Finální část projektu

Tato část projektu navazuje na první část a druhou část. Cílem je analyzovat a vizualizovat data (Statistika nehodovosti Policie ČR), která máme již stažena a předzpracována z minulých částí.

Vstupní data

Aby byly stejné podmínky pro všechny a nebylo nutné opravovat chyby v předchozích částí projektu, budeme na základě vašich požadavků v dotazníku k předmětu **pracovat se souborem accidents.pkl.gz**, který stáhnete z adresy: http://ehw.fit.vutbr.cz/izv/accidents.pkl.gz .

Serializovaný DataFrame obsahuje data organizovaná ve sloupích, které jsou pojmenované tak, jak jsou uvedeny v popisu datového souboru (*p1*, *p13a*, …). Data jsou uložena jako `float`, `int` nebo `str` podle sloupce. Neznámé hodnoty jsou u typů float reprezentované jako `np.nan`, u int hodnotou `-1` a v případě řetězců jako prázdný řetězec.

Jedná se o stejný soubor jako v případě druhé části projektu, datové typy nejsou převedeny na kategorické a datum vzniku nehody je uloženo jako řetězec nikoliv `datetime64`. Počítejte, že všechny skripty budou spouštěny s tímto souborem na vstupu a pokud je v rámci implementace zapotřebí provést změny (např. přetypování), musíte je udělat ve svém skriptu po načtení dat.

Odevzdávání a hodnocení

Soubory `geo.py, stat.ipynb, doc.py a doc.pdf` odevzdejte do 15. 1. 2021.
Hodnotit se bude zejména:

- správnost výsledků
- dodržení požadavků zadání
- vizuální zpracování grafů
- kvalita kódu
    - efektivita implementace (nebude hodnocena rychlost, ale bude kontrolováno, zda nějakým způsobem řádově nezvyšujete složitost)
    - přehlednost kódu
    - dodržení standardů a zvyklostí pro práci s jazykem Python (PEP8)
    - dokumentace kódu

Celkem lze získat až 60 bodů, přičemž k zápočtu je nutné získat ze všech částí projektů a bonusových úkolů minimálně 50 bodů.

Úkol 1: Vizualizace geografických dat (až 20 bodů)

Vytvořte soubor `geo.py`, který bude obsahovat jednu funkci pro vytváření geografického datového rámce a dvě funkce pro vizualizaci dat. Vyberte si jeden kraj, se kterým budete v tomto úkolu pracovat.

Vytváření geografických dat

Signatura funkce

```
def make_geo(df: pd.DataFrame) -> geopandas.GeoDataFrame:
```

Funkcionalita

Tato funkce vytvoří GeoDataFrame ze vstupního DataFrame tak, že **správně nastaví CRS** a odstraní všechny řádky, u kterých je pozice nehody neznámá. Argument `df` odpovídá DataFrame ze souboru *accidents.pkl.gz*.

Pozice nehod

Signatura funkce

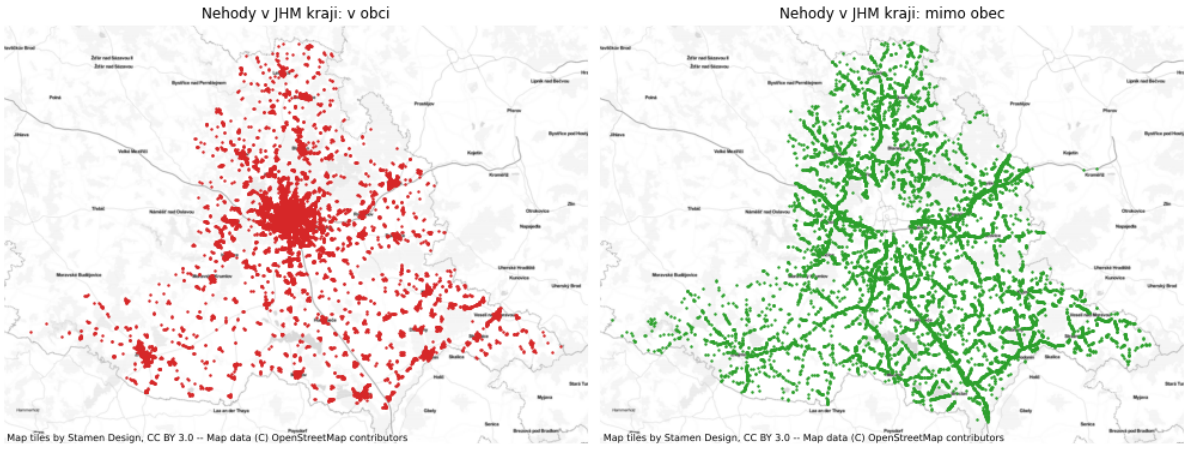
```
def plot_geo(gdf: geopandas.GeoDataFrame, fig_location: str = None,
             show_figure: bool = False):
```

Funkcionalita

Vykreslete graf obsahující dva podgrafy, kde znázorněte pozici nehod ve vybraném kraji podle sloupců *d* a *e* (souřadnic v systému **S-JTSK**) podle lokality nehody (sloupec *p5a*) - levý podgraf bude zobrazovat nehody v obci, pravý mimo obec. Použijte vhodnou podkladovou mapu a vhodnou projekci pro výsledné zobrazení, aby nedošlo k deformaci podkladové mapy. Podgrafy označte, nastavte správně limity os a celkově grafický vzhled.

Argument `gdf` odpovídá GeoDataFrame ze souboru *accidents.pkl.gz* zpracovaných funkcí `make_geo`. Graf uložte do souboru specifikovaného argumentem `fig_location` a případně zobrazte pokud `show_figure` je `True`.

Ukázka možného výstupu



Četnost nehod

Signatura funkce

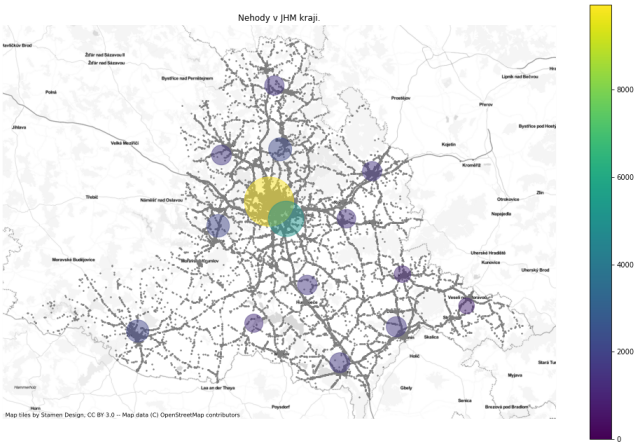
```
def plot_cluster(gdf: geopandas.GeoDataFrame, fig_location: str = None,
                 show_figure: bool = False):
```

Funkcionalita

Vykreslete všechny nehody ve vybraném kraji tak, že nehody blízko sebe budete shlukovat do jednoho clusteru a zobrazovat pomocí kruhové značky. Velikost značky a její barva bude odpovídat počtu nehod v daném clusteru. Bude zobrazený barevný sloupec ukazující, jakému počtu nehod odpovídá která barva (tzv. colorbar). Dále zobrazte na pozadí také malými body všechny nehody v kraji.

Vyberte vhodnou shlukovací metodu a počet clusterů tak, aby byla zachována přehlednost a vypovídající schopnost grafu. Pro argumenty platí stejná pravidla, jako pro funkci `plot_geo`.

Ukázka možného výstupu



Odevzdání úkolu

- odevzdává se jeden soubor `geo.py`
- soubor bude importován jako knihovna, veškeré operace mimo definice funkce budou prováděny pouze pokud je skript spuštěn přímo, tzn. v rámci podmínky
  `if __name__ == "__main__":`
- šablona tohoto souboru je v dokumentovém skladu WISu
- není třeba tvořit totožné grafy jako v ukázkách, vlastní invence se cení.
- vyhodnocení bude probíhat následovně:

```
import geo
import pandas as pd

gdf = geo.make_geo(pd.read_pickle("accidents.pkl.gz"))
geo.plot_geo(gdf, "geo1.png", False)
geo.plot_cluster(gdf, "geo2.png", False)
```

Úkol 2: Test hypotézy (až 10 bodů)

Vytvořte Jupyter notebook `stat.ipynb`, ve kterém budete ověřovat s 95% jistotou následující hypotézu:

Pokud viník nehody byl pod silným vlivem alkoholu, došlo častěji k těžkým zdravotním následkům.

K ověření hypotézy využijte χ^2 test s tím, že:

- viník nehody byl pod silným vlivem alkoholu, pokud *p11 >= 7,*
- těžké zdravotní následky nastaly, pokud došlo k usmrcení či těžkému zranění (*p13a + p13b > 0*) a
- vyloučíte (z celého `DataFrame`) nehody pod vlivem drog (*p11 == 4* nebo *p11 == 5*)

Předpokládejte, že soubor `accidents.pkl.gz` je ve stejném adresáři, jako vámi vytvořený notebook.

Odevzdávání

- odevzdáváte jeden soubor `stat.ipynb`
- všechny kroky, dílčí výsledky a závěry musí být **komentované** v Markdown buňkách.
- rozlišujte mezi buňkami v Jupyter notebooku typu `code` a `markdown`.
- před odevzdáním řešení zkuste restartovat celý notebook a spustit všechny buňky (*Run all cells*). Žádná nesmí skončit výjimkou.

Úkol 3: Vlastní analýza (až 30 bodů)

Cílem je vytvořit **zprávu či infografiku**, která hodnotí vámi vybrané ukazatele (parametry nehod) a upozorňuje čtenáře na jejich vliv (např. co je častou příčinou nehod, má-li vliv počasí na nehodovost, …). Pracujte s daty ze zpracovávané statistiky nehodovosti v ČR.

Vámi zvoleným postupem (programové generování, využití Latex, Office, Inkscape, …) vytvořte dokument `doc.pdf` o rozsahu jedné strany A4, který bude obsahovat minimálně:

- jeden graf,
- jednu tabulku,
- textový popis o rozsahu minimálně 6 vět,
- v textu minimálně další 3 vypočtené hodnoty,

Dále odevzdejte soubor `doc.py`, který bude pracovat se souborem `accidents.pkl.gz` ve stejném adresáři jako skript, který

- vygeneruje do souboru `fig.xxx` ve stejném adresáři graf v takové podobě, v jaké je použitý v dokumentu doc.pdf (např. PDF, SVG, PNG …, kde `xxx` odpovídá koncovce použitého formátu)
- vypíše na standardní výstup data pro tabulku a to buď v textovém formátu se sloupci oddělenými tabulátory, nebo ve specializovaném formátu (Latex, CSV, ...), pokud je takto tabulka použita v rámci dokumentu.
- vypíše na standardní výstup hodnoty použité v textu i s jednoduchým popisem. Např:

```
nehod celkem: 81501
nehod na lede: 12101
```

- může výstup také zapsat do souborů (např. `tab.tex`), ovšem při hodnocení bude brán v potaz pouze standardní výstup.

Hodnocení:
Při hodnocení se budou zohledňovat následující prvky:

- zda jste dokázali sami vybrat vhodná data, správně je vizualizovat a interpretovat
- vhodnost a věcnost komentářů
- vizuální dojem
- kvalita kódu pro generování

Odevzdávání:

- dokument `doc.pdf` s výslednou zprávou / infografikou
- soubor `doc.py`, který generuje vstupní data pro vytváření zprávy

Poznámky k implementaci

Stručnou dokumentaci všech částí (souboru a funkcí) uveďte přímo v odevzdaných souborech. Respektuje konvenci pro formátování kódu PEP 257 [PEP 257 -- Docstring Conventions] a PEP 8 [PEP 8 -- Style Guide for Python Code].

Graf by měl splňovat všechny náležitosti, které u grafu očekáváme, tj. měl by být přehledný a jeho velikost by měla být taková, aby se dal čitelně použít v šířce A4 (t.j. cca 18 cm). Toto omezení není úplně striktní, ale negenerujte grafy, které by byly přes celý monitor.

Primárně se počítá, že budete pracovat s následujícími externími knihovnami:
numpy, pandas, seaborn, matplotlib, scipy, scikit-learn, geopandas, contextily
a použijete techniky představené na přednáškách. Můžete použít libovolné knihovny představené při přednáškách, další pouze po schválení.

Výsledky budou vyhodnoceny na počítači se systémem Ubuntu 20.10, v prostředí Python 3.8 a s knihovnami v posledních stabilních verzích dle PIP.

Dotazy a připomínky

Dotazy a připomínky směřujte na fórum ve WIS případně na mail mrazek@fit.vutbr.cz.