Libraries installation and data uploading

```
!pip install googlemaps
!pip install simplejson
!pip install --upgrade google-auth-oauthlib
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting googlemaps
  Downloading googlemaps-4.7.3.tar.gz (32 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: requests<3.0,>=2.20.0 in /usr/local/lib/python3.8/dist-packages (from googlemaps) (2.25.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests<3.0,>=2.20.
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests<3.0,>=2.20.0->google
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests<3.0,>=2.20.0->g
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests<3.0,>=2.20.0->
Building wheels for collected packages: googlemaps
  Building wheel for googlemaps (setup.py) ... done
  Created wheel for googlemaps: filename=googlemaps-4.7.3-py3-none-any.whl size=40341 sha256=23b95f5d21707bed0305542e22f1
  Stored in directory: /root/.cache/pip/wheels/e0/57/cc/329195e1c9e5b7d270fba4671c451d111d08f7b7ef56f8ec30
Successfully built googlemaps
Installing collected packages: googlemaps
Successfully installed googlemaps-4.7.3
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting simplejson
  Downloading simplejson-3.18.1-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.5/135.5 KB 1.7 MB/s eta 0:00:00
Installing collected packages: simplejson
Successfully installed simplejson-3.18.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: google-auth-oauthlib in /usr/local/lib/python3.8/dist-packages (0.4.6)
Collecting google-auth-oauthlib
  Downloading google_auth_oauthlib-0.8.0-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: google-auth>=2.15.0 in /usr/local/lib/python3.8/dist-packages (from google-auth-oauthlib)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from google-auth-oauth
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.8/dist-packages (from google-auth>=2.15.0->google-aut
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from google-auth>=2.15.0-
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from google-auth>=2.15.0
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-packages (from google-auth>=2.15.0->google-
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0->
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0->
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.8/dist-packages (from pyasn1-modules>=0.2.1
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.0.0->req
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.0.0->requests-oau
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.0.0->reques
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.0.0->request
Installing collected packages: google-auth-oauthlib
  Attempting uninstall: google-auth-oauthlib
    Found existing installation: google-auth-oauthlib 0.4.6
    Uninstalling google-auth-oauthlib-0.4.6:
      Successfully uninstalled google-auth-oauthlib-0.4.6
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour
tensorboard 2.9.1 requires google-auth-oauthlib<0.5,>=0.4.1, but you have google-auth-oauthlib 0.8.0 which is incompatibl
Successfully installed google-auth-oauthlib-0.8.0
```

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import wilcoxon
%matplotlib inline
!python --version
import seaborn as sns
import geopy.distance
```

```
from random import random
!pip install -U geopandas
import geopandas as gpd
import shapely
from shapely.geometry import *
from geopandas import GeoDataFrame
from sklearn.cluster import DBSCAN
from geopy.distance import great_circle
from shapely.geometry import MultiPoint
from pyproj import Proj, transform
import googlemaps

import warnings
warnings.simplefilter(action='ignore')
```

```
Python 3.8.10
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting geopandas
  Downloading geopandas-0.12.2-py3-none-any.whl (1.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 14.2 MB/s eta 0:00:00
Requirement already satisfied: shapely>=1.7 in /usr/local/lib/python3.8/dist-packages (from geopandas) (2.0.0)
Collecting pyproj>=2.6.1.post1
  Downloading pyproj-3.4.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 59.1 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.8/dist-packages (from geopandas) (1.3.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-packages (from geopandas) (21.3)
Collecting fiona>=1.8
  Downloading Fiona-1.8.22-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 38.5 MB/s eta 0:00:00
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (22.2.0)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (1.15.0)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (7.1.2)
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (2022.12.7)
Collecting munch
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (57.4.0)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geopandas) (1
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geopandas) (20
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geop
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging->geopar
Installing collected packages: pyproj, munch, cligj, click-plugins, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.8.22 geopandas-0.12.2 munch-2.5.0 pyproj-3.4.1
```

```
#roads.geojson
!gdown --id 1Z8KSaAUgy7ACVe8y43eoZHKgmSCEBgLK
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1Z8KSaAUgy7ACVe8y43eoZHKgmSCEBgLK
To: /content/roads.geojson
100% 217k/217k [00:00<00:00, 65.8MB/s]
```

```
#buildings.geojson
!gdown --id 1DraoZ15VjT_PKi43hNkAzYlxh9c_4tQ1
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1DraoZ15VjT_PKi43hNkAzYlxh9c_4tQ1
To: /content/buildings.geojson
100% 143k/143k [00:00<00:00, 82.2MB/s]
```

```
#Location and built environment variables.xlsx
!gdown --id 1ADRDAbj0bN8805bRb-B2T9FX4OkJnrGn
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1ADRDAbj0bN8805bRb-B2T9FX4OkJnrGn
To: /content/Location and built environment variables.xlsx
100% 29.1k/29.1k [00:00<00:00, 39.0MB/s]
```

```
#vic_establishments.shp
!gdown --id 1WTid3_clTG1KMskm68hz1Mqw6WiBQIXB
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and
  warnings.warn(
```

```
Downloading...
From: https://drive.google.com/uc?id=1WTid3_clTG1KMskm68hz1Mqw6WiBQIXB
To: /content/vic_establishments.shp
100% 21.7k/21.7k [00:00<00:00, 29.4MB/s]
```

```
#vic_establishments.shx
!gdown --id 1yxDQJzWoNCpEj-anUt2bp0kLExgF-2Wm
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 anc
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1yxDQJzWoNCpEj-anUt2bp0kLExgF-2Wm
To: /content/vic_establishments.shx
100% 6.28k/6.28k [00:00<00:00, 9.15MB/s]
```

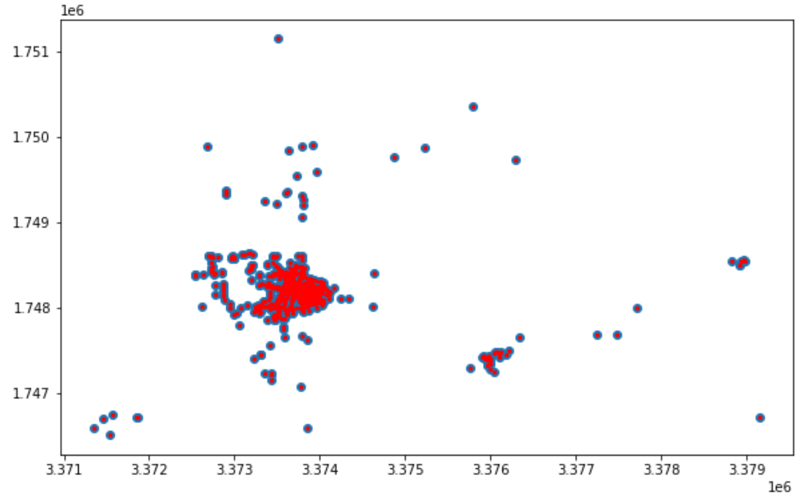
```
#vic_establishments.csv
!gdown --id 1ywVzPf0P67XlzPFKYKu1H_1Fmmqdl28h
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 anc
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1ywVzPf0P67XlzPFKYKu1H_1Fmmqdl28h
To: /content/vic_establishments.csv
100% 56.8k/56.8k [00:00<00:00, 55.6MB/s]
```

CLUSTER ANALYSIS FOR DISTANCES

```
vic = gpd.read_file("/content/vic_establishments.shx")
vic.plot(ax=vic.plot(figsize=(10, 6)), marker='o', color='red', markersize=5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7bbc57be0>
```




```
jorge = pd.read_csv('/content/vic_establishments.csv')
```

```
jorge
```

	X	Y	sid	category	subcategory	name	area

```
inProj = Proj(init='epsg:3034')
outProj = Proj(init='epsg:4326')
x1,y1 = 3373168.48041169,1748433.54613485
x2,y2 = transform(inProj,outProj,x1,y1)
print(x2,y2)
```

```
2.2477862000000357 41.93206870000004
```

```
jorge['Lon'],jorge['Lat'] = transform(inProj,outProj,jorge['X'],jorge['Y'])
```

```
df_establishments = jorge.loc[(jorge['Lon'] > 2.252) & (jorge['Lon'] < 2.2585) &
                              (jorge['Lat'] > 41.927) & (jorge['Lat'] < 41.9315), ['Lon','Lat','category','subcategory','a
```

```
df_establishments
```

	Lon	Lat	category	subcategory	area
64	2.254315	41.930450	retail	marketplace	NaN
74	2.253181	41.927347	retail	NaN	906.0
85	2.256540	41.929556	retail	convenience	134.0
121	2.256639	41.930846	retail	supermarket	722.0
126	2.253023	41.930651	retail	jewelry	228.0
...	...	...	...	...	...
766	2.254810	41.930281	services	NaN	101.0
767	2.252160	41.927119	services	NaN	454.0
769	2.253838	41.927914	services	NaN	294.0
770	2.253955	41.927728	services	NaN	435.0
771	2.254725	41.927993	services	NaN	1235.0

400 rows × 5 columns

```
df_south_establishments = df_establishments.loc[(df_establishments['Lon'] > 2.253) & (jorge['Lon'] < 2.260) &
                              (jorge['Lat'] > 41.927) & (jorge['Lat'] < 41.92937), ['Lon','Lat','category','subcategory','
```

```
df_south_establishments.index
```

```
Int64Index([ 74, 229, 322, 323, 325, 347, 348, 349, 350, 351, 396, 439, 441,
            473, 474, 475, 477, 478, 500, 501, 502, 505, 506, 513, 525, 526,
            527, 528, 529, 530, 531, 532, 533, 534, 535, 538, 553, 633, 655,
            656, 659, 662, 663, 664, 665, 666, 727, 764, 765, 769, 770, 771],
           dtype='int64')
```

```
df_establishments.drop(labels=df_south_establishments.index,axis=0,inplace=True)
```

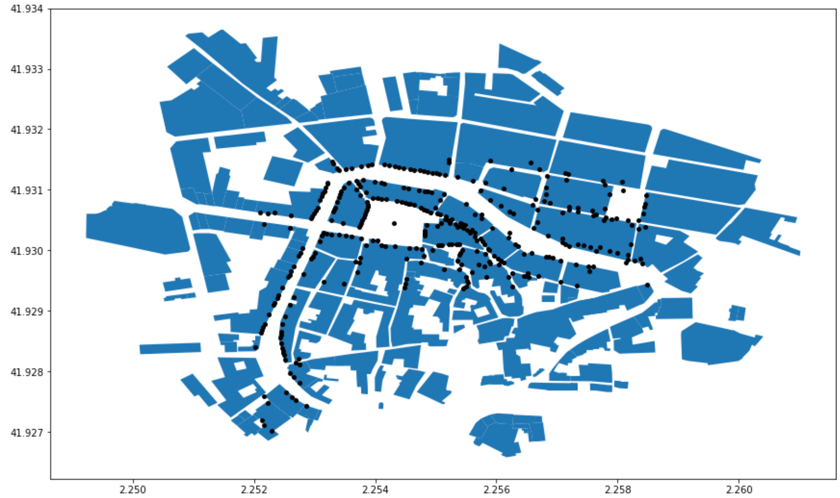
```
df_establishments
```

	Lon	Lat	category	subcategory	area
64	2.254315	41.930450	retail	marketplace	NaN
85	2.256540	41.929556	retail	convenience	134.0
121	2.256639	41.930846	retail	supermarket	722.0
126	2.253023	41.930651	retail	jewelry	228.0
127	2.253853	41.930643	retail	jewelry	121.0
...	...	...	...	...	...
750	2.256840	41.931233	services	NaN	322.0
751	2.256823	41.931045	services	NaN	322.0
763	2.254482	41.929392	services	NaN	886.0
766	2.254810	41.930281	services	NaN	101.0
767	2.252160	41.927119	services	NaN	454.0

348 rows × 5 columns

```
geometry = [Point(xy) for xy in zip(df_establishments['Lon'], df_establishments['Lat'])]
gdf = GeoDataFrame(df_establishments, geometry=geometry)
vic = gpd.read_file("/content/buildings.geojson")
gdf.plot(ax=vic.plot(figsize=(15, 9)), marker='o', color='black', markersize=15)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7bb639610>
```




```
coords = df_establishments[['Lat','Lon']]
coords2 = coords.to_numpy()
```

```
def get_centermost_point(cluster):
    centroid = (MultiPoint(cluster).centroid.x, MultiPoint(cluster).centroid.y)
    centermost_point = min(cluster, key=lambda point: great_circle(point, centroid).m)
    return tuple(centermost_point)

def nearest_neigh(row,centroids):
    dist=np.inf
    nearest=0
    point=np.array([row['Lat'],row['Lon']])
    for idx,cent in centermost_points.items():
        dist0=np.linalg.norm(point-np.array(cent))
        if dist>dist0:
            dist=dist0
            nearest=idx
    return nearest

for k in [0.047]:
    kms_per_radian = 6371.0088
    epsilon = k / kms_per_radian
    db = DBSCAN(eps=epsilon, min_samples=1, algorithm='ball_tree', metric='manhattan').fit(np.radians(coords2))
    cluster_labels = db.labels_
    num_clusters = len(set(cluster_labels))
    clusters = pd.Series([coords2[cluster_labels == n] for n in range(num_clusters)])
    print('Number of clusters: {}'.format(num_clusters),'Walking_distance: ',k)

    centermost_points = clusters.map(get_centermost_point)

    lats, lons = zip(*centermost_points)
    rep_points = pd.DataFrame({'lon':lons, 'lat':lats})
    rs = rep_points.apply(lambda row: df_establishments[(df_establishments['Lat']==row['lat']) &
                                                        (df_establishments['Lon']==row['lon'])].iloc[0], axis=1)

    fig, ax = plt.subplots(figsize=[10, 6])
    rs_scatter = ax.scatter(rs['Lon'], rs['Lat'], c='#99cc99', edgecolor='None', alpha=0.7, s=120)
    b_scatter = ax.scatter(df_establishments['Lon'], df_establishments['Lat'], c='k', alpha=0.9, s=3)
    ax.set_title('Full data set vs DBSCAN reduced set')
    ax.set_xlabel('Longitude')
    ax.set_ylabel('Latitude')
    ax.legend([b_scatter, rs_scatter], ['Full set', 'Reduced set'], loc='upper right')
    plt.show()

    df_establishments['Cluster']=df_establishments.apply(lambda x: nearest_neigh(x,centermost_points),axis=1)
    geometryn = [Point(xy) for xy in zip(rs['Lon'], rs['Lat'])]
```

```
gdfn = GeoDataFrame(rs, geometry=geometryn)
gdfn.plot(ax=vic.plot(figsize=(15, 9)), marker='o', color='red', markersize=25)

geometryb = [Point(xy) for xy in zip(df_establishments['Lon'], df_establishments['Lat'])]
gdf3 = GeoDataFrame(df_establishments, geometry=geometryb)
gdf3.plot(ax=vic.plot(figsize=(15, 9)), marker='o',c=df_establishments['Cluster'], markersize=25,alpha=0.9,edgecolor='whit
```

```
Number of clusters: 5 Walking distance:  0.047
```

```
centermost_points
```

```
0     (41.930237000000005, 2.2548144000000017)
1       (41.92718539999997, 2.252132200000016)
2      (41.92943729999995, 2.2584938000000347)
3      (41.93147910000003, 2.2558982000000665)
4      (41.93133499999998, 2.2562546999999493)
dtype: object
```

```
df_establishments
```

	Lon	Lat	category	subcategory	area	geometry	Cluster
64	2.254315	41.930450	retail	marketplace	NaN	POINT (2.25432 41.93045)	0
85	2.256540	41.929556	retail	convenience	134.0	POINT (2.25654 41.92956)	4
121	2.256639	41.930846	retail	supermarket	722.0	POINT (2.25664 41.93085)	4
126	2.253023	41.930651	retail	jewelry	228.0	POINT (2.25302 41.93065)	0
127	2.253853	41.930643	retail	jewelry	121.0	POINT (2.25385 41.93064)	0
...	...	...	...	...	...	...	...
750	2.256840	41.931233	services	NaN	322.0	POINT (2.25684 41.93123)	4
751	2.256823	41.931045	services	NaN	322.0	POINT (2.25682 41.93105)	4

```
rs
```

	Lon	Lat	category	subcategory	area	geometry
0	2.254814	41.930237	retail	clothes	101.0	POINT (2.25481 41.93024)
1	2.252132	41.927185	food	pub	454.0	POINT (2.25213 41.92719)
2	2.258494	41.929437	food	bar	412.0	POINT (2.25849 41.92944)
3	2.255898	41.931479	retail	NaN	584.0	POINT (2.25590 41.93148)
4	2.256255	41.931335	retail	NaN	722.0	POINT (2.25625 41.93133)

```
df_establishments.groupby('Cluster').count()
```

Cluster	Lon	Lat	category	subcategory	area	geometry
0	204	204	204	191	203	204
1	39	39	39	29	39	39
2	44	44	44	4	44	44
3	13	13	13	9	13	13
4	48	48	48	6	48	48

```
from sklearn.cluster import KMeans
sum_sq_d = []
K = range(1,11)

for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(df_establishments[['Lon', 'Lat']])
    sum_sq_d.append(km.inertia_)

plt.figure(figsize=(8,6))

plt.plot(K, sum_sq_d, 'rx-.')

plt.xlabel('Number of Clusters, k', fontsize=12)
plt.xticks(range(1,11), fontsize=12)

plt.ylabel('Sum of Squared Distances', fontsize=12)
plt.xticks(fontsize=12)
```

```
plt.title('Elbow Method For Determining k', fontsize=16)

plt.show()
```

```
import sklearn
from sklearn.cluster import KMeans

k = 5

kmeans = KMeans(n_clusters=k, init='k-means++')
kmeans.fit(df_establishments[['Lon', 'Lat']])

labels = kmeans.predict(df_establishments[['Lon', 'Lat']],)

centroids = kmeans.cluster_centers_
df_establishments['Classification_Kmeans'] = pd.Series(labels, index=df_establishments.index)
```

```
centroids_df = pd.DataFrame(data=centroids,columns=['Lon','Lat'])
```

```
labels_centroids = kmeans.predict(centroids)
centroids_df['Coordinates'] =  pd.Series(labels_centroids)
```

```
centroids_df
```

	Lon	Lat	Coordinates
0	2.254800	41.930691	0
1	2.253376	41.930579	1
2	2.255874	41.930122	2
3	2.252444	41.928390	3
4	2.257616	41.930380	4

```
centroids
```

```
array([[ 2.2548    , 41.93069084],
       [ 2.2533762 , 41.93057923],
       [ 2.25587433, 41.93012186],
       [ 2.25244366, 41.92838953],
       [ 2.25761623, 41.9303803 ]])
```

```
df_establishments.groupby('Classification_Kmeans').count()
```

	Lon	Lat	category	subcategory	area	geometry	Clus
Classification_Kmeans							
0	75	75	75	69	74	75	
1	87	87	87	81	87	87	
2	79	79	79	54	79	79	
3	41	41	41	31	41	41	
4	66	66	66	4	66	66	

```
rs
```

	Lon	Lat	category	subcategory	area	geometry
0	2.254814	41.930237	retail	clothes	101.0	POINT (2.25481 41.93024)
1	2.252132	41.927185	food	pub	454.0	POINT (2.25213 41.92719)
2	2.258494	41.929437	food	bar	412.0	POINT (2.25849 41.92944)
3	2.255898	41.931479	retail	NaN	584.0	POINT (2.25590 41.93148)
4	2.256255	41.931335	retail	NaN	722.0	POINT (2.25625 41.93133)

```
df_establishments
```

	Lon	Lat	category	subcategory	area	geometry	Cluster	Class
64	2.254315	41.930450	retail	marketplace	NaN	POINT (2.25432 41.93045)	0	
85	2.256540	41.929556	retail	convenience	134.0	POINT (2.25654 41.92956)	4	
121	2.256639	41.930846	retail	supermarket	722.0	POINT (2.25664 41.93085)	4	
126	2.253023	41.930651	retail	jewelry	228.0	POINT (2.25302 41.93065)	0	
127	2.253853	41.930643	retail	jewelry	121.0	POINT (2.25385	0	

```
geometryb = [Point(xy) for xy in zip(df_establishments['Lon'], df_establishments['Lat'])]
gdf3 = GeoDataFrame(df_establishments, geometry=geometryb)
gdf3.plot(ax=vic.plot(figsize=(15, 9)), marker='o',c=df_establishments['Classification_Kmeans'], markersize=25,alpha=0.9,edgec

geometryb = [Point(xy) for xy in zip(centroids_df['Lon'], centroids_df['Lat'])]
gdf3 = GeoDataFrame(centroids_df, geometry=geometryb)
gdf3.plot(ax=vic.plot(figsize=(15, 9)), marker='o', markersize=25,alpha=0.9,edgecolor='red',facecolor='red')
```

```
CENT_DBSCAN = {0:"2.254814,41.930237",1:"2.252132,41.927185",2:"2.258494,41.929437",
               3:"2.255898,41.931479",4:"2.256255,41.931335"}
CENT_KMEANS = {0:"2.25583343, 41.93008263",1:"2.25336722, 41.93056809",2:"2.2524401 , 41.9283593",
               3:"2.25756052, 41.93040757",4:"2.2548    , 41.93069084"}
```

```
df_establishments['Centr_DBSCAN'] = df_establishments['Cluster'].map(CENT_DBSCAN)
```

```
df_establishments['Centr_KMEANS'] = df_establishments['Classification_Kmeans'].map(CENT_KMEANS)
```

```
df_establishments['Coordinates'] = df_establishments['Lon'].map(str) + ","+ df_establishments['Lat'].map(str)
```

```
df_establishments
```

	Lon	Lat	category	subcategory	area	geometry	Cluster	Class
64	2.254315	41.930450	retail	marketplace	NaN	POINT (2.25432 41.93045)	0	
85	2.256540	41.929556	retail	convenience	134.0	POINT (2.25654 41.92956)	4	
121	2.256639	41.930846	retail	supermarket	722.0	POINT (2.25664 41.93085)	4	
126	2.253023	41.930651	retail	jewelry	228.0	POINT (2.25302 41.93065)	0	
127	2.253853	41.930643	retail	jewelry	121.0	POINT (2.25385 41.93064)	0	
...	...	...	...	...	...	...	...	

```
#source https://www.linkedin.com/pulse/calculating-distances-using-python-google-maps-r%C3%A9gis-nisengwe?articleId=6625061973
API_key = 'XXX'   #enter the key you got from Google. I put mine here

gmaps = googlemaps.Client(key=API_key)

df_establishments['DB_distance'] = df_establishments.apply(lambda row : gmaps.distance_matrix(row['Centr_DBSCAN'], row['Coordi
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)

df_establishments['KM_distance'] = df_establishments.apply(lambda row : gmaps.distance_matrix(row['Centr_KMEANS'], row['Coordi
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
```

```
df_establishments['DB_distance'].describe()
```

```
count    348.000000
mean      62.304598
std       48.222220
min        0.000000
25%       28.000000
50%       58.000000
75%       79.000000
max      267.000000
Name: DB_distance, dtype: float64
```

```
df_establishments[df_establishments['KM_distance']>75].count()
```

```
Lon                      56
Lat                      56
category                 56
subcategory              31
area                     56
geometry                 56
Cluster                  56
Classification_Kmeans    56
Centr_DBSCAN             56
Centr_KMEANS             56
Coordinates              56
DB_distance              56
KM_distance              56
dtype: int64
```

```
df_establishments[df_establishments['DB_distance']>75].count()
```

```
Lon                      113
Lat                      113
category                 113
subcategory               79
area                     113
geometry                 113
Cluster                  113
Classification_Kmeans    113
Centr_DBSCAN             113
Centr_KMEANS             113
Coordinates              113
DB_distance              113
KM_distance              113
dtype: int64
```

```
df_establishments.groupby(by='Cluster').agg({'area':'mean'})
```

Cluster	area
0	224.802956
1	292.692308
2	424.318182
3	386.230769
4	352.916667

```
df_establishments.groupby(by='Cluster').agg({'area':'std'})
```

Cluster	area
0	152.224758
1	129.033891
2	154.881766
3	153.049096
4	185.833354

```
df_establishments['Centr_KMEANS'].value_counts()
```

```
2.25336722, 41.93056809    88
2.2548    , 41.93069084    75
2.25583343, 41.93008263    75
2.25756052, 41.93040757    70
2.2524401 , 41.9283593     40
Name: Centr_KMEANS, dtype: int64
```

```
df_establishments.groupby(by='Classification_Kmeans').agg({'Classification_Kmeans':'count'})
```

Classification_Kmeans	Classification_Kmeans
0	75
1	88
2	40
3	70
4	75

```
df_establishments.groupby(by='Classification_Kmeans').agg({'area':'std'})
```

area

```
df_establishments.groupby(by='Classification_Kmeans').agg({'area':'mean'})
```

	area
Classification_Kmeans	
0	229.026667
1	240.659091
2	293.250000
3	418.142857
4	247.648649

```
df_establishments.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 348 entries, 64 to 767
Data columns (total 13 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Lon                    348 non-null    float64
 1   Lat                    348 non-null    float64
 2   category               348 non-null    object
 3   subcategory            239 non-null    object
 4   area                   347 non-null    float64
 5   geometry               348 non-null    geometry
 6   Cluster                348 non-null    int64
 7   Classification_Kmeans  348 non-null    int32
 8   Centr_DBSCAN           348 non-null    object
 9   Centr_KMEANS           348 non-null    object
 10  Coordinates            348 non-null    object
 11  DB_distance            348 non-null    int64
 12  KM_distance            348 non-null    int64
dtypes: float64(3), geometry(1), int32(1), int64(3), object(5)
memory usage: 36.7+ KB
```

```
LZ1 = (2.257956913777889,41.92980661304041)
LZ2 = (2.2570145861240007,41.93007216599587)
LZ3 = (2.2562189136567667,41.93044589346633)
LZ4 = (2.255090232524026,41.93114375772215)
LZ5 = (2.2533523604093944,41.93103082640491)
LZ6 = (2.252704186417077,41.92960302034483)
LZ7 = (2.252546089056864,41.92928249137102)
LZ8 = (2.2527329960227007,41.9276266528334)

df_establishments['LZ1'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ1, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ2'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ2, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ3'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ3, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ4'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ4, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ5'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ5, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ6'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ6, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ7'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ7, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
df_establishments['LZ8'] = df_establishments.apply(lambda row : gmaps.distance_matrix(LZ8, row['Coordinates'],
                          mode="walking")["rows"][0]["elements"][0]["distance"]["value"], axis = 1)
```

```
df_establishments
```

	Lon	Lat	category	subcategory	area	geometry	Cluster	Classification_Kmeans	Centr_DBSCAN	Cent
64	2.254315	41.930450	retail	marketplace	NaN	POINT (2.25432 41.93045)	0	2	2.254814,41.930237	2.25485771,41
85	2.256540	41.929556	retail	convenience	134.0	POINT (2.25654 41.92956)	4	1	2.256255,41.931335	2.25583706,41
121	2.256639	41.930846	retail	supermarket	722.0	POINT (2.25664 41.93085)	4	4	2.256255,41.931335	2.25754692,4
126	2.253023	41.930651	retail	jewelry	228.0	POINT (2.25302 41.93065)	0	0	2.254814,41.930237	2.25339331,41
127	2.253853	41.930643	retail	jewelry	121.0	POINT (2.25385 41.93064)	0	0	2.254814,41.930237	2.25339331,41
...	...	...	...	...	...	...	...	...	...	
750	2.256840	41.931233	services	NaN	322.0	POINT (2.25684 41.93123)	4	4	2.256255,41.931335	2.25754692,4
751	2.256823	41.931045	services	NaN	322.0	POINT (2.25682 41.93105)	4	4	2.256255,41.931335	2.25754692,4

```
from google.colab import files
df_establishments.to_excel('df_establishments.xls')
files.download('df_establishments.xls')
```

41.93028)

```
df_establishments.columns
```

```
Index(['Lon', 'Lat', 'category', 'subcategory', 'area', 'geometry', 'Cluster',
       'Classification_Kmeans', 'Centr_DBSCAN', 'Centr_KMEANS', 'Coordinates',
       'DB_distance', 'KM_distance', 'LZ1', 'LZ2', 'LZ3', 'LZ4', 'LZ5', 'LZ6',
       'LZ7', 'LZ8'],
      dtype='object')
```

```
ftg = df_establishments[['category', 'subcategory', 'area', 'Cluster',
       'Classification_Kmeans', 'LZ1', 'LZ4', 'LZ5', 'LZ6', 'LZ8']]
```

	category	subcategory	area	Cluster	Classification_Kmeans	LZ1	LZ4	LZ5
64	retail	marketplace	NaN	0	2	230	115	28
85	retail	convenience	134.0	4	4	130	15	128
121	retail	supermarket	722.0	4	0	19	96	239
126	retail	jewelry	228.0	0	1	287	173	30
127	retail	jewelry	121.0	0	1	250	136	8
...	...	...	...	...	...	...	...	...
750	services	NaN	322.0	4	0	28	142	285
751	services	NaN	322.0	4	0	11	126	269
763	services	NaN	886.0	0	2	287	173	30
766	services	NaN	101.0	0	2	205	91	53
767	services	NaN	454.0	1	3	642	528	384

348 rows × 10 columns