be near his home location, so specific municipalities of Athens (24 different municipalities of Athens) are investigated.

To be sure that his new pharmacy store will have good profit, he requires to:

- set up his business to a municipality that live lot of people (so, we examine each municipality population)
- to be as near as possible to a Hospital
- to be as far as possible to other competitors’ Pharmacy stores

This report is focused to propose the better location for Pharmacist to set up a new “Pharmacy” store, according to above requirements.

DATA SECTION

Below data are used to examine this case:

First of all, I need a list of Athens (Attica, Greece) Municipalities with their population figure. According to "General Secretariat of National Statistical Service of Greece" there is a table that shows population (latest figures from year 2011) per Municipality. So, I use below page to extract required information:

http://www.citypopulation.de/php/greece-attiki.php

GREECE: Attica

Municipalities

The resident population of the municipalities in the Attica Region (Attikí) according to census results.

Name	Native	Status	Population Census 1991-03-17	Population Census 2001-03-18	Population Census 2011-03-16
Dímos Acharnón [Acharnae]	Δήμος Αχαρνών	Municipality	65,035	82,555	106,943
Dímos Agías Paraskevís [Agia Paraskevi]	Δήμος Αγίας Παρασκευής	Municipality	48,557	60,065	59,704
Dímos Agías Varváras [Agia Varvara]	Δήμος Αγίας Βαρβάρας	Municipality	29,426	31,354	26,550
Dímos Agíon Anargýron - Kamateroú [Agii Anargyri]	Δήμος Αγίων Αναργύρων - Καματερού	Municipality	49,577	58,244	62,529
Dímos Agíou Dimitríou [Agios Dimitrios]	Δήμος Αγίου Δημητρίου	Municipality	59,662	68,719	71,294
Dímos Agkistríou [Angistri, Agistri]	Δήμος Αγκιστρίου	Municipality	755	886	1,142
Dímos Aigáleo [Egaleo]	Δήμος Αιγάλεω	Municipality	81,607	77,917	69,946
Dímos Aíginas [Aegina]	Δήμος Αίγινας	Municipality	11,103	12,716	13,056
Dímos Alímou [Alimos]	Δήμος Αλίμου	Municipality	32,514	39,800	41,720
Dímos Amarousíou [Marousi]	Δήμος Αμαρουσίου	Municipality	64,083	71,551	72,333
Dímos Aspropýrgou [Aspropyrgos]	Δήμος Ασπροπύργου	Municipality	15,674	27,927	30,251
Dímos Athinaíon [Athens]	Δήμος Αθηναίων	Municipality	816,556	789,166	664,046
Dímos Chaïdaríou [Chaidari]	Δήμος Χαϊδαρίου	Municipality	48,608	48,494	46,897

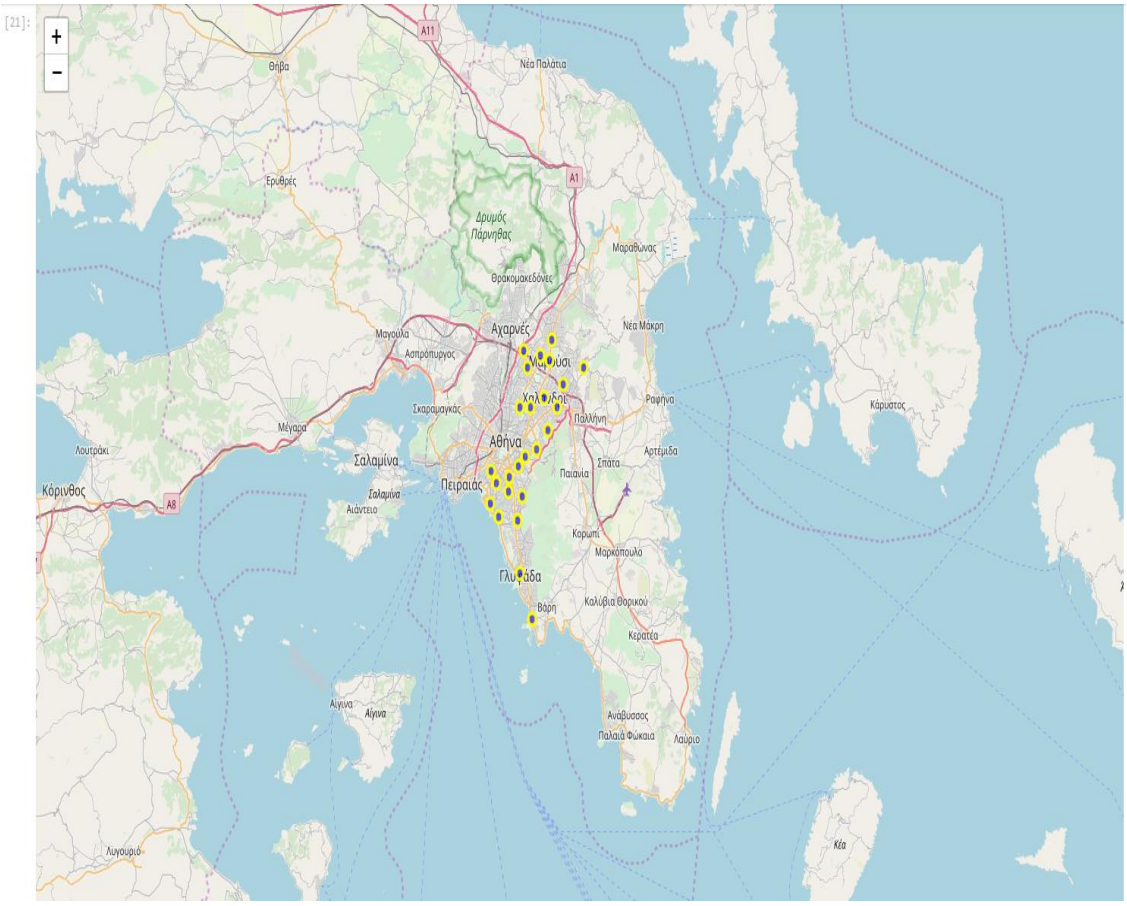
As we can see, this table has 6 columns:

- first column "Name", list all names of Athens (Attica) Municipalities
- second column "Native" has its municipality name in Greek
- third column "Status" states that each name corresponds to Municipality
- fourth column shows each municipality Population figure from year 1991
- fifth column shows each municipality Population figure from year 2001
- sixth column shows each municipality Population figure from year 2011

From above table, first and last column have the information we need.

Then, to continue my investigation I ask my friend Pharmacist to provide me with name of municipalities that is interested to set up his business. He is only

interesting to specific 26 municipalities: Agia Paraskevi, Agios Dimitrios, Alimos, Marousi, Chalandri, Dafni-Ymittos, Elliniko-Argyroupoli, Filothei-Psychiko, Galatsi, Glyfada, Ilioupoli, Irakleio, Kessariani, Kallithea, Kifisia, Lykovrysi-Pefki, Metamorfosi, Nea Ionia, Nea Smyrni, Palaio Faliro, Papagou-Cholargos, Penteliri-Voula-Vouliagmeni, Vrilissia, Vyronas, Zografos.



Then, I prepared a csv file that lists all above municipalities and their coordination: (https://github.com/abarou12/Coursera_Capstone/blob/master/coordinates.csv)

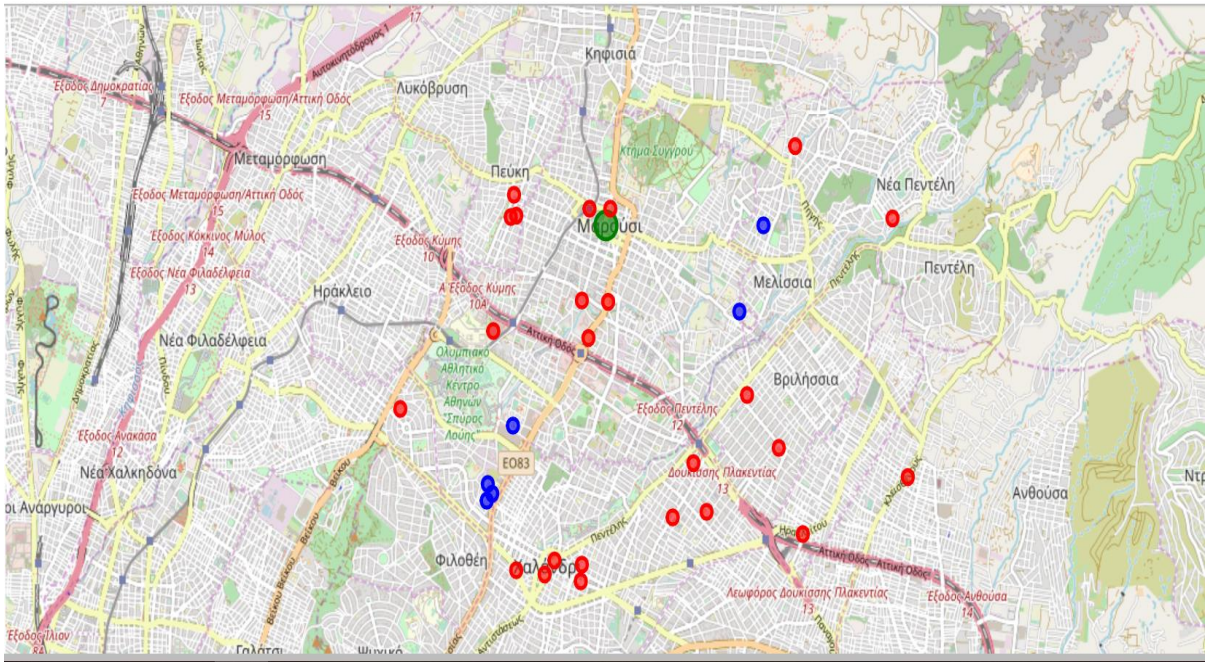
	A	B	C	D
1	**Municipality**	**Latitude**	**Longitude**	
2	Agia Paraskevi	38.012630	23.820550	
3	Agios Dimitrios	37.936670	23.733200	
4	Alimos	37.913680	23.715060	
5	Marousi	38.054570	23.807500	
6	Chalandri	38.021530	23.798270	
7	Dafni-Ymittos	37.950140	23.734320	
8	Elliniko-Argyroupoli	37.911070	23.749140	
9	Filothei-Psychiko	38.012420	23.772500	
10	Galatsi	38.012620	23.753180	
11	Glyfada	37.862820	23.753440	
12	Ilioupoli	37.931910	23.757250	
13	Irakleio	38.048110	23.767800	
14	Kessariani	37.968318	23.763790	
15	Kallithea	37.955521	23.702030	
16	Kifisia	38.072810	23.811930	
17	Lykovrysi-Pefki	38.058970	23.791150	
18	Metamorfosi	38.063229	23.760931	
19	Nea Ionia	39.3729263	22.9340436	
20	Nea Smyrni	37.944288	23.711756	
21	Palaio Faliro	37.925369	23.700654	
22	Papagou-Cholargos	37.992266	23.804931	
23	Penteli	38.04763	23.869628	
24	Vari-Voula-Vouliagmeni	37.821658	23.775413	
25	Vrilissia	38.032979	23.831904	
26	Vyronas	37.959333	23.751301	
27	Zografos	37.974215	23.784107	
28				
29				
30				

To continue, I use Foursquare.com (https://foursquare.com/) to fetch

- all other competitor's Pharmacy with their coordinates and distance from center of each Municipality
- all Hospitals for each interested Municipality.

Using Folium (https://pypi.org/project/folium/), I create relative maps to view interested results: e.g. for municipality named "Marousi", a map is created that

shows competitors' Pharmacy (red cycles) and Medical Centers/Hospitals (blue cycles)



With above data, I will examine and compare all 26 municipalities and using K-means algorithm, I propose the best location for new "Pharmacy" to be set up.

METHODOLOGY SECTION

First, from internet page “City Population” I extract -using Beautiful soup - only Municipalities table data, and create a List with table’s data.

```
[2]: myList=list()
res = requests.get("http://www.citypopulation.de/php/greece-attiki.php")
soup = BeautifulSoup(res.text, 'lxml')
table = soup.find("table", class_="data")

for items in table.find_all("tr")[:-1]:
    data = [' '.join(item.text.split()) for item in items.find_all(['th','td'])]
    myList.append(data)
print(myList)
```

```
[['Name', 'Native', 'Status', 'PopulationCensus1991-03-17', 'PopulationCensus2001-03-18', 'PopulationCensus2011-03-16', ''], ['Dímos Acharnón [Acharnae]',
'Δήμος Αχαρνών', 'Municipality', '65,035', '82,555', '106,943', '→'], ['Dímos Agías Paraskevís [Agia Paraskevi]', 'Δήμος Αγίας Παρασκευής', 'Municipality',
'48,557', '60,065', '59,704', '→'], ['Dímos Agías Varváras [Agia Varvara]', 'Δήμος Αγίας Βαρβάρας', 'Municipality', '29,426', '31,354', '26,550', '→'], ['D
ímos Agíon Anargýron - Kamateroú [Agii Anargyri]', 'Δήμος Αγίων Αναργύρων - Καματερού', 'Municipality', '49,577', '58,244', '62,529', '→'], ['Dímos Agíou D
imitríou [Agios Dimitrios]', 'Δήμος Αγίου Δημητρίου', 'Municipality', '59,662', '68,719', '71,294', '→'], ['Dímos Agkistríou [Angistri, Agistri]', 'Δήμος A
γκιστρίου', 'Municipality', '755', '886', '1,142', '→'], ['Dímos Aigáleo [Egaleo]', 'Δήμος Αιγάλεω', 'Municipality', '81,607', '77,917', '69,946', '→'],
['Dímos Aíginas [Aegina]', 'Δήμος Αίγινας', 'Municipality', '11,103', '12,716', '13,056', '→'], ['Dímos Alímou [Alimos]', 'Δήμος Αλίμου', 'Municipality',
'32,514', '39,800', '41,720', '→'], ['Dímos Amarousíou [Marousi]', 'Δήμος Αμαρουσίου', 'Municipality', '64,083', '71,551', '72,333', '→'], ['Dímos Aspropýr
gou [Aspropyrgos]', 'Δήμος Ασπροπύργου', 'Municipality', '15,674', '27,927', '30,251', '→'], ['Dímos Athínaíon [Athens]', 'Δήμος Αθηναίων', 'Municipality',
'816,556', '789,166', '664,046', '→'], ['Dímos Chaïdaríou [Chaidari]', 'Δήμος Χαϊδαρίου', 'Municipality', '48,608', '48,494', '46,897', '→'], ['Dímos Chala
ndríou [Chalandri]', 'Δήμος Χαλανδρίου', 'Municipality', '67,724', '75,327', '74,192', '→'], ['Dímos Dáfnis - Ymittoú [Dafni-Ymittos]', 'Δήμος Δάφνης - Υμη
ττού', 'Municipality', '37,003', '36,804', '33,628', '→'], ['Dímos Dionýsou [Dionysos]', 'Δήμος Διονύσου', 'Municipality', '17,249', '32,504', '40,193',
'→'], ['Dímos Elefsínas [Eleusina]', 'Δήμος Ελευσίνας', 'Municipality', '25,992', '29,879', '29,902', '→'], ['Dímos Ellinikoú - Argyroúpolis [Elliniko-Argy
roupoli]', 'Δήμος Ελληνικού - Αργυρούπολης', 'Municipality', '44,802', '51,299', '51,356', '→'], ['Dímos Filadelfeías - Chalkidónos [Filadelfeia-Chalkidon
a]', 'Δήμος Φιλαδελφείας - Χαλκηδόνος', 'Municipality', '36,422', '35,607', '35,556', '→'], ['Dímos Filothéis - Psychikoú [Filothei-Psychiko]', 'Δήμος Φιλο
θέης - Ψυχικού', 'Municipality', '31,888', '30,754', '26,968', '→'], ['Dímos Fylís [Fyli]', 'Δήμος Φυλής', 'Municipality', '33,731', '39,137', '45,965',
'→'], ['Dímos Galatsíou [Galatsi]', 'Δήμος Γαλατσίου', 'Municipality', '59,533', '63,418', '59,345', '→'], ['Dímos Glyfádas [Glyfada]', 'Δήμος Γλυφάδας',
```

Then I transform above List to Dataframe, rename columns and reset index:

```
[98]: data_transposed = zip(*myList)
df = pd.DataFrame(myList, columns=['Municipality', 'Native', 'Status', 'Population1991', 'Population2001', 'Population2011', 'tot'])
df=df.tail(-1)
df = df.reset_index(drop=True)
df.head()
```

[98]:

	Municipality	Native	Status	Population1991	Population2001	Population2011	tot
1	Dímos Acharnón [Acharnae]	Δήμος Αχαρνών	Municipality	65,035	82,555	106,943	→
2	Dímos Agías Paraskevís [Agia Paraskevi]	Δήμος Αγίας Παρασκευής	Municipality	48,557	60,065	59,704	→
3	Dímos Agías Varváras [Agia Varvara]	Δήμος Αγίας Βαρβάρας	Municipality	29,426	31,354	26,550	→
4	Dímos Agíon Anargýron - Kamateroú [Agii Anargyri]	Δήμος Αγίων Αναργύρων - Καματερού	Municipality	49,577	58,244	62,529	→
5	Dímos Agíou Dimitríou [Agios Dimitrios]	Δήμος Αγίου Δημητρίου	Municipality	59,662	68,719	71,294	→

```
[102]: df.shape
```

```
[102]: (65, 7)
```

From dataframe shape, we can see that there are 65 different Municipalities in Athens, Attica area.

Then I drop some columns that are not needed, and I keep Municipalities name, the one that is inside brackets:

```
[103]: df.drop(["Native", "Status", "Population1991", "Population2001", "tot"], axis=1, inplace=True)
```

```
[104]:
```

[104]:

	Municipality	Population2011
0	Dímos Agías Paraskevís [Agia Paraskevi]	59,704
1	Dímos Agías Varváras [Agia Varvara]	26,550
2	Dímos Agíon Anargýron - Kamateroú [Agii Anargyri]	62,529
3	Dímos Agíou Dimitríou [Agios Dimitrios]	71,294
4	Dímos Agkistríou [Angistri, Agistri]	1,142
5	Dímos Aigáleo [Egaleo]	69,946
6	Dímos Aíginas [Aegina]	13,056
7	Dímos Alímou [Alimos]	41,720
8	Dímos Amarousíou [Marousi]	72,333

From "Municipality" columns, we keep only names inside brackets

[105]:
```
df['Municipality']=df['Municipality'].apply(lambda st: st[st.find("[")+1:st.find("]")])
df.head(5)
```

[105]:

	Municipality	Population2011
0	Agia Paraskevi	59,704
1	Agia Varvara	26,550
2	Agii Anargyri	62,529
3	Agios Dimitrios	71,294
4	Angistri, Agistri	1,142

Then, we read csv file to dataframe:

[106]:
```
csv=pd.read_csv("coordinates.csv", delimiter=';',
                header = 0)
csv
```

[106]:

	Municipality	Latitude	Longitude
0	Agia Paraskevi	38.012630	23.820550
1	Agios Dimitrios	37.936670	23.733200
2	Alimos	37.913680	23.715060
3	Marousi	38.054570	23.807500
4	Chalandri	38.021530	23.798270
5	Dafni-Ymittos	37.950140	23.734320
6	Elliniko-Argyroupoli	37.911070	23.749140
7	Filothei-Psychiko	38.012420	23.772500
8	Galatsi	38.012620	23.753180
9	Glyfada	37.862820	23.753440
10	Ilioupoli	37.931910	23.757250
11	Irakleio	38.048110	23.767800

To continue, I merge both dataframes and convert Population values to type float, so to be able to make calculation with these data:

```
[108]: result = pd.merge(csv, df, how='inner', on=['Municipality', 'Municipality'])
```

```
[187]: result.head(2)
```

[187]:

	Municipality	Latitude	Longitude	Population2011
0	Agia Paraskevi	38.01263	23.82055	59.704
1	Agios Dimitrios	37.93667	23.73320	71.294

```
[110]: print (result.dtypes)
```

```
Municipality       object
Latitude          float64
Longitude         float64
Population2011     object
dtype: object
```

```
[111]: result['Population2011'] = result.Population2011.str.replace(',', '.').astype(float)
```

```
[112]: print (result.dtypes)
```

```
Municipality       object
Latitude          float64
Longitude         float64
Population2011    float64
dtype: object
```

Next, we fetch via Foursquare all Hospitals that are in these 65 Municipalities:

```
[127]: search_query = 'Hospital'
categoryId = '4bf58dd8d48988d104941735'
```

```
[128]: def getNearbyVenues(names, population, lat1, long1, radius=3000):

    venues_list=[]
    for name, pop, lat, lng in zip(names,population, lat1, long1):

        # create the API request URL
        url1 = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}&locale={}&categoryId={}

        # make the GET request
        results = requests.get(url1).json()["response"]["venues"]

        # return only relevant information for each nearby venue
        venues_list.append([(
            name, pop, lat, lng,
            v['name'],
            "Hospital",
            v['location']['lat'],
            v['location']['lng'] ) for v in results])

        nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])

    return(nearby_venues)
```

```
[129]: all_venues1 = getNearbyVenues(names=result['Municipality'],
                                    population=result["Population2011"],
                                    lat1=result['Latitude'],
                                    long1=result['Longitude']
                                   )
```

Rename dataframe headers:

```
[131]: all_venues1.rename(columns={0:'Municipality Name',1:"Population", 2:"Municipality Latitude", 3:"Municipality Longitude", 4:"Hospital Name", 5: "HCategory",
```

```
[132]: all_venues1.head()
```

[132]:

	Municipality Name	Population	Municipality Latitude	Municipality Longitude	Hospital Name	HCategory	Hospital Latitude	Hospital Longitude
0	Agia Paraskevi	59.704	38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.798580
1	Agia Paraskevi	59.704	38.01263	23.82055	Υγεία	Hospital	38.028521	23.789724
2	Agia Paraskevi	59.704	38.01263	23.82055	VVIP 700 suite MITERA HOSPITAL	Hospital	38.029407	23.788968
3	Agia Paraskevi	59.704	38.01263	23.82055	Pet Hospital Maragos Thomas	Hospital	38.027768	23.788915
4	Agios Dimitrios	71.294	37.93667	23.73320	ΡΕΑ Μαιευτική Γυναικολογική Κλινική	Hospital	37.940895	23.697430

So, we can see that dataframe “all_venues1” consists of 8 columns: Municipality name and its coordination, Municipality Population, and every Hospital that are belonging to this Municipality – Hospital name and its

coordination. This dataframe contains of above for every Hospital for every 65 Municipalites.

To continue, I do the same process, to fetch all near Pharmacies for every Hospital that exist in my dataframe:

```
[136]: search_query = 'Pharmacy'
       categoryId = '4bf58dd8d48988d10f951735'
```

```
[137]: def getNearbyVenues1(m_names, population, m_lat1, m_long1,h_names, h_cat, h_lat1, h_long1, radius=1500):

           venues1_list=[]
           for m_name, pop, m_lat, m_long,h_name, hcat, lat, lng in zip(m_names, population, m_lat1, m_long1,h_names,h_cat, h_lat1, h_long1):

               # create the API request URL
               url2 = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}&locale={}&categoryId={}


               # make the GET request
               results1 = requests.get(url2).json()["response"]["venues"]

               # return only relevant information for each nearby venue
               venues1_list.append([(
                   m_name, pop, m_lat, m_long, h_name,hcat, lat,  lng,
                   v['name'],
                   "Pharmacy",
                   v['location']['lat'],
                   v['location']['lng'],
                   v['location']['distance']) for v in results1])

               nearby_venues1 = pd.DataFrame([item for venue1_list in venues1_list for item in venue1_list])

           return(nearby_venues1)
```

```
[138]: all_pharmacies = getNearbyVenues1(m_names=all_hospital['Municipality Name'],
                                         population=all_hospital["Population"],
                                         m_lat1=all_hospital['Municipality Latitude'],
                                         m_long1=all_hospital['Municipality Longitude'],
                                         h_names=all_hospital["Hospital Name"],
                                         h_cat=all_hospital["HCategory"],
                                         h_lat1=all_hospital['Hospital Latitude'],
                                         h_long1=all_hospital['Hospital Longitude']
                                        )
```

```
[139]: all_pharmacies.shape

[139]: (275, 13)
```

```
[142]: all_pharmacies.head()
```

[142]:

Latitude	MunicipalityLongitude	HospitalName	HCategory	HospitalLatitude	HospitalLongitude	PharmacyName	PhCategory	PharmacyLatitude	PharmacyLongitude	PharmacyDistance
38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Sotiropoulou pharmacy (Φαρμακείο Σωτηροπούλου)	Pharmacy	38.021996	23.799477	1781
38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy	Pharmacy	37.994760	23.795063	1289
38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy Fantidou	Pharmacy	38.019905	23.803622	1609
38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Dr. Pharmacy	Pharmacy	38.020636	23.797969	1629
38.01263	23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Martsoukas Pharmacy	Pharmacy	38.021656	23.803836	1802

As, we can see, last dataframe column is “PharmacyDistance” which is the distance between Hospital and Pharmacy, as second Foursquare query, I use Hospital coordinates to fetch each Pharmacy near to it.

Next, I need to calculate how many different Hospitals exist per Municipality and do the same for Pharmacies (e.g. to also calculate how many pharmacies exist near each Hospital):

```
[147]: group_Hospitals['HospSum']=group_Hospitals.groupby(['MunicipalityName']).transform('count')
       group_Hospitals.head()
```

```
/home/jupyterlab/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

[147]:

	MunicipalityName	HospitalName	HospSum
0	Agia Paraskevi	ΙΑΣΩ General	4
6	Agia Paraskevi	Υγεία	4
13	Agia Paraskevi	VVIP 700 suite MITERA HOSPITAL	4
20	Agia Paraskevi	Pet Hospital Maragos Thomas	4
27	Agios Dimitrios	ΡΕΑ Μαιευτική Γυναικολογική Κλινική	3

```
[154]: group_pharmacies['PharmSum']=group_pharmacies.groupby(['HospitalName']).transform('count')
       group_pharmacies.head()
```

```
/home/jupyterlab/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

[154]:

	HospitalName	PharmacyName	PharmSum
0	ΙΑΣΩ General	Sotiropoulou pharmacy (Φαρμακείο Σωτηροπούλου)	6
1	ΙΑΣΩ General	Pharmacy	6
2	ΙΑΣΩ General	Pharmacy Fantidou	6
3	ΙΑΣΩ General	Dr. Pharmacy	6
4	ΙΑΣΩ General	Martsoukas Pharmacy	6

```
[155]: group_pharmacies.drop("PharmacyName", axis=1, inplace=True)
```

New columns “HospSum” is the sum of Hospitals per Municipality, and new column “PharmSum” is the sum of Pharmacies per Hospital.

Then, I merge above two dataframes to original one, to have concatenate these two new columns:

```
[158]: resultH_PH = pd.merge(resultH, group_pharmacies, how='inner', on=['HospitalName', 'HospitalName'])
       resultH_PH.head()
```

[158]:

ngitude	HospitalName	HCategory	HospitalLatitude	HospitalLongitude	PharmacyName	PhCategory	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	HospSum	PharmSum
23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Sotiropoulou pharmacy (Φαρμακείο Σωτηροπούλου)	Pharmacy	38.021996	23.799477	1781	4	6
23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy	Pharmacy	37.994760	23.795063	1289	4	6
23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy Fantidou	Pharmacy	38.019905	23.803622	1609	4	6
23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Dr. Pharmacy	Pharmacy	38.020636	23.797969	1629	4	6
23.82055	ΙΑΣΩ General	Hospital	38.006005	23.79858	Martsoukas Pharmacy	Pharmacy	38.021656	23.803836	1802	4	6

According to original requirements, my friend Pharmacist wants to open its new Pharmacy store to a Municipality that have Hospital and also good Population Density.

So, “Density%” per Municipality is defined as: (“Population”*100%)/(HospSum*PharmSum)

```
[159]: def f(x, y, z):
           return (x*100)/(y*z)

[160]: resultH_PH['Density'] = resultH_PH.apply(lambda x: f(x.Population, x.HospSum, x.PharmSum), axis=1)
       resultH_PH.head()
```

[160]:

spitalName	HCategory	HospitalLatitude	HospitalLongitude	PharmacyName	PhCategory	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	HospSum	PharmSum	Density
ΑΣΩ General	Hospital	38.006005	23.79858	Sotiropoulou pharmacy (Φαρμακείο Σωτηροπούλου)	Pharmacy	38.021996	23.799477	1781	4	6	248.766667
ΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy	Pharmacy	37.994760	23.795063	1289	4	6	248.766667
ΑΣΩ General	Hospital	38.006005	23.79858	Pharmacy Fantidou	Pharmacy	38.019905	23.803622	1609	4	6	248.766667
ΑΣΩ General	Hospital	38.006005	23.79858	Dr. Pharmacy	Pharmacy	38.020636	23.797969	1629	4	6	248.766667
ΑΣΩ General	Hospital	38.006005	23.79858	Martsoukas Pharmacy	Pharmacy	38.021656	23.803836	1802	4	6	248.766667

```
[161]: resultH_PH.shape

[161]: (275, 16)
```

Then, to clear the dataframe I drop some columns I do not need any more:

Let's drop some columns we do not need anymore:

```
[162]: fresultH_PH=resultH_PH.drop(["MunicipalityLatitude", "MunicipalityLongitude","Population","HCategory","PhCategory","HospSum","PharmSum"], axis=1)
       fresultH_PH.head()
```

[162]:

	MunicipalityName	HospitalName	HospitalLatitude	HospitalLongitude	PharmacyName	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	Density
0	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Sotiropoulou pharmacy (Φαρμακείο Σωτηροπούλου)	38.021996	23.799477	1781	248.766667
1	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Pharmacy	37.994760	23.795063	1289	248.766667
2	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Pharmacy Fantidou	38.019905	23.803622	1609	248.766667
3	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Dr. Pharmacy	38.020636	23.797969	1629	248.766667
4	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Martsoukas Pharmacy	38.021656	23.803836	1802	248.766667

```
[188]: fresultH_PH.shape

[188]: (275, 10)
```

Final dataframe “fresultH_PH” consists of 10 columns and 275 rows.

To be able to group Municipalities having similar Density value and Pharmacy distance and since my data are unsupervised, I use machine learning algorithm “k-means”:

```
[165]: kclusters = 8

       # run k-means clustering
       kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(fresultH_PH_clustering1)

       # check cluster labels generated for each row in the dataframe
       kmeans.labels_[0:6]

[165]: array([5, 6, 7, 7, 5, 5], dtype=int32)
```

Let's create a new dataframe that includes the cluster

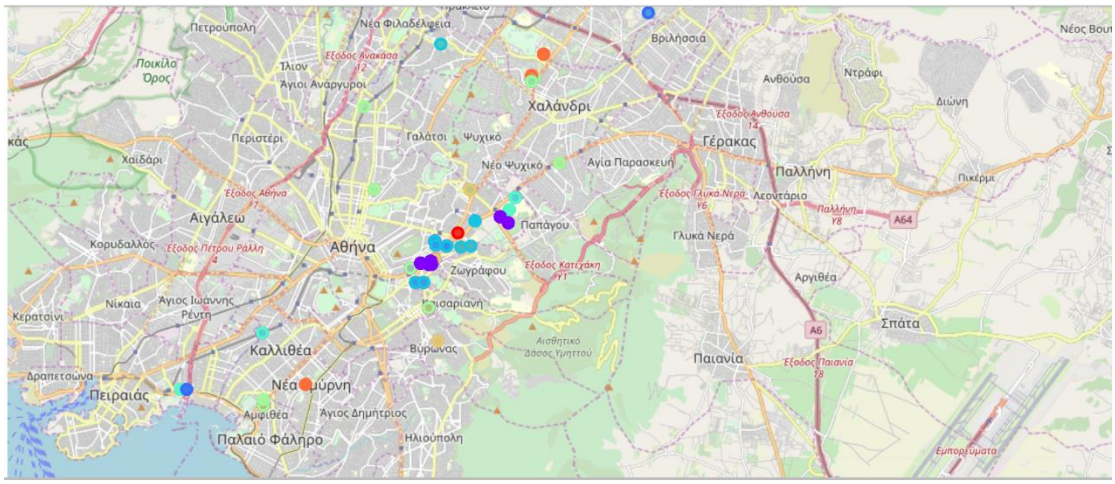
```
[166]: fresultH_PH_merged = fresultH_PH

       # add clustering labels
       fresultH_PH_merged['ClusterLabels'] = kmeans.labels_


       fresultH_PH_merged.head() # check the last columns!
```

[166]:

	MunicipalityName	HospitalName	HospitalLatitude	HospitalLongitude	PharmacyName	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	Density	ClusterLabels
0	Agia Paraskevi	ΙΑΣΩ General	38.006005	23.79858	Sotiropoulou pharmacy (Φαρμακείο	38.021996	23.799477	1781	248.766667	5

Now, another one column is added “ClusterLabels” in final dataframe. Each row belongs to a cluster with label “ClusterLabels”.

To visualize how cluster is in Athens, Attica area, we can see below map:


Πετρούπολη
Νέα Φιλαδέλφεια
Βριλήσσια
Ποικίλο Όρος
Ίλιον
Άγιοι Ανάργυροι
Εξόδος Αναξάρχου
Χαλάνδρι
Ανθούσα
Ντράφι
Διώνη
Περιστέρι
Γαλάτσι
Ψυχικό
Γέρακας
Χαϊδάρι
Νέο Ψυχικό
Αγία Παρασκευή
Παλλήνη
Πικέρμι
Εξόδος Αθήνα
Αιγάλεω
Παπάγου
Εξόδος Γλυκά Νερά
Λεωνταρίου
Παλλήνη
Κορυδαλλός
Εξόδος Πέτρου Ράλλη
Αθήνα
Γλυκά Νερά
Ζωγράφου
Εξόδος Κατεχάκη
Αργυθέα
Νίκαια
Άγιος Ιωάννης Ρέντης
Κερατσινί
Καλλιθέα
Βύρωνας
Αλσητικό Δάσος Υμηττού
Σπάτα
Παιανία
Εξόδος Παιανία
Δραπετσώνα
Πειραιάς
Νέα Σμύρνη
Άγιος Δημήτριος
Παλαιό Φάληρο
Ηλιούπολη
Εμπορεύματα

RESULTS SECTION

To better understand Cluster grouping and to be able to propose the best location according to original requirements open new Pharmacy store, Cluster visualization applied:

Let's visualize a plot with axes "PharmacyDistance" and "Density" and where each Cluster appears in the plot:

[176]:
```
import matplotlib.pyplot as plt
X = fresultH_PH_clustering1.values[0:,1:]
X = np.nan_to_num(X)
```

[177]:
```
X
```

[177]:
```
array([[  23.79857951,   38.02199624,   23.79947662, 1781.        ,
         248.76666667],
       [  23.79857951,   37.99476043,   23.79506251, 1289.        ,
         248.76666667],
       [  23.79857951,   38.019905  ,   23.803622  , 1609.        ,
         248.76666667],
       ...,
       [  23.69730893,   37.958954  ,   23.700054  ,  288.        ,
        1118.23333333],
       [  23.69730893,   37.9580409 ,   23.70378262,  624.        ,
        1118.23333333],
       [  23.69730893,   37.952438  ,   23.680405  , 1726.        ,
        1118.23333333]])
```

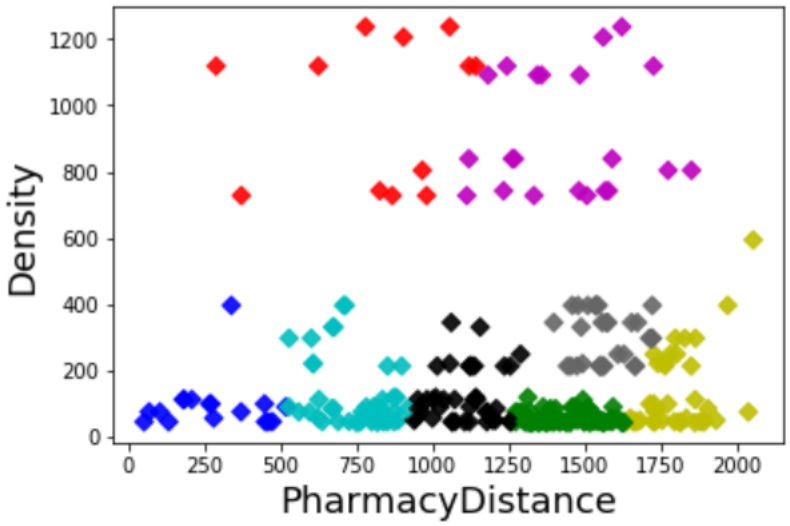
[178]:
```
fresultH_PH_clustering1.head(2)
```

[178]:

	HospitalLatitude	HospitalLongitude	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	Density
0	38.006005	23.79858	38.021996	23.799477	1781	248.766667

[179]:
```
LABEL_COLOR_MAP = {0 : 'b',    #cluster label: 0 - blue color
                   1 : 'g',    #cluster label: 1 - green color
                   2 : 'r',    #cluster label: 2 - red color
                   3 : 'c',    #cluster label: 3 - cyan color
                   4 : 'm',    #cluster label: 4 - magenta color
                   5 : 'y',    #cluster label: 5 - yellow color
                   6 : 'k',    #cluster label: 6 - black color
                   7 : '0.4',  #cluster label: 7 - grey color
                   }

label_color = [LABEL_COLOR_MAP[l] for l in kmeans.labels_]
```

[180]:
```
area =0.009* np.pi * ( X[:, 1])**2
plt.scatter(X[:, 3], X[:, 4], s=area, marker='D', c=label_color, alpha=0.9)
plt.xlabel('PharmacyDistance', fontsize=18)
plt.ylabel('Density', fontsize=18)

plt.show()
```



We can clearly see that clusters “cluster 4: Magenta” and “cluster 2: Red” contain Hospital having far other Pharmacies competitors and belonging to Municipalities with big Density.

We confirm above finding by sorting the final dataframe first by descending “Density” and then by ascending “PharmacyDistance”

```
[182]: fresultH_PH_merged.sort_values(['Density', 'PharmacyDistance'], ascending=[False, True], inplace=True)
       fresultH_PH_merged.head(10)
```

[182]:

	MunicipalityName	HospitalName	HospitalLatitude	HospitalLongitude	PharmacyName	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	Density	ClusterLabels
59	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	Pharmacy Angeliki Dragogianni (Φαρμακείο Αγγελ...	37.998847	23.774811	779	1236.533333	2
60	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	Pharmacy	37.994760	23.795063	1056	1236.533333	2
61	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	pharmacy ananiadis	37.996741	23.764802	1622	1236.533333	4
55	Marousi	Σισμανόγλειο Γενικό Νοσοκομείο Αττικής	38.046173	23.828541	KALTSOUNIS PHARMACY	38.038098	23.829667	904	1205.550000	2
		Σισμανόγλειο								

DISCUSSION SECTION

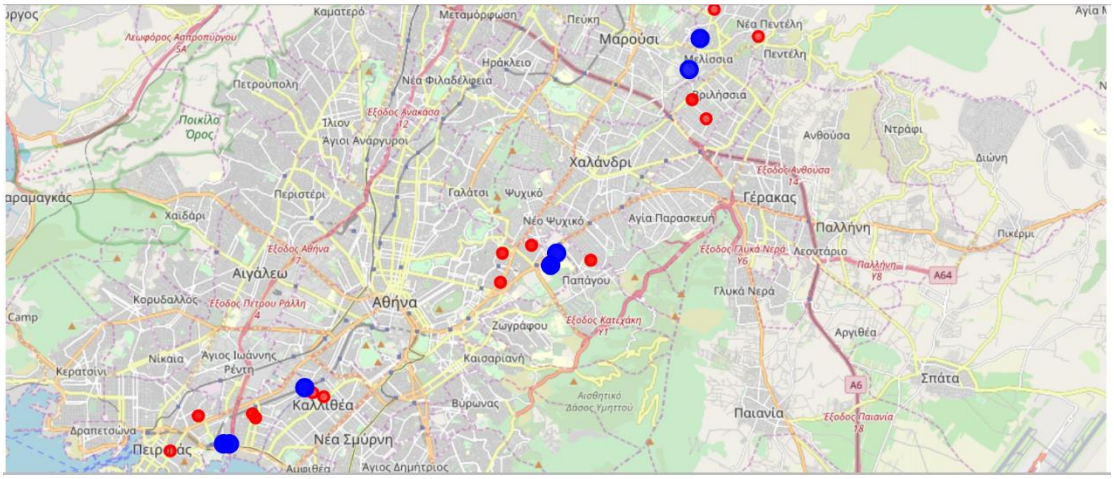
To able to propose more locations as alternatives, my friend Pharmacist can open his new Pharmacy Store near any hospital belonging in clusters 2 or 4:

```
[183]: d35 = fresultH_PH_merged[(fresultH_PH_merged.ClusterLabels==2) | (fresultH_PH_merged.ClusterLabels==4)]
[184]: d35
```

[184]:

	MunicipalityName	HospitalName	HospitalLatitude	HospitalLongitude	PharmacyName	PharmacyLatitude	PharmacyLongitude	PharmacyDistance	Density	ClusterLabels
59	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	Pharmacy Angeliki Dragogianni (Φαρμακείο Αγγελ...	37.998847	23.774811	779	1236.533333	2
60	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	Pharmacy	37.994760	23.795063	1056	1236.533333	2
61	Chalandri	General Hospital of Athens "Giorgos Gennimatas	37.996760	23.783293	pharmacy ananiadis	37.996741	23.764802	1622	1236.533333	4
55	Marousi	Σισμανόγλειο Γενικό Νοσοκομείο Αττικής	38.046173	23.828541	KALTSOUNIS PHARMACY	38.038098	23.829667	904	1205.550000	2

To visualize the Hospitals as Blue circles and competitor's pharmacies are Red circles for Clusters 2 and 4, please check below map:



CONCLUSION SECTION

My proposal to my friend Pharmacist to which location to open new Pharmacy store so to be near to Hospital but far away from other pharmacies competitors and in a Municipality with big Density population is to be:

In “Chalandri” Municipality, near hospital

“General Hospital of Athens "Giorgos Gennimatas”!