Population 2019 Paris and name districts

Paris is a famous city. but the people don’t know about the district.
With my proyect you can learrn about it, we are going to learn the name of the district and the population. We ae going to use two types of tables of data: data coordinates and data population

Data coordinates:

First of all we need the location of each district:

	Code_commune_INSEE	Nom_commune	Code_postal	Libellé_d_acheminement	coordonnees_gps
0	75106	PARIS 06	75006	PARIS	48.8489680919,2.33267089859
1	75112	PARIS 12	75012	PARIS	48.8351562307,2.41980703497
2	75113	PARIS 13	75013	PARIS	48.8287176845,2.36246822852
3	75118	PARIS 18	75018	PARIS	48.8927350746,2.34871193387
4	75102	PARIS 02	75002	PARIS	48.8679033789,2.34410716666

In the last table , the coordinates are in the same column, we need to divise them in tho column
In Python:

```
#divise gps columns and latitude and longitude
df[["latitude", "longitude"]] = df.coordonnees_gps.str.split(",",expand=True,)
```

Also we need floats , so we are going to convert the new columns:

```
df["latitude"] = df.latitude.astype(float)
df["longitude"] = df.longitude.astype(float)
```

Data population:

To known more information abour each district we are going to use a table of the population in each district

	Nom de la commune	Population municipale 2019 (Pop légale 2016)	Population municipale 2018 (Pop légale 2015)	Population municipale 2016 (Pop légale 2013)	Progression sur 1 an	Progression sur 3 ans	Progression en valeur absolue sur 3 ans	Code_commune_INSEE
0	Paris 15e Arrondissement	233484	234994	237120	-0.64	-1.53	-3636	75115
1	Paris 20e Arrondissement	195604	195556	194771	0.02	0.43	833	75120
2	Paris 18e Arrondissement	195060	197580	199519	-1.28	-2.23	-4459	75118
3	Paris 19e Arrondissement	186393	185654	185953	0.40	0.24	440	75119
4	Paris 13e Arrondissement	181552	183216	183713	-0.91	-1.18	-2161	75113

Chart: 94 Citoyens Source: Insee Get the data Created with Datawrapper

After explain the data, we are going to combine the two source of information to have a new table with the data of location and population.

In python:

```
#merge the two csv
paris_merged = df
```

```
paris_merged = paris_merged.join(df_population.set_index('Code_commune_INSEE'),
on='Code_commune_INSEE')
paris_merged.head() # see the new data
```

The result of this combination is in this new dataframe named paris_merged:

	Code_commune_INSEE	Nom_commune	Code_postal	Libellé_d_acheminement	coordonnees_gps	latitude	longitude	Nom de l commun
0	75106	PARIS 06	75006	PARIS	48.8489680919,2.33267089859	48.8489680919	2.332671	Paris 6 Arrondissemer
1	75112	PARIS 12	75012	PARIS	48.8351562307,2.41980703497	48.8351562307	2.419807	Paris 12 Arrondissemer
2	75113	PARIS 13	75013	PARIS	48.8287176845,2.36246822852	48.8287176845	2.362468	Paris 13 Arrondissemer
3	75118	PARIS 18	75018	PARIS	48.8927350746,2.34871193387	48.8927350746	2.348712	Paris 18 Arrondissemer
4	75102	PARIS 02	75002	PARIS	48.8679033789,2.34410716666	48.8679033789	2.344107	Paris 2 Arrondissemer

Now there are many columns, we can delete some to have the useful data:
In python:
paris_merged.drop(['Code_commune_INSEE', 'Libellé_d_acheminement', 'coordonnees_gps'], axis='columns', inplace=True)

The result is:

	Nom_commune	Code_postal	latitude	longitude	Nom de la commune	Population municipale 2019 (Pop légale 2016)	Population municipale 2018 (Pop légale 2015)	Population municipale 2016 (Pop légale 2013)	Progression sur 1 an	Progression sur 3 ans	Progression en valeur absolue sur 3 ans
0	PARIS 06	75006	48.848968	2.332671	Paris 6e Arrondissement	40916	42428	43479	-3.56	-5.89	-2563
1	PARIS 12	75012	48.835156	2.419807	Paris 12e Arrondissement	141494	142340	144719	-0.59	-2.23	-3225
2	PARIS 13	75013	48.828718	2.362468	Paris 13e Arrondissement	181552	183216	183713	-0.91	-1.18	-2161
3	PARIS 18	75018	48.892735	2.348712	Paris 18e Arrondissement	195060	197580	199519	-1.28	-2.23	-4459
4	PARIS 02	75002	48.867903	2.344107	Paris 2e Arrondissement	20260	20796	21741	-2.58	-6.81	-1481

Visualization with folium:

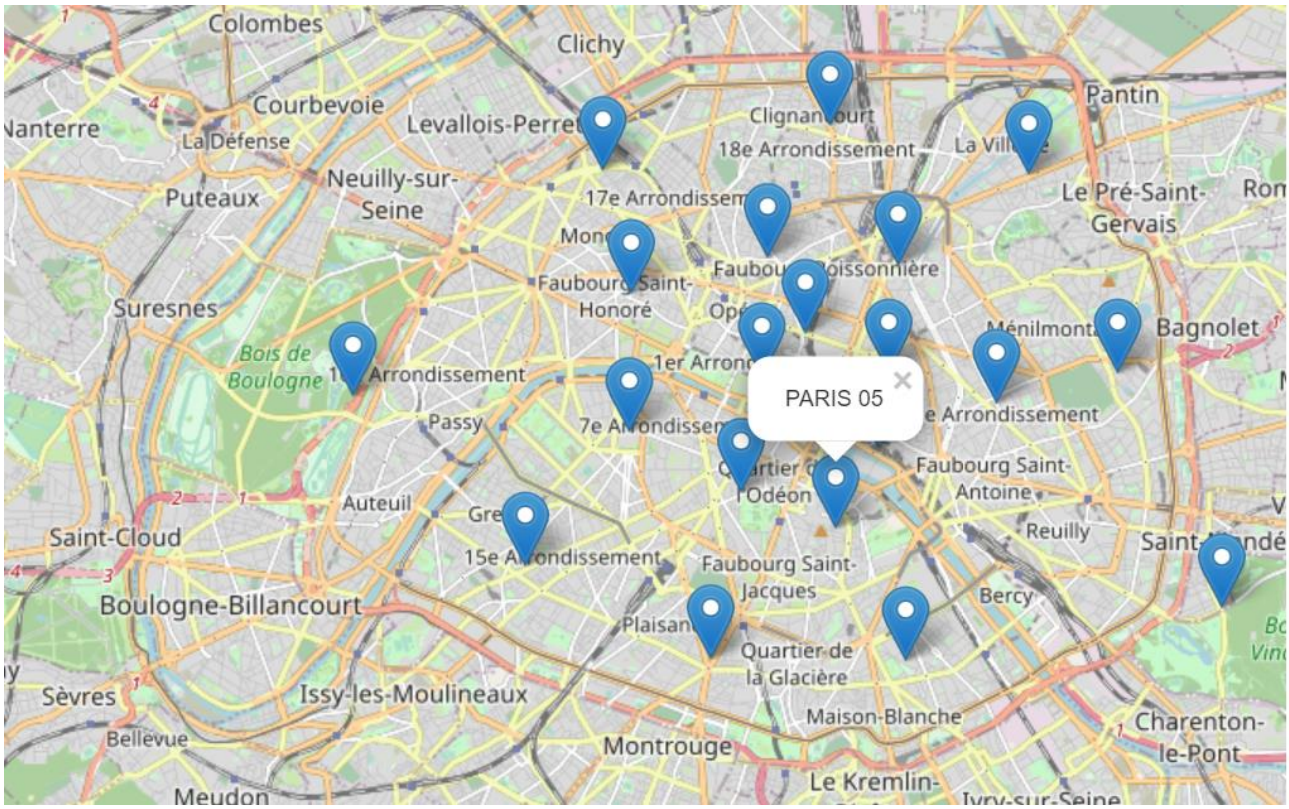
With the district data we are going to see the markers of each district, click in the marker and you will be obtein the district’s name

in Python:

```
# combined data
geolocator = Nominatim(user_agent="Paris_explorer")
address = "Paris"
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
latlng = (48.85837, 2.2944813)
map_paris = folium.Map(location=[latitude, longitude], zoom_start=10)
```

```
# add markers to map
locations = df[['latitude', 'longitude']]
locationlist = locations.values.tolist()
len(locationlist)
map = folium.Map(location=latlng, zoom_start=12)
for point in range(0, len(locationlist)):
  # label = '{}, {}'.format(Code_postal, Nom_commune)
   #label = folium.Popup(label, parse_html=True)
   folium.Marker(locationlist[point], popup=df['Nom_commune'][point]).add_to(map)
map
```

Result of the code execution:



Population

Finally we are going to represent the population with a circle et the location. If the circle is bigger it means the population is bigger

In Python:

```
# create map of Paris  using latitude and longitude values
geolocator = Nominatim(user_agent="Paris_explorer")
address = "Paris"
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
latlng = (48.85837, 2.2944813)
map_paris = folium.Map(location=[latitude, longitude], zoom_start=10)
paris_merged['Population municipale 2019 (Pop légale 2016)'] = paris_merged['Population
municipale 2019 (Pop légale 2016)'].astype(float)
# add markers to map
locations = paris_merged[['latitude', 'longitude']]
```

```
locationlist = locations.values.tolist()
len(locationlist)
map = folium.Map(location=latlng, zoom_start=12)
# I can add marker one by one on the map
for  point in range(0, len(locationlist)):

    folium.Circle(
      location=locationlist[point],
      popup=paris_merged.iloc[point]['Nom_commune'],
      radius= (paris_merged['Population municipale 2019 (Pop légale 2016)'][point])/900,
      color='crimson',
      fill=True,
      fill_color='crimson'
   ).add_to(map)
map
```

Result of the code execution: