Badji Mokhtar Annaba University
Department of Urban Planning

Master in Smart Urbanism, Semester 1 — UEM1.1.2: Programming Workshop

Lab Session 2: Interactive Population Density Map

Instructor: Dr. Hatem Zehir — Level: Master 1

Objectives

Explore the **population density** of the 12 communes of Annaba wilaya on an interactive map. Run code of each cell, read the results, **answer the questions**.

Set-up

Open **Jupyter Notebook** (Anaconda) or **Google Colab**. **Each box is one cell**: copy it into its own cell and run it (`Shift+Enter`) before the next. Internet and pandas $\geq$ 2.1 needed (`!pip install -U pandas`).

Cell 1 — install (once) and import

```
!pip install geopandas folium osmnx lxml
import pandas as pd, geopandas as gpd, osmnx as ox, folium
```

Exercise 1: Download and join the data

Cells 2 and 3 download the data (it can take up to a minute). Cell 4 joins them on the commune name.

Cell 2 — population table

```
url = "https://citypopulation.de/en/algeria/admin/23__annaba/"
tables = pd.read_html(url, storage_options={"User-Agent": "Mozilla/5.0"})
pop = tables[0].iloc[:, [0, 1, 4]].copy()   # name, status, population 2008
pop.columns = ["key", "status", "population"]
pop["key"] = pop["key"].str.normalize("NFKD").str.encode("ascii", "ignore").str.decode("ascii").str.lower()
pop = pop[pop["status"].isin(["Commune", "Province Capital"])]
print("Total population:", pop["population"].sum())
pop.sort_values("population", ascending=False)
```

Cell 3 — commune boundaries (OpenStreetMap, 12 communes)

```
ids = {1616090: "ain berda", 1616109: "annaba", 1616145: "barrahal", 1616147: "chetaibi",
       1616091: "cheurfa", 1616108: "el bouni", 1616083: "el eulma", 1616094: "el hadjar",
       1616122: "oued el aneb", 1616121: "seraidi", 1616146: "sidi amar", 1616127: "treat"}
communes = ox.geocode_to_gdf(["R" + str(i) for i in ids], by_osmid=True)
communes["key"] = communes["osm_id"].map(ids)
communes = communes[["key", "geometry"]]
communes.plot()
```

Cell 4 — join the two tables

```
data = communes.merge(pop, on="key")
print(len(data), "communes joined")
```

Questions.

(a) How many communes? Total population?
(b) Largest and smallest population?
(c) Are all communes the same size?
(d) How many communes were joined?
(e) Which communes have under 10,000 inhabitants?

Exercise 2: Compute the density and draw the map

Density = population $\div$ area (km^2). Cell 6 saves `density_map.html`.

Cell 5 — area and density

```
data["area_km2"] = data.to_crs(32632).area / 1000000     # area in km2 (UTM 32N)
data["density"] = data["population"] / data["area_km2"]
print("Mean density:", data["density"].mean())
data[["key", "population", "area_km2", "density"]].sort_values("density", ascending=False)
```

Cell 6 — choropleth map and HTML export

```
m = folium.Map(location=[36.85, 7.6], zoom_start=10, tiles="Esri.WorldGrayCanvas")
folium.Choropleth(geo_data=data, data=data, columns=["key", "density"],
                  key_on="feature.properties.key", fill_color="YlOrRd",
                  legend_name="Density (inhabitants per km2)").add_to(m)
m.save("density_map.html")
m
```

Questions.

(a) Highest and lowest density?
(b) Mean density? How many communes above it?
(c) Is the most populated commune also the densest?
(d) Legend range? Which colour means high density?
(e) Open `density_map.html`. Where is the population concentrated, and why?

Deliverables (teams of 4, one week): **report** (PDF, 5 pages max: answers + map screenshot), **notebook** (`.ipynb`) and `density_map.html`.