Vysoké učení technické v Brně
Fakulta informačních technologií

UPA – COVID-19

Filip Bali (xbalif00)
Natália Holková (xholko02)
Roland Žitný (xzitny01)

16. decembra 2021

Obsah

1 Prvá časť: návrh spracovania a uloženia dát

1.1 Vybrané dotazy

V prvej časti projektu sme si vybrali tieto dotazy:

1.1.1 Dotazy skupiny A

A1

Vytvorte čiarový (spojnicový) graf zobrazujúcí vývoj covidovej situácie po mesiacoch pomocou nasledujúcích hodnôt: počet novo nakazených za mesiac, počet novo vyliečených za mesiac, počet novo hospitalizovaných osôb za mesiac, počet vykonaných testov za mesiac. Pokiaľ nebude výsledný graf dobre čitateľný, zvážte logaritmickú mierku alebo rozdeľte hodnoty do viacerých grafov.
Pre bod **A1** je nutné získať počet *nakazených, vyliečených* a *hospitalizovaných* osôb za mesiac a počet vykonaných *testov* za mesiac.

A3

Vytvorte seriu stĺpcových grafov, ktoré zobrazia:

1. graf: počty vykonaných očkovaní v jednotlivých krajoch (celkový počet od začiatku očkovania).
2. graf: počty vykonaných očkovaní ako v predchádzajúcom bode naviac rozdelené podľa pohlavia. Diagram môže mať napr. dve časti pre jednotlivé pohlavia.
3. graf: počty vykonaných očkovaní, ešte ďalej rozdelené podľa vekovej skupiny. Pre potreby tohto diagramu postačia 3 vekové skupiny (0-24 rokov, 25-59, nad 59).

V bode **A3** sa musíme oboznámiť s jednotlivými *krajmi*, ich priradenými *kódmi* a následne s počtami *očkovaní* v jednotlivých krajoch a ich závislosťami na *pohlavie* a *vekové skupiny*.

1.1.2 Dotaz skupiny B

B1

Zostavte 4 rebríčky krajov "best in covid" za posledné 4 štvrťroky (1 štvrťrok = 1 rebríček). Ako kritérium voľte počet novo nakazených prepočítaný na jedného obyvateľa kraja. Pre jeden štvrťrok zobrazte výsledky tiež graficky. Graf bude pre každý kraj zobrazovať celkový počet novo nakazených, celkový počet obyvateľov a počet nakazených na jedného obyvateľa. Graf môžete zhotoviť kombinácie dvou grafov do jedného (jeden stĺpcový graf zobrazí prvé dve hodnoty a druhý, čiarový graf, hodnotu tretiu).
V bode **B1** sa zameriame znova na *kraje*, počet ich *obyvateľov* a následne počty *nakazených* v prepočte na jedného obyvateľa daných krajov.

1.1.3 Dotaz skupiny C

C1

Hľadanie skupín podobných miest z hľadiska vývoja covidu a vekového rozloženia obyvateľov.
Atribúty: počet nakazených za posledné 4 štvrťroky, počet očkovaných za posledné 4 štvrťroky, počet obyvateľov vo vekové skupine 0..14 rokov, počet obyvateľov vo vekovej skupine 15 - 59, počet obyvateľov nad 59 rokov. Pre potreby projektu vyberte ľubovoľných 50 miest, pre ktoré nájdete potrebné hodnoty (môžete napr. využiť nejaký rebríček 50 najľudňatejších miest v ČR).
Vo finálnom dotaze **C1** je nutné získať data o jednotlivých *mestách* a *veku* ich obyvateľov, kedy budeme sledovať počet *nakazených* a *očkovaných*.

1.1.4 Vlastné dotazy

Prvý vlastný dotaz

Vytvorte stĺpcový graf zobrazujúci vývoj covideovej sitúacie na základe počtov úmrtí v jednotlivých krajoch.
Pre tento dotaz je nutné získať počet *úmrtí* na kraje.

Druhý vlastný dotaz

Vytvorte stĺpcový graf popisujúci štatistiku o nasledujúcich dvoch hodnotách:

- Počet očkovaných osôb v jednotlivých krajoch.
- Počet úmrtí v jednotlivých krajoch.

Pre tento dotaz je nutné sledovať počty *očkovaní* a *úmrtí* v jednotlivých krajoch.

1.1.5 Zdroje potrebných dát

- **Nakazení**: dátum | vek | pohlavie | kraj_kód | okres_kód | zahraničie | zem_zahraničie (link)
- **Vyliečení**: dátum | vek | pohlavie | kraj_kód | okres_kód (link)
- **Hospitalizovaní**: dátum | prvý_záznam | kum_prvý_záznam | počet | bez priznakov | ... (link)
- **Testy**: dátum | pocet_PCR | pocet_AG | typológie | pozitívny (link)
- **Očkovanie(regiony)**: dátum | vakcína | kraj_kód | kraj_názov | veková_skupina | typy_dávok | celkom_dávok (link)
- **Očkovanie(ľudia)**: dátum | vakcína | kód | poradie | veková_skupina | pohlavie | pocet_dávok (link)
- **Očkovanie(geograficky)**: dátum | názov_vakcíny | kód_vakcíny | poradie_dávky | kraj_kod | cznuts | okres_kod | orp_kod | počet_dávok (link)
- **Mŕtvi**: dátum | vek | pohlavie | kraj_kód | okres_kód (link)
- **Prehľad vykázaných očkovaní pod profesií**: dátum | vakcína | kraj_nuts_kod | kraj_nazev | zarizeni_kod | zarizeni_nazev | poradi_davky | vekova_skupina | orp_bydliste | orp_bydliste_kod | pohlavi | vakcina_kod | ukoncujici_davka (link)
- **Obyvateľstvo podla päť ročných vekových skupín a pohlavia v krajoch a okresoch**: idhod | hodnota | stapro_kód | pohlavia_číselník | pohlavie_kód | vek_ciselnik | vek_kód | oblasť_číselník | oblasť_kód | dátum | pohlavie_text | vek_text (link)
- **Číselník krajov**: hodnota | text | cznuts | ruian | kraj_skratka (link)
- **Číselník okresov**: hodnota | text | cznuts | ruian | okres_lau | okres_skratka (link)

1.2 Schéma dát a ich spracovanie

Nakazení: výsledné data ohľadne nakazených osob bolo nutné očistiť od nepotrebných informácií a previesť pohlavie podľa slovníka.

date	age	gender	region	district

Vyliečení: tieto data zostali v pôvodnej podobe s prevedením pohlavia podľa slovníka.

date	age	gender	region	district

Hospitalizovaní: v dátach o hospitalizovaných nám postačujú iba informácie o dátume a počte hospitalizovaných. Ostatné boli odstránené a hodnota dátumu upravená pre naše potreby. Počty hospitalizovaných osôb sú zoskupené podľa mesiacov.

month	patients

Testy: pri počte vykonaných testov sa ponechali iba dátum a počet PCR a AG testov, ktoré sa následne zoskupili do jednej hodnoty vyjadrujúcej počet celkových testov za jednotlivé mesiace.

month	tests

Očkovanie(regióny): pre získanie počtu očkovaní v jednotlivých krajoch sme museli vyčitiť data od nepotrebných informácií a následne ich zoskupiť podla krajov.

region	count

date	age_group	gender

Očkovanie(ľudia): pre pozorovanie počtu očkovaných ľudí vzhľadom na pohlavie a vekovú skupinu, bolo nutné získať data zo spomenutého druhého zdroja a jemne očistiť data.
Očkovanie(geograficky): data o úmrtiach budú ďalej vhodné pri odpovedaní na vlastné dotazy.

date	age	gender	region	district

Mŕtvi: data o úmrtiach budú ďalej vhodné pri odpovedaní na vlastné dotazy.

date	age	gender	region	district

Obyvateľstvo podla päť ročných vekových skupín a pohlavia v krajoch a okresoch: data o úmrtiach budú ďalej vhodné pri odpovedaní na vlastné dotazy.

date	age	gender	region	district

Číselník krajov: data o úmrtiach budú ďalej vhodné pri odpovedaní na vlastné dotazy.

date	age	gender	region	district

Číselník okresov: data o úmrtiach budú ďalej vhodné pri odpovedaní na vlastné dotazy.

date	age	gender	region	district

1.3 Indexovanie

2 Druhá časť: implementovaný systém pre získavanie, ukladanie a spracovanie

Pre prehľadnosť boli jednotlivé dotazy implementované vo vlastných súboroch, ktoré sa nachádzajú v *Python* balíčku `queries`. V prípade potreby je tieto dotazy môžné volať samostatne, všetky naraz pomocou súboru `vizualizer.py`, alebo interaktívne zobrazovať cez grafické užívateľské rozhranie. Bližšie informácie k spusteniu sa nachádzajú v sekcii 3.4.
Pre získavanie dát z NoSQL databáze *MongoDB* sa používa knižnica `pymongo`. Na extrakciu dát do .csv súboru, jeho opätovné načítanie a úpravu pred vytváraním grafov slúži knižnica `pandas`. Zobrazovanie grafov sa rieši cez knižnicu `seaborn`.

2.1 Riešenie dotazu A1

Extrakcia dát z NoSQL databáze

Extrakcia dát z *MongoDB* prebieha vo funkcii `A1_extract_csv`, ktorá sa nachádza v súbore `queriesA1.py`.
aggregate infected - group by month and year and count total number in that month Najskôr agregujeme počty nakazených, ktorý sa nachádzajú v tabuľke *infected* (tabuľka 1.2), podľa rokov a mesiacov a výsledok prevedieme do `Pandas.DataFrame`. Rovnako postupujeme pri dátach o vyliečených, ktoré sú dostupné v tabuľke 1.2. Dáta o hospitalizovaných a počtoch vykonaných testov nie je potrebné agregovať, stačí ich iba previesť do `Pandas.DataFrame`. Jednotlivé `DataFrame` sú zlúčené podľa mesiacov a výsledok je zapísaný do .csv súboru.

Štruktúra .csv súboru

Súbor vytvorený funkciou `A1_extract_csv` má štruktúru:

- *month* - mesiac v tvare `YYYY-MM`
- *infected* - počet nakazených za daný mesiac
- *cured* - počet vyliečených za daný mesiac
- *hospitalized* - počet hospitalizovaných za daný mesiac
- *tests* - počet vykonaných testov za daný mesiac

Grafické zobrazenie výsledkov z .csv súboru

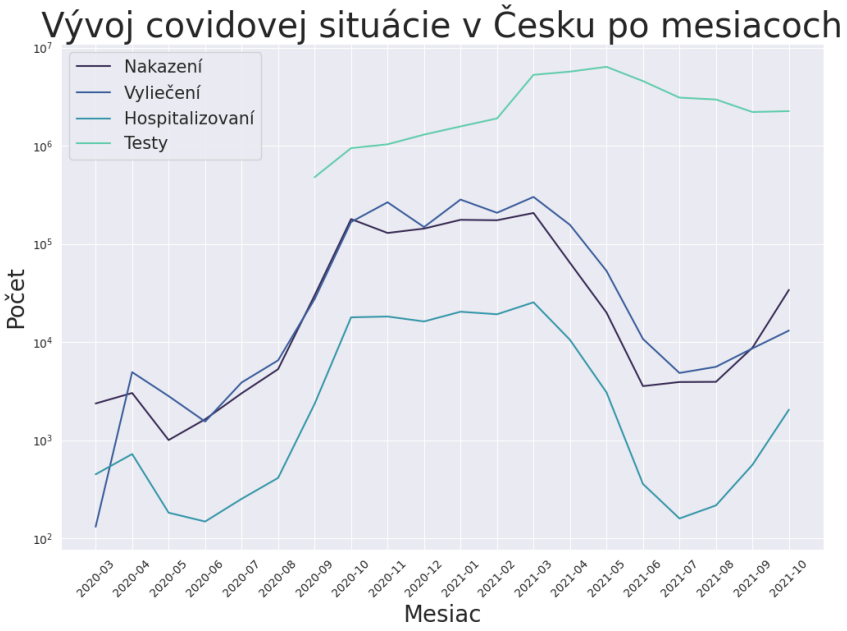
Vytvorenie grafu z extrahovaných dát v .csv súbore prebieha vo funkcii `A1_plot_graph`, ktorá sa nachádza v súbore `queries/A1.py`. Bol vytvorený spojnicový graf, ktorý zobrazuje vývoj počtov nakazených, vyliečených, hospitalizovaných a vykonaných testov od začiatku pandémie. Počty sú agregované po mesiacoch. Údaje o vykonaných testoch začali byť vedené až neskoršie, čo možno vidieť na grafe. Pre prehľadnosť je použitá logaritmická mierka na osu y. Z grafu možno jasne vidieť nástup jednotlivých vĺn.
Výsledný graf je možné vidieť na obrázku 2.1.

2.2 Riešenie dotazu A3

Extrakcia dát z NoSQL databáze

Extrakcia dát z *MongoDB* prebieha vo funkcii `A3_extract_csv`, ktorá sa nachádza v súbore `queriesA3.py`.
Proces agregácie je zreťazené spracovanie dokumentu obsahujúci očkovaných podľa profesií. Agregácia je rozdelená do nasledujúcich etáp:

- *match* - Z dokumentu vyberieme len záznamy, ktorých hodnota v atríbúte *gender* je hodnota *M* alebo *Z*
- project - Následne vyberieme atribúty dokumentu *region*, *gender*, *age_group*
- group - Zoskupíme záznamy dokumentu pod *_id*, pričom je zložené z atribútov v slede: *region*, *gender*, *age_group*. Ich následný počet je sčítaný pod atribút *count*.
- sort - Záznamy usporiadame podľa atribútu *region*.



Obr. 1: Dotaz A1 - graf

Štruktúra .csv súboru

Súbor vytvorený funkciou `A3_extract_csv` má štruktúru:

- *region* - skratka kraja
- *count* - počet zaočkovaných ľudí kraja so zvoleným pohlavím a vekovou skupinou
- *gender* - pohlavie (M - muži, Ž - ženy)
- *age_group* - vekové skupiny delené po 5 ročných intervaloch

Grafické zobrazenie výsledkov z .csv súboru

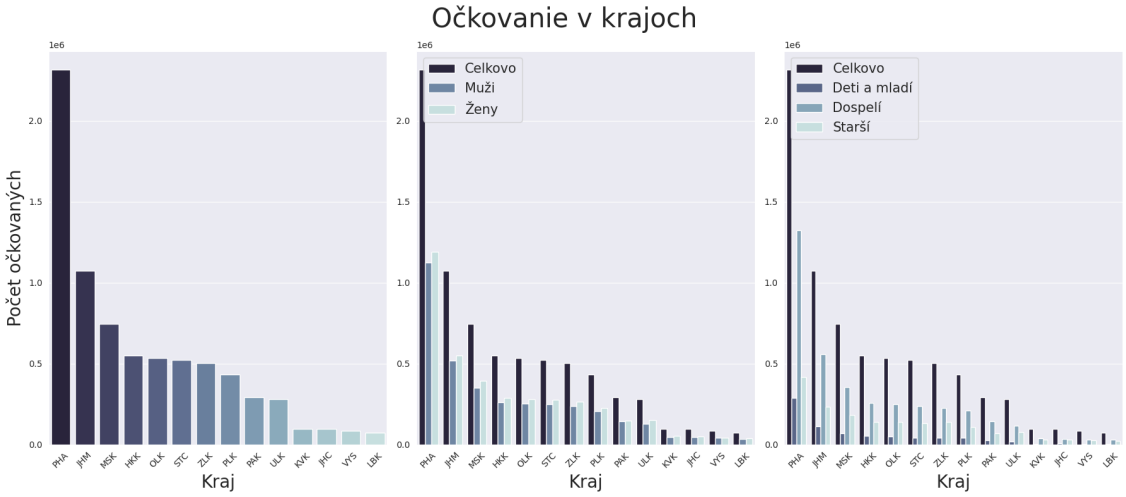
Vytvorenie grafu z extrahovaných dát v .csv súbore prebieha vo funkcii `A3_plot_graph`, ktorá sa nachádza v súbore `queries/A3.py`.
Boli vytvorené celkovo tri stĺpcové grafy. Prvý graf zobrazuje celkový počet doterajších zaočkovaných na základe krajov. Druhý graf je podobný prvému, ale očkovaní sú tu ďalej rozdelení poďľa pohlavia. Posledný graf ukazuze rozdelenie podľa vekovej skupiny, ktorou môžu byť *deti a mladí* (0-24), *dospelí* (25-59) a starší (60+). Kraje vo všetkých grafoch sú zoradené zostupne podľa počtu zaočkovaných.
Z grafov možno vidieť, že najviac zaočkovaných je v kraji *Hlavní město Praha*.
Výsledný graf je možné vidieť na obrázku 2.2.

2.3 Riešenie dotazu B1

Extrakcia dát z NoSQL databáze

Extrakcia dát z *MongoDB* prebieha vo funkcii `B1_extract_csv`, ktorá sa nachádza v súbore `queriesB1.py`.
Úloha vyžaduje agregáciu z viacerých dokumentov (infikovaný, demografické dáta, číselník krajov).
Dokument obsahujúci dáta infikovaných je v agregácií spracovaný v nasledujúcich zreťazených etápách:

- match - Z dokumentu vyberieme len záznamy, ktorých rok v atribúte *date* je 2021.
- project - Následne vyberieme atribúty dokumentu *date*, *region*, *quarter*, kde *quarter* je atribút odvodený od atribútu *date* a to tak, že špecifikuje v ktorom kvartáli roka 2021 sa záznam nachádza.
- group - Zoskupíme záznamy dokumentu pod *_id*, pričom je zložené z atribútov v slede: *region*, *quarter*. Ich následný počet je sčítaný pod atribút *count*.



Obr. 2: Dotaz A3 - graf

- sort - Záznamy usporiadame podľa atribútov v slede: *region*, *quarter*.

Dokument demografických dát je spracovaný v jedinej etape *project*, ktorá vyberá atribúty *value*, *territory_code*, *territory_txt*, *valid_date*, *gender_code*, *gender_txt*, *age_code*, *age_txt*.
Dokument obsahujúci číselník krajov je rovnako spracovaný v jedinej etape *project*, ktorá vyberá atribúty *cznuts*, *region_shortcut*, *value*.

Štruktúra .csv súboru

Funkcia `B1_extract_csv` vytvára 4 .csv súbory `B1_Q1.csv`, `B1_Q2.csv`, `B1_Q3.csv` a `B1_Q4.csv`, jeden pre každý štvrťrok. Každý súbor má nasledovnú štruktúru:

- *region_shortcut* - skratka kraja
- *quarter* - číslo štvrťroka v poradí
- *population* - celková populácia kraja
- *infected* - počet nakazených v kraji za daný štvrťrok
- *infected_normalized* - počet nakazaných normalizovaný vzhľadom na populáciu kraja

Grafické zobrazenie výsledkov z .csv súboru

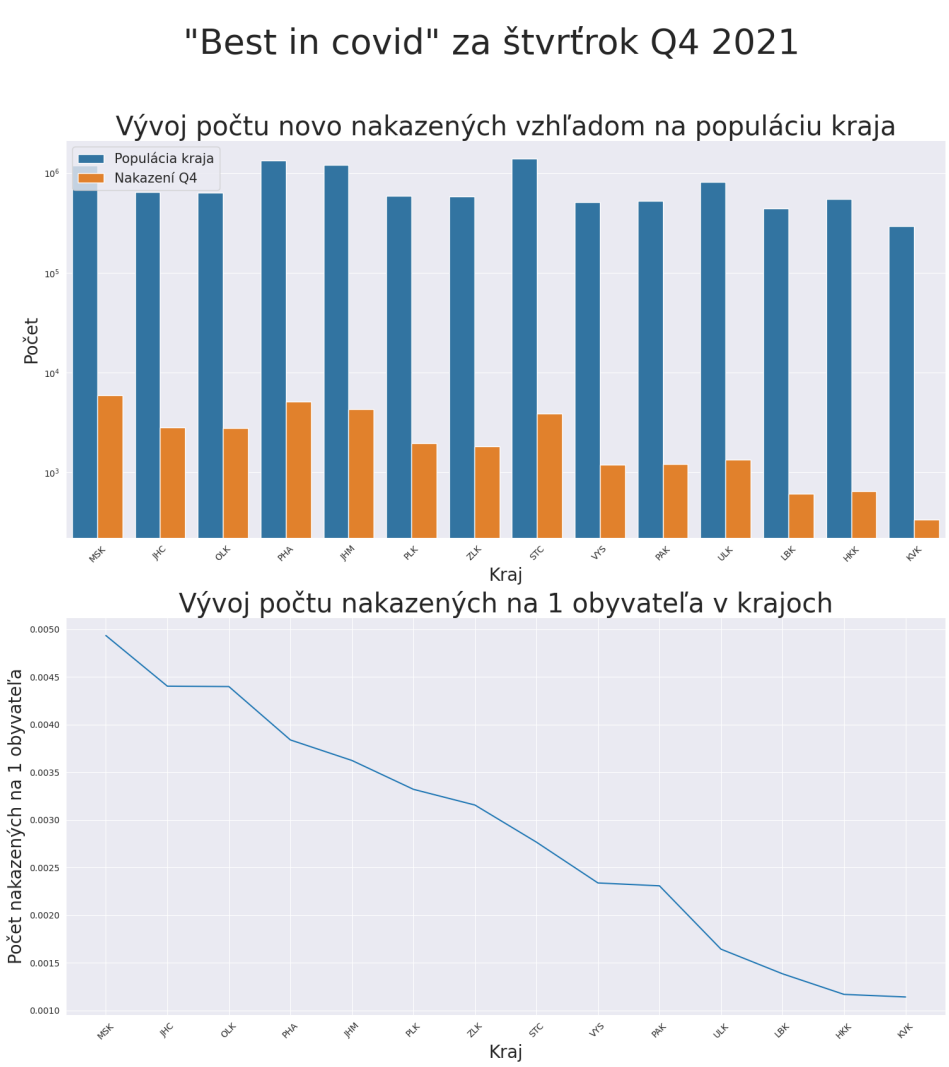
Vytvorenie grafu z extrahovaných dát v .csv súbore prebieha vo funkcii `B1_plot_graph`, ktorá sa nachádza v súbore `queries/B1.py`.
Predvolene sme zvolili, že budeme vykresľovať rebríčky *"best in covid"* krajov z posledného štvrťroka roku 2021. Horný stĺcový graf zobrazuje populáciu kraja a počet nakazených za daný štvrťrok. Dolný spojnicový graf predstavuje počet nakazených na 1 obyvateľa na základe kraju. Obe grafy majú zoradenú os x zostupne podľa tohto počtu nakazených na 1 obyvateľa.
Výsledný graf je možné vidieť na obrázku 2.3.

2.4 Riešenie dolovacej úlohy C1

Extrakcia dát z NoSQL databáze

Úloha vyžaduje agregáciu z viacerých dokumentov (infikovaný, demografické dáta, číselník okresov, geograficky prehľad vykázaných očkovaní).
Dokument obsahujúci dáta infikovaných je v agregácií spracovaný v nasledujúcich zreťazených etápách:

- match - Z dokumentu vyberieme len záznamy, ktorých rok v atribúte *date* je 2021.
- project - Následne vyberieme atribúty dokumentu *date*, *district*.



Obr. 3: Dotaz B1 - graf

- group - Zoskupíme záznamy dokumentu pod *_id*, ktorý obsahuje atribút *region*. Ich následný počet je sčítaný pod atribút *count*.

Dokument obsahujúci demografické dáta je v agregácií spracovaný v nasledujúcich zreťazených etápách:

- match - Z dokumentu vyberieme len záznamy, ktorých dátum v atribúte *valid_date* je 2020-12-31 (tento dátum predstavuje najnovšie dáta v datovej sade). Zároveň je prítomný atribút *territory_txt* z ktorého je vyberaný zoznam 50 miest. Zoznam miest je pevne daný.
- project - Následne vyberieme atribúty dokumentu *value*, *territory_txt*, *territory_code*, *age_txt*, *age_code*, *gender_txt*, *gender_code*.

Dokument obsahujúci geografický prehľad vykázaných očkovaní je v agregácií spracovaný v nasledujúcich zreťazených etápách:

- match - Z dokumentu vyberieme len záznamy, ktorých rok v atribúte *date* je 2021. Zároveň je prítomný operátor *or*, ktorý zaistuje, že vybrané záznamy budú pochádzať len od plne zaočkovaných osôb (Minimálne dve dávky u vakcín s kódom: *CO01*, *CO02, CO03*. Minimálne jedna dávka u vakcíny s kódom *CO04*.)
- project - Následne vyberieme atribúty dokumentu *date*, *district_name*, *orp*, *shot_count*.

- group - Zoskupíme záznamy dokumentu pod *_id*, ktorý je zložený z atribútu *district_name*. Následné sú naprieč všetkými záznamami v jednotlivých krajoch sčítané hodnoty z atribútu *shot_count* a výsledná suma každý kraj je pod atribútom *vaccinated_count*.

Dokument obsahujúci číselník okresov je spracovaný v jedinej etape *project*, ktorá vyberá atribúty *text*, *cznuts*, *okres_lau*, *region_shortcut*.

Štruktúra .csv súborov

Obe .csv súbory majú nasledovnú štruktúru:

- *district_name* - názov okresného mesta
- *infected_count* - počet nakazených v meste
- *vaccinated_count* - počet očkovaných v meste
- *population_count(age_0-14)* - veľkosť populácie vo vekovej skupine 0-14 rokov
- *population_count(age_15-59)* - veľkosť populácie vo vekovej skupine 15-59 rokov
- *population_count(age_59-above)* - veľkosť populácie vo vekovej skupine 60+ rokov

Normalizácia, diskretizácia a spracovanie odľahlých hodnôt

Zvolili sme si normalizovať stĺpec `population_count(age_0-14)`. Normalizáciu sme vykonali podelením pôvodného stĺpca celkovou populáciou mesta (teda súčet všetkých troch vekových skupín).
Rozhodli sme sa odstraňovať odľahlé hodnoty zo stĺpca `vaccinated_count` a nahradiť ich mediánom. Pri každej hodnote počítame, či |počet_očkovaných − medián| > štandartná odchýlka, v tom prípade nahrádzame mediánom.
Diskretizácia bola aplikovaná na vekové rozloženie v jednotlivých krajoch. Výsledkom je rozdelenie dát vekového rozloženia do troch skupín:

- Vek v intervale 0 až 14
- Vek v intervale 15 až 59
- Vek v intervale 60 a viac

2.5 Riešenie 1. vlastného dotazu

Extrakcia dát z NoSQL databáze

Extrakcia dát z *MongoDB* prebieha vo funkcii `VL1_extract_csv`, ktorá sa nachádza v súbore `queries/VL1.py`.
Úloha vyžaduje agregáciu z viacerých dokumentov (mrtvý, číselník krajov).
Dokument obsahujúci dáta mrtvých je v agregácií spracovaný v nasledujúcich zreťazených etápach:

- match - Z dokumentu vyberieme len záznamy, z atribútu *date* v ktorých je overený dátum a sú z desaťročia 2020.
- project - Následne vyberieme atribúty dokumentu *date*, *region*.
- group - Zoskupíme záznamy dokumentu pod *_id*, ktorý obsahuje atribút *region*. Následne sú pod atribút *dead_count* sčítané všetky záznamy patriace pod jednotlivé kraje.

Dokument obsahujúci číselník krajov je spracovaný v jedinej etape *project*, ktorá vyberá atribúty *region_shortcut*, *cznuts*.

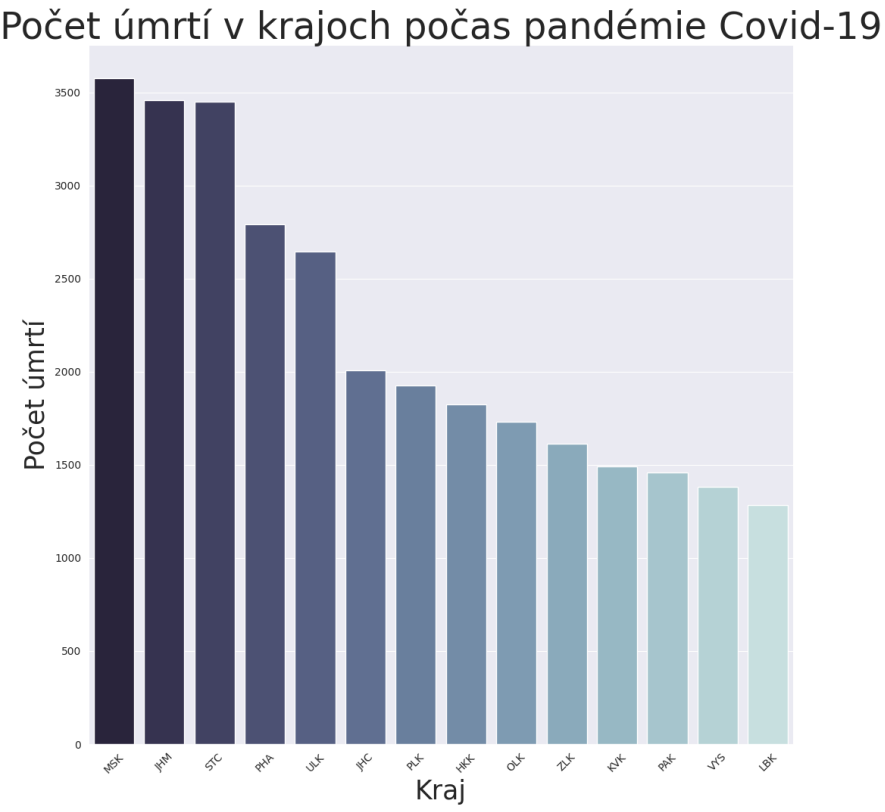
Štruktúra .csv súboru

Súbor vytvorený funkciou `VL1_extract_csv` má štruktúru:

- *region_shortcut* - skratka kraja
- *dead_count* - celkový počet mŕtvych v kraji

Grafické zobrazenie výsledkov z .csv súboru

Vytvorenie grafu z extrahovaných dát v .csv súbore prebieha vo funkcii `VL1_plot_graph`, ktorá sa nachádza v súbore `queries/VL1.py`.
V tomto dotaze graficky zobrazujeme rozloženie ľudí, ktorí umreli na covid-19 počas celej pandémie, v závislosti na kraj ich pôvodu. Je to riešené spojnicovým grafom, ktorý je zostupne zoradený.
Z grafu je možné vidieť, že k najviac úmrtiam došlo v *Moravskoslezskom kraji.*
Výsledný graf je možné vidieť na obrázku 2.5.



Obr. 4: Prvý vlastný dotaz - graf

2.6 Riešenie 2. vlastného dotazu

Extrakcia dát z NoSQL databáze

Extrakcia dát z *MongoDB* prebieha vo funkcii `VL2_extract_csv`, ktorá sa nachádza v súbore `queries/VL2.py`.
V tomto dotaze skúmame počty očkovaných a zomrelých v jednotlivých krajoch, preto využívame dáta z dvoch rôznych zdrojov.
Úloha vyžaduje agregáciu z viacerých dokumentov (úmrtia, demografické dáta, číselník krajov).
Dokument obsahujúci dáta mrtvých je v agregácií spracovaný v nasledujúcich zreťazených etápach:

- match - Z dokumentu vyberieme len záznamy, z atribútu *date* v ktorých je overený dátum a sú z desaťročia 2020.
- project - Následne vyberieme atribúty dokumentu *date*, *region.*
- group - Zoskupíme záznamy dokumentu pod *_id*, ktorý obsahuje atribút *region.* Následne sú pod atribút *dead_count* sčítané všetky záznamy patriace pod jednotlivé kraje.

Dokument obsahujúci demografické dáta je spracovaný v jedinej etape *project*, ktorá vyberá atribúty *value*, *territory_code*, *territory_txt*, *valid_date*, *gender_code*, *gender_txt*, *age_code*, *age_txt*.
Dokument obsahujúci číselník krajov je rovnako spracovaný v jedinej etape *project*, ktorá vyberá atribúty *text*, *cznuts*, *region_shortcut*.

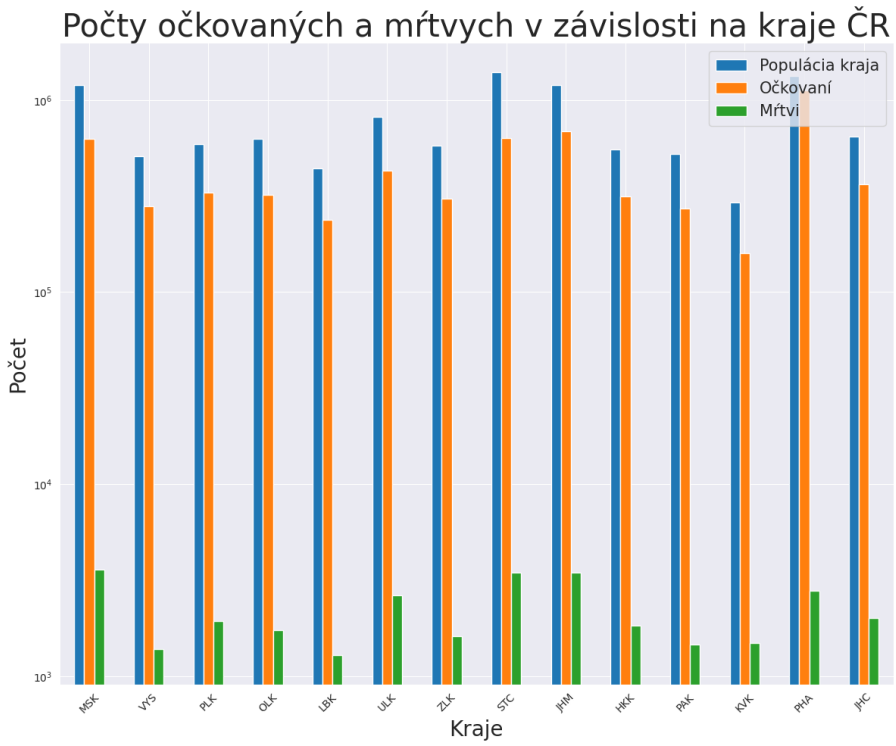
Štruktúra .csv súboru

Súbor vytvorený funkciou `VL2_extract_csv` má štruktúru:

- *region_shortcut* - skratka kraja
- *dead_count* - celkový počet mŕtvych v kraji
- *vaccinated_count* - celkový počet zaočkovaných v kraji
- *population* - celková populácia kraja

Grafické zobrazenie výsledkov z .csv súboru

Vytvorenie grafu z extrahovaných dát v .csv súbore prebieha vo funkcii `VL2_plot_graph`, ktorá sa nachádza v súbore `queries/VL2.py`.
V tomto dotaze zobrazujeme celkovú populáciu kraja, počet očkovaných a počet zomrelých v kraji. Stĺpcový grafy používa logaritmickú mierku pre os y hlavne z dôvodu veľkého rozdielu medzi populáciou kraja a počtom zomrelých.
Výsledný graf je možné vidieť na obrázku 2.6.

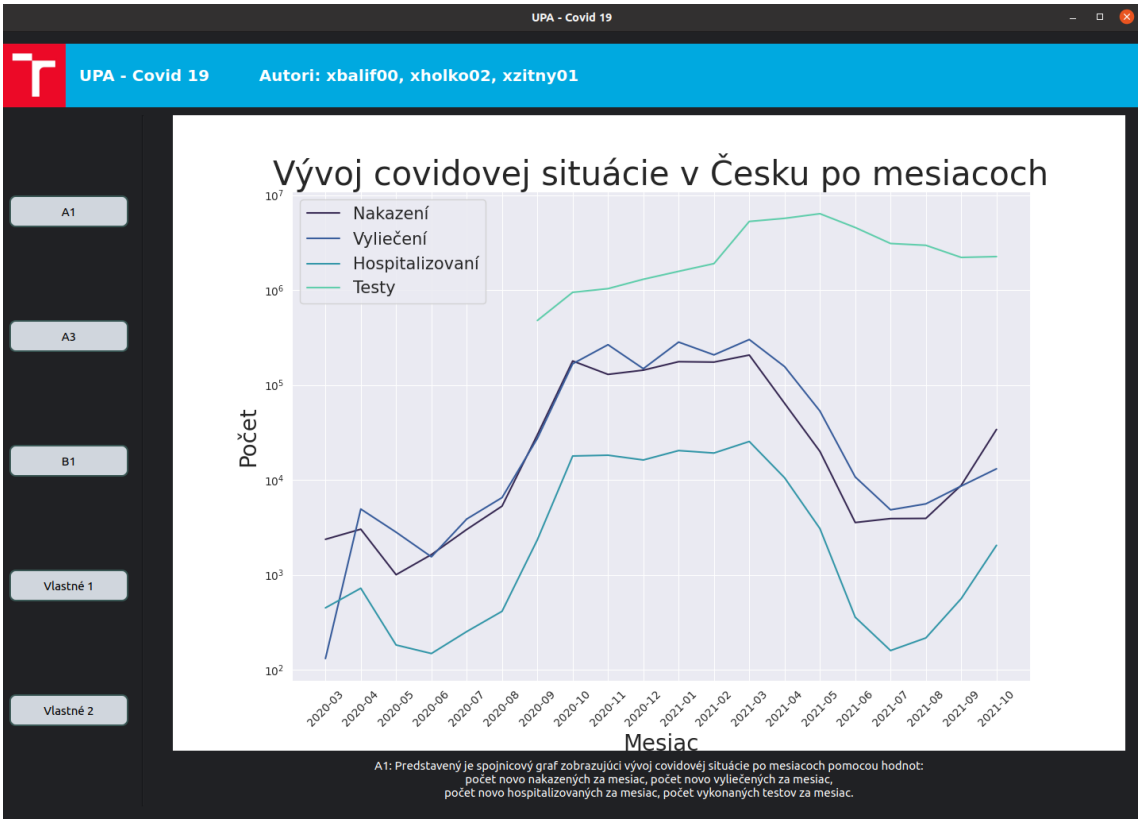


Obr. 5: Druhý vlastný dotaz - graf

2.7 Grafické užívateľské rozhranie

Grafické užívateľské rozhranie bolo vytvorené pomocou multiplatformnej knižnici *PyQt*[1] a slúži pre jednoduché zobrazenie grafou vytvorených z dotazov preberaných v sekciách 2.1, 2.2, 2.3, 2.5 a 2.6. Toto rozhranie pozostáva z piatich tlačidiel, ktoré prezentujú výstup dotazov spolu s jednoduchým popisom, čo je možné vidieť na obrázkoch 2.7 a 2.7.

[1] `https://wiki.python.org/moin/PyQt`



Obr. 6: GUI - dotaz A1

Obr. 7: GUI - dotaz A3

3 Vybrané technológie a spustenie

3.1 Technológie

- ***Python3.8***
- ***MongoDB*** - NoSQL DB
- ***Pandas*** - manipulácia a analýza dát
- ***Seaborn*** - tvorba grafov
- ***PyQt5*** - tvorba GUI

3.2 Inštalácia

Inštalácia MongoDB - `https://docs.mongodb.com/manual/administration/install-community/`

```
python3 -m venv ./env-upa
source ./env-upa/bin/activate
pip3 install -r requirements.txt
```

3.3 Spustenie 1. časti projektu

Príklad spustenia 1. časti projektu:

```
python3 data_loader.py --mongo mongodb://localhost:27017/ -f download_data_folder -d UPA-db
```

3.4 Spustenie 2. časti projektu

Pred spustením 2. časti projektu je **nutné** znovu spustiť 1. časť podľa 3.3 aby bola zaručená správna štruktúra databáze. Bolo pripravených viacero možností, ako spúšťať dotazy a získavať výsledky. Pri všetkých možnostiach je potrebné používať rovnakú databázu, ktorá bola vytvorená v predchádzajúcom kroku.

Vygenerovanie všetkých .csv a grafov naraz

Pripravený skript `vizualizer.py` postupne extrahuje z databáze .csv pre všetky dotazy a zároveň vytvorí príslušné grafy.

```
python3 vizualizer.py --mongo mongodb://localhost:27017/ -d UPA-db
```

Spúštanie úloh jednotlivo

V prípade nutnosti získať výsledky iba z jedinej úlohy, je možné toho docieliť pomocou príkazov:

```
python3 queries/A1.py --mongo mongodb://localhost:27017/ -d UPA-db    # pre dotaz A1
python3 queries/A3.py --mongo mongodb://localhost:27017/ -d UPA-db    # pre dotaz A3
python3 queries/B1.py --mongo mongodb://localhost:27017/ -d UPA-db    # pre dotaz B1
python3 queries/C1.py --mongo mongodb://localhost:27017/ -d UPA-db    # pre dotaz C1
python3 queries/VL1.py --mongo mongodb://localhost:27017/ -d UPA-db   # pre dotaz VL1
python3 queries/VL2.py --mongo mongodb://localhost:27017/ -d UPA-db   # pre dotaz VL2
```

Spúštanie cez grafické užívateľské rozhranie

Grafy z dotazov skupiny A, B a vlastných dotazov je navyše možné zobrazovať interaktívne cez grafické užívateľské rozhranie popísané v sekcii 2.7. Pred spustením je **nutné** spustiť vygenerovanie grafov cez `vizualizer.py`. Spustiť GUI je možné nasledovne:

```
python3 GUI/main.py
```