Finální část projektu - varianta Nehody

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 nastaveny stejné podmínky pro všechny a nebylo nutné opravovat chyby v předchozích částí projektu, budeme na **pracovat se soubory ve formátu pkl.gz**, které stáhnete z adres: http://ehw.fit.vutbr.cz/izv/accidents.pkl.gz, http://ehw.fit.vutbr.cz/izv/locations.pkl.gz, http://ehw.fit.vutbr.cz/izv/vehicles.pkl.gz, http://ehw.fit.vutbr.cz/izv/consequences.pkl.gz, http://ehw.fit.vutbr.cz/izv/pedestrians.pkl.gz
Stahování nebude součástí skriptů, pro poslední úkol předpokládejte, že všechny tyto soubory budou ve stejném adresáři, jako váš skript.

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/html` odevzdejte do 4. 1. 2026.
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

```python
def make_geo(df_accidents: pd.DataFrame, df_locations: pd.DataFrame
             ) -> geopandas.GeoDataFrame:
```

Funkcionalita
Tato funkce vytvoří GeoDataFrame ze vstupního DataFrame tak, že spojí dataframe **správně nastaví CRS** a odstraní všechny řádky, u kterých je pozice nehody neznámá. Argument **`df_accidents`** odpovídá DataFrame ze souboru *accidents.pkl.gz*, nebude filtrovat kraje (tj. vrátí všechny kraje). Argument **`df_locations`** odpovídá DataFrame ze souboru *locations.pkl.gz*. Pozor, vynechejte nulové pozice d,e (neznámé) a pokud d < e, tak došlo k záměně souřadnic *x* a *y* a je nutné v těchto případech **souřadnice prohodit**! Zkontrolujte si, že všechny pozice jsou v ČR a nejsou v např. v severním Německu (prohozené souřadnice) či v Pobaltí (nulové souřadnice) - např. vykreslením všech bodů.

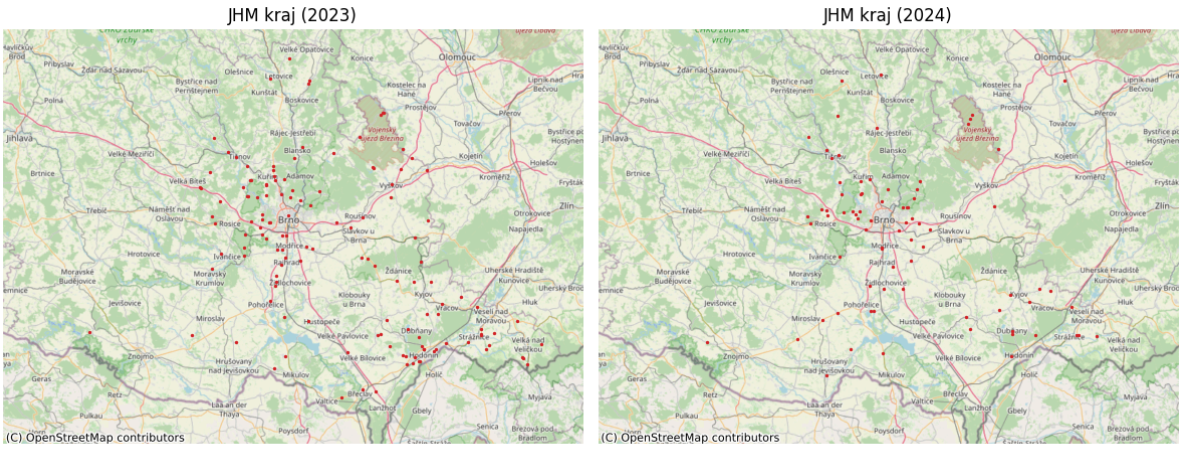
Pozice nehod
Signatura funkce

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

Funkcionalita
Vykreslete graf obsahující dvou podgrafů, kde znázorněte pozici nehod ve vybraném kraji podle sloupců *d* a *e* (souřadnic v systému **S-JTSK**), které byly zaviněny zvěří (p10 == 4). Jednotlivé podgrafy budou odpovídat rokům 2023 a 2024. 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 zpracovaných funkcí `make_geo`. Graf uložte do souboru specifikovaného argumentem **`fig_location`** a případně zobrazte pokud **`show_figure`** je **`True`**. Pokud se nevytvoří správně obrázek ve fig_location (přímý název souboru, např. “geo1.png”), nemůže být úkol hodnocen.

Ukázka možného výstupu



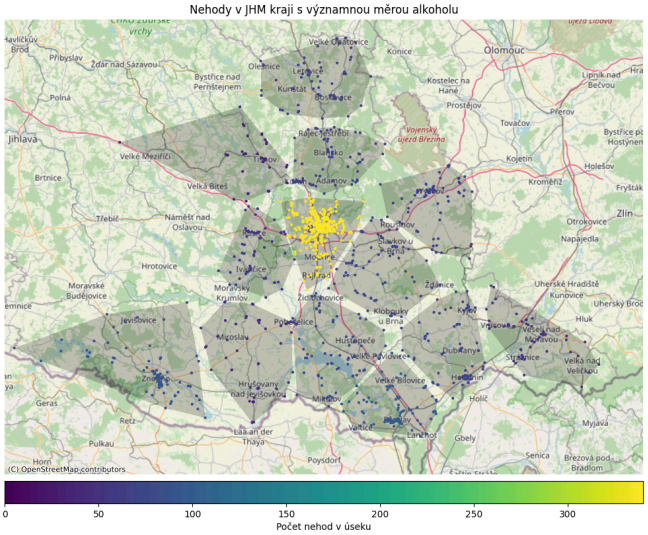
Četnost nehod
Signatura funkce

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

Funkcionalita
Vykreslete všechny nehody ve vybraném kraji tak, že znázorníte oblasti s významnou měrou alkoholu (p11>=4). Vytvořte vhodný počet oblastí, které budou podbarveny četností nehod v dané oblasti. Tímto způsobem bude možné najít kritická místa, kde vzniká nejvíce nehod. Oblasti tvořte pomocí vhodného shlukovacího algoritmu. Zobrazte také grafický ukazatel mapující barvu na počet nehod v úseku. Clustery podbarvěte barvou (s nastaveným alpha kanálem, pomůže vám convex_hull oblast). Pokud nějaká nehoda má špatně určený region, tak to neřešte, jen správně nastavte rozsah mapy (xlim a ylim). Znázorněte oblasti jako polygony a nehody jako body.

Vyberte vhodnou shlukovací metodu a počet clusterů tak, aby byla zachována přehlednost a vypovídající schopnost grafu. Do komentáře do kódu tuto volbu zdůvodněte (napište, jaké metody jste zkoušeli a podobně). 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

```python
if __name__ == "__main__":
```

- šablona tohoto souboru je v dokumentovém skladu ISu
- není třeba tvořit totožné grafy jako v ukázkách, vlastní invence se cení.
- vyhodnocení bude probíhat stejně, jako je definováno v části main.

Úkol 2: Test hypotézy (až 10 bodů)

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

Hypotéza 1:

Na silnicích první třídy se při nehodách umíralo se stejnou pravděpodobností jako na silnicích třetí třídy.

K ověření hypotézy využijte χ^2 test s tím, že také určíte, jestli nehody na silnicích 1. třídy vedly častěji či méně často k nehodě s následky na zdraví (sloupec *p9*) než na dálnicích (*p36)*. V tom vám může pomoct “expected” výstup χ^2 testu.

Hypotéza 2:

Škoda při nehodách značky X je nižší, než při nehodách značky Y a tato odchylka je statisticky významná.

Využijte vhodný test, který vyberete na základě distribuce dat, nevyužívejte χ^2 test. Určete, zda je škoda na vozidlech vyšší či nižší a zda je tato vlastnost na požadované hladině významnosti. Vyberte dvě různé kombinace značek, najděte případ, kde bude vyšší a najděte případ, kde to nemůžete určit.

Předpokládejte, že soubory `accidents.pkl.gz` a `vehicles.pkl.gz` jsou 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 10 vět,
- v textu minimálně další 3 vypočtené hodnoty (ne přímo opsané hodnoty z tabulky),

Místo textového dokumentu můžete vytvořit interaktivní HTML stránku doc.html (+ potřebné soubory), které splňují výše uvedené požadavky. HTML verzi volte jen v případě použití interaktivní grafiky (plotly a podobně). V ostatních případech odevzdejte PDF verzi.

Dále odevzdejte soubor `doc.py`, který bude pracovat se soubory `pkl.gz` (definované výše) 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). Ve vhodných případech preferujte vektorové formáty.
- 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/html` 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 zmíněné na přednáškách. Můžete použít libovolné knihovny představené na přednáškách, další pouze po schválení. **Nyní nejste omezeni, zda musíte používat Seaborn, Matplotlib a podobně.** Volba knihoven a nástrojů je jen na vás.

Výsledky budou vyhodnoceny na počítači se systémem Ubuntu, v prostředí Python 3.13 a s knihovnami v posledních stabilních verzích dle PIP.

Dotazy a připomínky

Dotazy a připomínky směřujte na mail mrazek@fit.vutbr.cz.