

Python

- žádný objekt s více než 1000 bytovými jednotkami v datasetu není 😊 (možná překlep?)

Ale kdybychom změnili počet HP (bytových jednotek) třeba na 50, tak bych použila knihovnu pandas a geodesic (přemýšlela jsem i nad explicitní haversine formula, ale to mi přišlo zbytečně složité) a pro clustery KMeans (vyhledala jsem si, co se pro clusterování používá a Kmeans vyšlo jako nejobecnější)

- importovala jsem knihovny pandas, geodesic a Kmeans
- načetla jsem si soubor .csv (který jsem původně chtěla zpracovávat v SQL, ale pak jsem si vybrala Python – nechtěla jsem se zaplést do radiánů), vyměnila separátor (středník), změnila kódování (na češtinu) a zkontrolovala jsem ho
- nahradila jsem čárky tečkami ve sloupcích longitude, latitude a fbb_penetration_desetinne_cislo, přetypovala na desetinné číslo a odstranila řádky, ve kterých nejsou uvedené souřadnice
- vytvořila jsem si funkci *compute_distances()*, která iteruje přes města jednotlivé vzdálenosti – vypočítá vzdálenost od daného města ke každému objektu, dokud nebudou vyčerpána všechna města (město = city, souřadnice = coords), použila jsem funkci geodesic
- vytvořila jsem si funkci *filter_city()*, která filtruje objekty do vzdálenosti 100 km, které mají méně než 30 % penetrace a alespoň 50 bytových jednotek – funkce zároveň vytvoří i sloupec s městem a vytvořila jsem si kopii, aby se to nepletlo s původním df
- vytvořila jsem si funkci *cluster_objects()* pro geografické seskupení již vyfiltrovaných dat, rozdělila jsem pomocí Kmeans objekty do clusterů a spočítala, kolik domácností je v každé skupině, přidala jsem nový sloupec cluster
- vytvořila jsem si slovník cities se souřadnicemi měst
- pak jsem spustila hlavní funkci