

Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra geoinformatiky

**PUBLIKACE PROSTOROVĚ-STATISTICKÝCH
DAT V PROSTŘEDÍ SHINY**

Bakalářská práce

Tomáš POTOČIAR

Vedoucí práce Mgr. Karel Macků, Ph.D.

Olomouc 2022
Geoinformatika a geografie

ANOTACE

Bakalářská práce se zabývá možnostmi vizualizace pomocí knihovny Shiny umožňující vytvářet v prostředí R interaktivní webové aplikace. V rámci práce je vytvořena webová aplikace nesoucí název ShinyVision zaměřená na prostorové, statistické a tabelární vizualizace publikovaná formou webové platformy shinyapps.io. Aplikace umožňuje uživatelům pracovat s prostorovými daty v rámci metody bodových, liniových a plošných znaků či pomocí pokročilých metod kartogramu nebo kartodiagramu. Pro statistické data byly vybrány vizualizace histogram, boxplot, korelační diagram, korelační matice, paralelní osy a liniový graf. Tabelární vizualizace slouží pro prohlížení zdrojových dat ve formě tabulky. Uživatel má možnost pracovat s ukázkovými datovými sady nebo si nahrát data vlastní. Aplikace Shiny se skládá z uživatelského rozhraní a serverové části. Tvorba aplikace je založena na reaktivním programování umožňující okamžitou aktualizaci výstupů při změně uživatelských vstupů. Na závěr práce jsou stanovena kritéria vyhodnocující limitaci tohoto nástroje – z hlediska možnosti vizualizace, omezení datových formátů, maximální velikosti dat nebo z hlediska licenčních omezení.

KLÍČOVÁ SLOVA

Shiny; prostředí R; aplikace; vizualizace

Počet stran práce: 45

Počet příloh: 2 (z toho 2 volné)

ANOTATION

The bachelor thesis deals with the possibilities of visualization using the Shiny library that creates interactive web applications in the R environment. Within the thesis, a web application called ShinyVision is created, focusing on spatial, statistical and tabular visualizations. The app is published on a web platform shinyapps.io. The application allows users to work with spatial data within point, line and area feature methods or using advanced choropleth maps or cartodiagram methods. The visualizations selected for statistical data were histogram, boxplot, correlation chart, correlation matrix, parallel axes and line chart. The tabular visualization is used to view the source data in table form. The user has the option to work with sample datasets or upload their own data. The Shiny application consists of a user interface and a server part. The creation of the application is based on reactive programming allowing immediate updating of outputs when user inputs change. At the end of the work, criteria are established to evaluate the limitations of the tool - in terms of visualization capabilities, data format limitations, maximum data size or licensing restrictions.

KEYWORDS

Shiny; R environment; application; visualization

Number of pages 45

Number of appendixes: 2

Prohlašuji, že

- bakalářskou práci včetně příloh jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.
- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,
- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užívat (§ 35 odst. 3),
- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,
- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,
- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,
- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne

Tomáš POTOČIAR

Děkuji vedoucímu práce Mgr. Karlovi Macků, Ph.D. za podněty, připomínky a poskytnutí testovacích dat. Také děkuji členům týmu Rie-G za užitečnou zpětnou vazbu při tvorbě práce.

UNIVERZITA PALACKÉHO V OLMOUCI

Přírodovědecká fakulta
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Tomáš POTOČIAR
Osobní číslo: R190351
Studijní program: B1301 Geografie
Studijní obor: Geoinformatika a geografie
Téma práce: Publikace prostorově-statistických dat v prostředí Shiny
Zadávající katedra: Katedra geoinformatiky

Zásady pro vypracování

Cílem práce je demonstrovat možnosti publikace prostorových a statistických dat pomocí knihovny Shiny, která umožňuje v prostředí R vytvářet interaktivní webové aplikace. Student sestaví ukázkovou datovou sadu, která bude vhodná pro publikaci – bude obsahovat jak prostorovou, tak bohatou atributovou složku. Následně vybere atraktivní metody vizualizace relevantní pro exploratorní analýzu statistických dat. Tyto metody vizualizace budou prostorové (kartogram, kartodiagram atd.), neprostorové (histogram, boxplot, paralelní osy atd.) a tabelární (pro prohlížení zdrojových dat). Všechny vybrané metody budou následně vhodně zorganizovány do webové aplikace sestavené v prostředí Shiny. Závěrem student stanoví vhodná kritéria, která umožní vyhodnotit limitace tohoto nástroje – z hlediska zpracovatelných datových formátů, limitace velikostí dat, možností vizualizace nebo z hlediska licenčních omezení.

Student vyplní údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, do Metainformačního systému katedry geoinformatiky a současně provede zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) bude odevzdána v digitální podobě na CD nebo DVD a text práce bude spolu s vybranými přílohami odevzdán ve dvou svazcích na sekretariát katedry. O práci student vytvoří webové stránky, které budou v souladu s pravidly dostupnými na stránkách katedry. Diplomová práce bude zpracována podle zásad dle Voženilka (2002) a také podle šablon dostupných na stránkách katedry. Na závěr práce připojí student jednostránkové resumé v anglickém jazyce. Jako shrnutí bakalářské práce bude vytvořen poster.

Rozsah pracovní zprávy: max. 50 stran
Rozsah grafických prací: dle potřeby
Forma zpracování bakalářské práce: tištěná

Seznam doporučené literatury:

1. SIEVERT, C. (2020). Interactive Web-Based Data Visualization with R, plotly, and shiny (1st edition). <https://doi.org/10.1201/9780429447273>
2. WICKHAM, H. (2021). Mastering Shiny: Build Interactive Apps, Reports, and Dashboards Powered by R. O'Reilly Media, Inc.
3. YAU, N. (2013). Data Points – Visualization That Means Something. Indianapolis: John Wiley & Sons, Inc.
4. YAU, N. (2011). Visualize This: The FlowingData Guide to Design, Visualization, and Statistics. Wiley Publishing, Inc.
5. VOŽENÍLEK, V., (2002). Diplomové práce z geoinformatiky. Olomouc: Vydavatelství Univerzity Palackého. ISBN 80-244-0469-9.

Vedoucí bakalářské práce: Mgr. Karel Macků, Ph.D.
Katedra geoinformatiky

Datum zadání bakalářské práce: 4. května 2021
Termín odevzdání bakalářské práce: 4. května 2022

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA GEOINFORMATIKY
17. listopadu 50, 771 46 Olomouc
-1-

L.S.



doc. RNDr. Martin Kubala, Ph.D.
děkan

prof. RNDr. Vít Voženílek, CSc.
vedoucí katedry

OBSAH

SEZNAM POUŽITÝCH ZKRATEK	10
ÚVOD	11
1 CÍLE PRÁCE.....	12
2 METODY A POSTUPY ZPRACOVÁNÍ.....	13
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	15
3.1 Vizualizace dat.....	15
3.1.1 Metody prostorové vizualizace.....	15
3.1.2 Metody statistické vizualizace	16
3.2 Webové aplikace	18
3.3 Jazyk R	18
3.4 RStudio	18
3.5 Shiny	19
3.5.1 Reaktivní programování.....	19
3.5.2 Možnosti publikace	19
3.5.3 Alternativy Shiny.....	20
4 APLIKACE SHINYVISION	22
4.1 Implementace Shiny do RStudio.....	22
4.2 Layout aplikace	22
4.3 Uživatelské rozhraní	24
4.4 Výběr dat.....	25
4.5 Prostorové vizualizace	27
4.5.1 Metoda kartogramu.....	27
4.5.2 Metoda kartodiagramu	30
4.6 Statistické vizualizace	31
4.6.1 Histogram	32
4.6.2 Boxplot	32
4.6.3 Korelační diagram	33
4.6.4 Korelační matice	34
4.6.5 Paralelní osy	35
4.6.6 Liniový graf.....	36
4.7 Tabelární vizualizace.....	37
4.8 Prohlížení prostorových dat	38
5 LIMITY A OMEZENÍ.....	39
5.1 Licenční omezení shinyapps.io	39
5.2 Velikost importovaných dat	40
5.3 Datové formáty	40
5.4 Struktura importovaných dat.....	41
5.5 Možnosti vizualizace	41
5.6 Interaktivita.....	41

6	VÝSLEDKY	42
7	DISKUZE	43
8	ZÁVĚR	45
	POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
	PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
API	Application Programming Interface
AWS	Amazon Web Services
BI	Business Intelligence
CRAN	The Comprehensive R Archive Network
CSS	Cascading Style Sheets
CSV	Comma Separated Values
ČSÚ	Český statistický úřad
HEX	Hexadecimal
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
NUTS	Nomenklatura územních statistických jednotek
PDF	Portable Document Format
PNG	Portable Network Graphics
SHP	Shapefile
UI	User Interface
WGS84	World Geodetic System 1984

ÚVOD

Webové aplikace jsou nedílnou součástí moderní doby informačních technologií. Výhodou webové aplikace oproti aplikaci na lokálním zařízení je jednoznačně dostupnost. Kdokoliv pouze s přístupem k internetu si může aplikaci zobrazit a nepotřebuje instalovat doplňující software a knihovny. Jednou z možností, jak jednoduše vytvářet a veřejně sdílet webové aplikace, je prostředí R s přídatným balíčkem Shiny. I když jsou pomocí tohoto balíčku vytvářeny interaktivní komplexní webové aplikace, není podmínkou znalost jazyků pro tvorbu webu (HTML, CSS, JavaScript), stačí znát možnosti programovacího jazyka R.

V dnešní době existuje velké množství dat, které nikdy nebyly vizualizovány. A právě vizualizace dat dokáže zviditelnit informace, jenž by v případě písmen a čísel v tabulce nebyly nikdy nalezeny. Jazyk R byl vyvinut primárně pro zpracování dat a jejich vizualizaci. V kombinaci s vytvářením interaktivních aplikací se jedná o opravdu silný nástroj pro poskytování důležitých informací uživatelům.

Bakalářská práce se zabývá možnostmi vizualizace prostorových a statistických dat v prostředí R pomocí knihovny Shiny. Řeší problematiku vizualizací dat, vysvětluje princip knihovny Shiny a předkládá možnosti publikace vytvořených aplikací. Zaměřuje se na tvorbu webové aplikace od vzniku uživatelského rozhraní, přes pochopení unikátního programovacího paradigmatu až po samotnou publikaci hotové aplikace.

1 CÍLE PRÁCE

Cílem bakalářské práce je demonstrovat možnosti publikace prostorových a statistických dat pomocí knihovny Shiny, která umožňuje v prostředí R vytvářet interaktivní webové aplikace. Student sestaví ukázkovou datovou sadu, která bude vhodná pro publikaci – bude obsahovat jak prostorovou, tak bohatou atributovou složku. Následně vybere atraktivní metody vizualizace relevantní pro exploratorní analýzu statistických dat. Tyto metody vizualizace budou prostorové (kartogram, kartodiagram, prohlížení prostorových dat), neprostorové (histogram, boxplot, korelační diagram, korelační matice, paralelní osy, liniový graf), tabelární (pro prohlížení zdrojových dat). Všechny vybrané metody budou následně vhodně zorganizovány do webové aplikace sestavené v prostřední Shiny. Závěrem student stanoví vhodná kritéria, která umožní vyhodnotit limitace tohoto nástroje – z hlediska zpracovatelných datových formátů, limitace velikostí dat, možností vizualizace nebo z hlediska licenčních omezení.

Vytvořená aplikace umožní prohlížení dat atraktivními metodami vizualizace. Aplikace najde uplatnění při detailním rozboru datové sady, usnadní vytváření grafů a map ve webovém prostředí a může sloužit jako studijní podklad pro tvorbu dalších aplikací v prostředí Shiny.

2 METODY A POSTUPY ZPRACOVÁNÍ

Tvorba bakalářská práce začala nastudováním řešené problematiky. Následovalo zprovoznění vývojového prostředí (IDE) RStudia společně s implementováním jazyka R na lokálním zařízení. Během tvorby aplikace Shiny byly do prostředí R přidávány potřebné přídatné balíčky a funkčnost aplikace byla testována ve webovém prohlížeči.

Použité metody

Tvorba aplikace probíhala formou psaní R kódu v prostředí RStudia. Bylo využíváno metody reaktivního programování, kdy jsou v části uživatelského rozhraní (UI) přidávány vstupní prvky, jejichž hodnoty vstupují na stranu serveru a jsou zde zakomponovány do výpočtů a funkcí. Ze serverové strany jsou výstupní hodnoty posílány zpět do výstupních prvků obsažených v UI. Pro úpravu vzhledu bylo využito CSS stylů jednak jako vlastností funkcí v R, tak i jako odděleného CSS souboru importovaného do aplikace. Funkčnost aplikace byla testována v IDE pomocí možnosti spuštění aplikace na lokálním prostředí ve webovém prohlížeči. Pro publikování vytvořené aplikace bylo využito webové platformy *shinyapps.io*, která je propojená s vývojovým prostředím RStudia pomocí přídatné knihovny, uživatelského účtu a bezpečnostního tokenu odlišující jednotlivé aplikace.

Pro prostorová data byly využity metody kartogramu, kartodiagramu, plošných, liniových a bodových znaků na základě funkcionality knihovny *Leaflet* (Cheng et al. 2022). Pro ukázkou statistické vizualizace byly vybrány statistické grafy – histogram a boxplot, vzájemné vztahy – korelační diagram, korelační matice, paralelní osy a liniový graf zobrazující časový vývoj hodnot. Prohlížet data lze pomocí tabelární vizualizace.

Použitá data

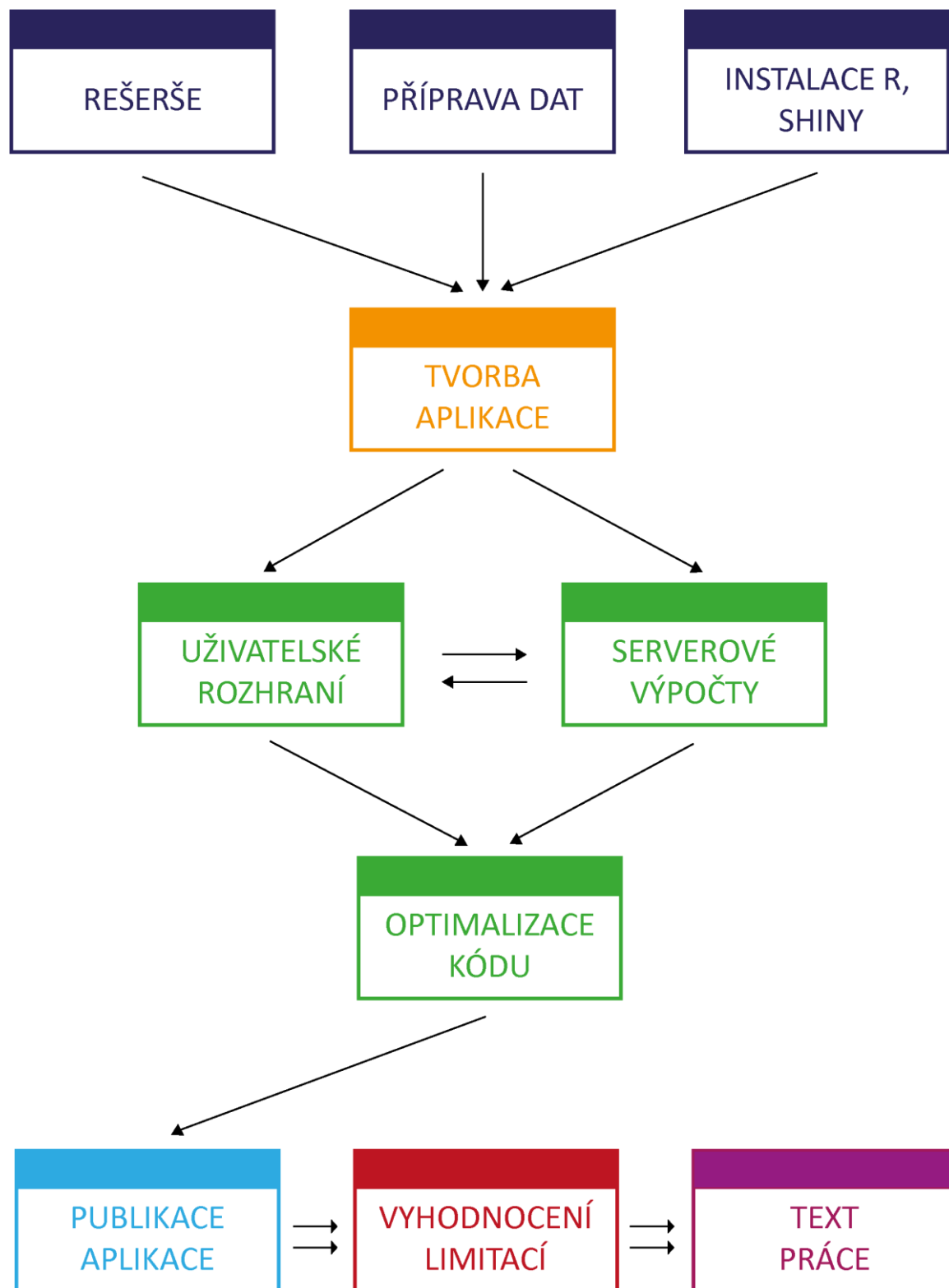
Pro demonstraci vizualizací bylo použito několik datových sad obsahujících prostorovou a atributovou složku. Datová sada *ukazatelé kvality života* ze statistik Eurostatu za rok 2013 a 2015 vztahující se na evropské územní jednotky NUTS2 byla vypůjčena od vedoucího bakalářské práce Mgr. Karla Macků, Ph.D. a použita u prostorových i statistických vizualizací. Dále byla vytvořena datová sada *demografické údaje ČR* ze statistik ČSÚ v letech 2010 a 2020 vztahující se na územní jednotky krajů, okresů a ORP a datová sada *populace vybraných států Evropy* ze statistik Eurostatu v letech 1960 až 2021. Územní celky ČR v prostorové formě byly získány z geodatabáze *ArcČR 500*, verze 3.3.

Použité programy

Tab. 2.1 Informace o použitém softwaru

Software	Verze	Využití
RStudio	2021.09.02	tvorba R kódu, vývoj a publikování aplikace
ArcGIS Pro	2.7.1	tvorba a úprava ukázkových prostorových dat
Visual Studio Code	1.66.2	úprava CSS stylů a JavaScript kódu
Microsoft Excel	2203	tvorba a úprava datových sad

Postup zpracování



3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

V kapitole je čtenář obeznámen s možnostmi vizualizací dat. Čtenáři je vysvětlen pojem webová aplikace, je uveden do prostředí R a seznámen s frameworkem Shiny a jeho funkcionalitou. Kapitola se také zabývá možnostmi publikace aplikací Shiny a jejími alternativami.

3.1 Vizualizace dat

Vizualizací dat se rozumí vizuální zobrazení statistických, prostorových a jiných typů číselných i nečíselných dat pomocí statických nebo interaktivních obrázků a grafiky. Hlavním cílem vizualizace dat je objasnit vzorce, mezery, schémata, pravidelnosti a souvislosti, které nemusí být snadno identifikovatelné čtením nezpracovaných dat nebo dlouhých textů.

Společnost je stále více vystavená datům. Do roku 2003 bylo celosvětově vyprodukováno celkem pět exabytů dat, v roce 2013 byl tento počet dat produkován každý den (Gunelius 2014) a za rok 2020 odhadovaná celosvětová produkce dat odpovídala 59 zettabytům (CloverDX 2021). Význam a pochopení čísel stále často zůstávají ztraceny ve sloupcích a řádcích. Vzhledem k tomu, že znalosti jsou stále více závislé na velkém množství dat, vizualizace se stává nejen praktickým, ale i nezbytným nástrojem pro lepší porozumění (Gatto 2015).

3.1.1 Metody prostorové vizualizace

Bodové znaky

Bodový znak je vyjadřovací prostředek znázorňující v mapě objekt nebo jev a zároveň může znázorňovat jeho vlastnost. Může však být použit i jako součást jiných vyjadřovacích prostředků. Užití bodových znaků jako metody vyžaduje respektování všech ustálených pravidel tematické kartografie, které s použitím bodových znaků souvisí. Tato pravidla vycházejí přímo z definovaných vlastností kartografického znaku a z parametrů bodového znaku (Voženílek a Kaňok 2011).

Liniové znaky

Liniový znak se vyskytuje v mapách jednak jako samostatný vyjadřovací prostředek, jednak jako součást složitějších plošných a jiných vyjadřovacích prostředků (kartogram, kartodiagram a další). Tak jako u bodových znaků tak i u liniových znaků je vybírán znak podle potřeby znázorňovaného objektu nebo jevu, z hlediska kvalitativního nebo kvantitativního (Voženílek a Kaňok 2011).

Plošné znaky

Každá mapa obsahuje minimálně jeden plošný (areálový) kartografický znak. Metoda plošných znaků je jednou z nejčastěji využívaných metod a teoreticky dobře rozpracovanou kartografy. Metoda je používána ke znázorňování plošných prostorových jevů, které lze v měřítku mapy plošně zakreslit. Je postavena na používání plošných bodových znaků, které vystupují buď jako samostatný vyjadřovací prostředek, nebo jako součást složitějších metod (Voženílek a Kaňok 2011).

Kartogram

Metoda kartogramu umožňuje kvantitativní srovnávání jednotlivých dílčích územních celků, ve zvláštních případech i regionalizaci jevu. Podstatou metody kartogramu je znázornění jevu vyjádřeného relativními hodnotami tak, aby byly dílčí

územní celky srovnatelné, a proto musí být kvantitativní data přepočtena na jednotku plochy dílčího územního celku, např. počet obyvatel na 1 km², hustota říční sítě na 10 km² nebo průměrný výnos žita na 1 ha kraje apod. (Voženílek a Kaňok 2011).

Kartodiagram

Kartodiagramy jsou mapová díla pro znázorňování kvantity, především pro znázorňování absolutních hodnot jevu. Mají široké uplatnění ve všech oborech, které pracují s prostorovými daty, jako je ekonomie, demografie, humánní i fyzická geografie a další obory. Jsou vhodné především pro srovnání konkrétních hodnot v dílčích územních jednotkách na mapě, např. počet obyvatel, objem výroby, poměr vývozu a dovozu apod. Někdy jsou označovány málo používaným termínem „diagramové mapy“ (Voženílek a Kaňok 2011).

3.1.2 Metody statistické vizualizace

Histogram

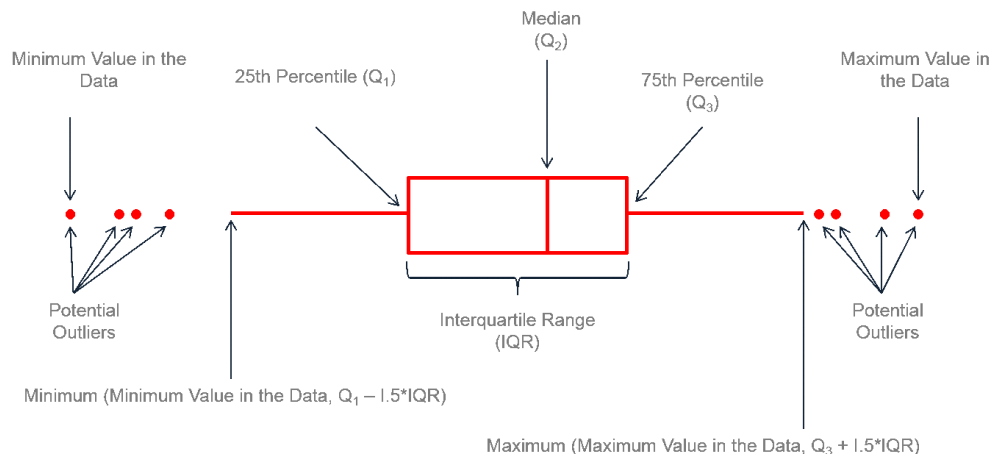
Histogram je přesné grafické znázornění rozdělení číselné proměnné. Jako vstupní data přijímá pouze číselné hodnoty. Graf je rozdělen do několika polí (angl. bins) a počet pozorování v každém poli je reprezentován výškou sloupce. Histogramy jsou používány ke studiu rozložení jedné nebo několika proměnných. Kontrola rozložení jednotlivých proměnných je pravděpodobně prvním úkolem, který by měl být proveden při získání nové datové sady (Holtz 2018c).

Boxplot

Boxplot, nebo krabicový graf, je vhodný způsob vizuálního zobrazení rozdělení dat pomocí kvartilů. Je složen z několika základních prvků:

- čára rozdělující box na dvě části představuje medián dat. Pokud je medián 10, pak pod a nad hodnotou 10 je stejný počet datových bodů,
- konce boxu znázorňují horní a dolní kvartil. Pokud je třetí kvartil 15, pak 75 % pozorování je nižších než 15,
- rozdíl mezi kvartily 1 a 3 je nazýván mezikvartilové rozpětí,
- tečky (nebo jiné značky) za krajní přímkou ukazují potenciální odlehlé hodnoty.

Ačkoli se boxploty mohou zdát ve srovnání s histogramem primitivní, jejich výhodou je menší využití místa v grafickém poli. Lze je tak využít při porovnávání rozdělení mezi více skupinami nebo soubory dat (Holtz 2018a).



Obr. 3.1 Boxplot a jeho znázorněné prvky (r-graph-gallery.com).

Korelační diagram

Korelační diagram, známý také jako graf rozptylu, bodový graf nebo X-Y graf, využívá soubor bodů, umístěných v grafu pomocí kartézských souřadnic, k zobrazení hodnot dvou proměnných a umožňuje zjistit, zda existuje mezi proměnnými vztah či korelace. Sílu korelace lze určit podle toho, jak těsně jsou body na grafu u sebe. Body končící daleko mimo obecný shluk bodů, jsou nazývány odlehle hodnoty. (Ribecca 2019b).

Korelační matice

Korelační matice umožňují analyzovat vztah mezi každou dvojicí číselných proměnných souboru dat. Vztah mezi jednotlivými dvojicemi proměnných je vizualizován ve tvaru matice pomocí grafu rozptylu nebo symbolu, který znázorňuje korelaci. Korelační matice jsou spíše určené pro explorační analýzu, jelikož umožňují na první pohled vizualizovat vztahy celého souboru dat (Holtz 2018b).

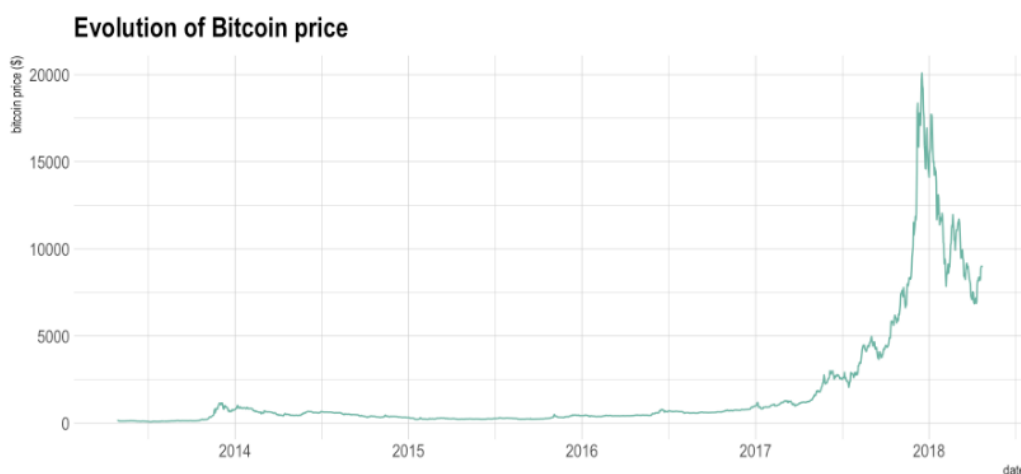
Paralelní osy

Tento typ vizualizace je používán pro vykreslování vícerozměrných číselných dat. Paralelní osy (paralelní souřadnicové grafy) jsou ideální pro porovnávání mnoha proměnných dohromady a sledování vztahů mezi nimi. Každá proměnná má svou vlastní osu a všechny osy jsou umístěny rovnoběžně. Osy mohou mít různá měřítka a jednotlivé proměnné mohou pracovat s jinou měrnou jednotkou. Hodnoty jsou vykreslovány jako řada čar spojující se přes všechny osy.

Pořadí, v jakém jsou osy uspořádány, může ovlivnit způsob, jakým čtenář data pochopí. Jedním z důvodů je, že vztahy mezi sousedními proměnnými jsou snáze vnímatelné než u nesousedních proměnných. Změna pořadí os tedy může pomoci při odhalování vzorců nebo korelací mezi proměnnými (Ribecca 2019a).

Liniový graf

Liniový graf zobrazuje vývoj jedné nebo několika číselných proměnných. Body jsou spojeny přímými úsečkami. Je podobný korelačnímu diagramu, ovšem body jsou seřazeny (obvykle podle hodnoty na ose x) a spojeny přímými úsečkami. Liniový graf je často používán k vizualizaci trendu dat v časových intervalech - časové řady - proto je přímka často zakreslena chronologicky (Holtz 2018d).



Obr. 3.2 Vývoj hodnoty Bitcoinu zobrazený pomocí liniového grafu (data-to-viz.com).

3.2 Webové aplikace

Webová aplikace je aplikační program uložený na vzdáleném serveru a poskytovaný prostřednictvím internetu přes rozhraní prohlížeče. Webové aplikace mohou být navrženy pro nejrůznější použití a může je používat kdokoli, od organizace až po jednotlivce. Mezi běžně používané webové aplikace může patřit webmail, online kalkulačky nebo e-shopy. K některým webovým aplikacím lze přistupovat pouze prostřednictvím určitého prohlížeče, většina z nich je však dostupná bez ohledu na prohlížeč.

Webové aplikace není třeba stahovat, protože je k nim přistupováno prostřednictvím sítě. Uživatelé mohou k webové aplikaci přistupovat prostřednictvím webového prohlížeče, například Google Chrome, Mozilla Firefox nebo Safari. Webové aplikace mají obvykle krátký vývojový cyklus a mohou je vytvářet i malé vývojové týmy. Většina webových aplikací je napsána v jazycích JavaScript, HTML5 nebo kaskádových stylech (CSS). Tyto jazyky jsou obvykle využívány k programování na straně klienta, zpravidla pro tvorbu uživatelského rozhraní. Programování na straně serveru slouží k vytváření skriptů, které bude webová aplikace používat. Při programování na straně serveru jsou běžně používány jazyky Python, Java a Ruby (TechTarget Contributor 2019).

3.3 Jazyk R

R je jazyk a prostředí pro statistické výpočty a grafiku. Jedná se o projekt GNU (Free Software Foundation 2022), vyvinutý v Bellových laboratořích (dříve AT&T, nyní Lucent Technologies) Johnem Chambersem a jeho kolegy. R lze považovat za odlišnou implementaci jazyka S, jelikož jsou tyto dva jazyky velice podobné. R nabízí širokou škálu statistických a grafických technik. R lze rozšiřovat pomocí balíčků a knihoven. S distribucí R je dodáváno osm balíčků a mnoho dalších je k dispozici prostřednictvím internetových stránek CRAN, které pokrývají velmi širokou škálu moderní statistiky.

R je k dispozici jako svobodný software (free and open source) za podmínek GNU General Public License (Free Software Foundation 2022). Nepotřebuje kompilaci a funguje na široké škále platform UNIX, Windows a MacOS.

R, stejně jako S, je navržen na základě skutečného počítačového jazyka a umožňuje uživatelům přidávat další funkce definováním nových funkcí. Pro výpočetně náročné úlohy lze propojit kód v jazycích C, C++ a Fortran a volat jej za běhu. Pokročilí uživatelé mohou psát kód v jazyce C a manipulovat s objekty R přímo (The R Foundation 2022).

3.4 RStudio

RStudio je integrované vývojové prostředí (IDE) pro jazyk R. Obsahuje konzolový editor podporující přímé spuštění kódu a umožňující zvýrazňovat syntaxi. Součástí prostředí jsou nástroje pro vykreslování, historii, ladění programů a správu pracovního prostoru.

Existují dvě základní verze, *RStudio Desktop* poskytující RStudio IDE jako nativní desktopovou aplikaci a *RStudio Server* umožňující poskytnout rozhraní založené na webovém prohlížeči běžící na vzdáleném Linuxovém serveru, které přináší výkon a funkce prostředí RStudio IDE do serverového nasazení jazyka R (RStudio 2022a).

3.5 Shiny

Shiny je open source framework pro vytváření webových aplikací pomocí jazyka R. Je navržen pro datové vědce a analytiku, kterým je umožněno vytvářet interaktivní, komplexní, složité webové aplikace bez nutnosti znalosti HTML, CSS nebo JavaScript. Nicméně, lze vytvářet i jednoduché aplikace a také kombinovat jazyky pro tvorbu webových aplikací s jazykem R a dosáhnout tak překvapivého výsledku.

Každá aplikace Shiny je složena ze dvou částí, *uživatelského rozhraní* (UI) definující vzhled aplikace a umožňující uživatelům interagovat s jejím prostředím a *funkci serveru* starající se o fungování aplikace a vykonávající výpočty na straně serveru. Shiny využívá reaktivní programování (kap. 3.5.1) k automatické aktualizaci výstupů při změně vstupů od uživatele (Wickham 2021a). Do prostředí RStudio je framework importován v rámci CRAN knihovny *Shiny* (Chang et al. 2021b).

```
library(shiny)

ui <- fluidPage(
  # front end interface
)

server <- function(input, output, session) {
  # back end logic
}

shinyApp(ui, server)
```

Obr. 3.3 Základní části aplikace Shiny v jazyku R.

3.5.1 Reaktivní programování

Reaktivní programování je elegantní a výkonné programovací paradigma, ale zpočátku může být dezorientující, protože se jedná o zcela odlišné paradigma než psaní skriptů. Klíčovou myšlenkou reaktivního programování je automatické aktualizování všech souvisejících výstupů při změně vstupů. Na rozdíl od základního kódu R, který je většinou vykonáván od shora dolů, v Shiny je určitá část kódu spouštěna pouze tehdy, když je to potřeba (např. uživatel klikne na tlačítko pro vykreslení grafu).

Vstup (angl. Input) je objekt podobný seznamu, který obsahuje všechna vstupní data odeslaná z prohlížeče, pojmenovaná podle identifikátoru. *Výstup (angl. Output)* je velmi podobný vstupu: je to také objekt podobný seznamu pojmenovaný podle identifikátoru. Hlavní rozdíl spočívá v tom, že je používán pro odesílání výstupu namísto přijímání vstupu. Objekt výstupu je vždy používán s funkcí pro vykreslování (Wickham 2021a).

3.5.2 Možnosti publikace

Publikovat Shiny aplikace a sdílet je mezi uživateli lze dvěma základními způsoby, publikovat aplikaci jako R skript nebo publikovat aplikaci jako webovou aplikaci. Publikování R skriptu je nejjednodušší způsob sdílení aplikace, který však funguje pouze v případě, když uživatelé mají na svém zařízení R a jsou obeznámeni s jeho používáním. Uživatelé mohou pomocí skriptů spouštět aplikace ze své vlastní relace R. Skripty lze publikovat např. pomocí veřejného repositáře GitHub.

Pokud uživatel nemá k dispozici R, je vhodné aplikaci Shiny publikovat jako webovou aplikaci. RStudio nabízí tři možnosti publikace: *shinyapps.io*, *Shiny Server* a *RStudio Connect* (Shiny from RStudio 2020).

Shinyapps.io

Shinyapps.io je samoobslužná platforma usnadňující sdílení Shiny aplikací na webu. Služba je poskytována v rámci cloudu na sdílených serverech, které provozuje společnost RStudio. Každá aplikace je spouštěna samostatně, oddělená od ostatních aplikací, a pracuje s daty, které byly nahrány spolu s aplikací, databázemi či webovými službami. Importované data a data aplikace jsou zabezpečeny šifrovacími funkcemi.

Aplikaci Shiny lze publikovat přímo z prostředí RStudia pomocí balíčku *rsconnect* (Atkins et al. 2021). Každá aplikace je v *shinyapps.io* umístěna na vlastním virtualizovaném serveru, který je nazýván instance. Na každé instanci je spuštěna identická kopie kódu, balíčků a dat publikované z prostředí R. RStudio nabízí bezplatné i placené plány pro používání *shinyapps.io* odlišující se různým omezením. (*shinyapps.io* team 2021).

Shiny Server

Shiny Server je doprovodný program k aplikaci Shiny vytvářející webový server určený k hostování aplikací Shiny. Je zdarma, má otevřený zdrojový kód a je k dispozici na GitHubu. Shiny Server lze spustit pouze na serverech Linux s podporou Ubuntu. Na jednom Shiny Serveru lze publikovat více Shiny aplikací na více webových stránkách (Shiny from RStudio 2020).

RStudio Connect

RStudio Connect je platforma pro publikaci prací vytvořených v jazycích R a Python. Jedná se o komerční prostředí, které firmám umožňuje uchovávat a sdílet informace na jednom sjednoceném místě. Podporuje:

- Shiny aplikace,
- R Markdown dokumenty,
- Plumber API,
- Dashboardy,
- Jupyter Notebooks,
- interaktivní obsah v jazyce Python.

Pomocí RStudio Connect lze publikovat aplikace, dokumenty a reporty přímo z prostředí RStudia. Dále umožňuje automatizovaně distribuovat a aktualizovat data (RStudio 2022b).

3.5.3 Alternativy Shiny

Dash

Jedná se o framework v jazyce Python vytvořený společností Plotly sloužící k vytváření webových aplikací. Je napsán pomocí knihoven Flask, Plotly.js a React.js. Umožňuje rychle a vizuálně vyjadřovat různé analýzy. Stejně jako u Shiny, Dash využívá reaktivního programování. Výhodou Shiny může být kratší kód pro vytvoření aplikace, jednodušší tvorba uživatelských elementů a více možností, kde publikovat výsledné aplikace. Dash aplikace lze publikovat pomocí Dash Enterprise nebo v rámci Amazon webových služeb AWS. Výhodou Dash je jednodušší stylování vzhledu aplikace a jednotlivých prvků a také efektivnost jazyka Python (Appsilon 2022).

Streamlit

Streamlit je open-source Python knihovna, která umožňuje snadno vytvářet a sdílet webové aplikace pro strojové učení a datovou vědu. Vizualizace, grafy, widgety a další různé prvky jsou přidávány do aplikace pomocí vlastních API. Vytvořené aplikace lze publikovat pomocí Streamlit Cloud webového rozhraní, kdy je účet Streamlit Cloud propojen s repositářem na GitHub a Streamlit Cloud pak spouští aplikace přímo z kódu uloženého v repositáři. Streamlit Cloud lze využívat na základě plánů, do kterých patří i verze zdarma (Streamlit Inc. 2022).

Power BI

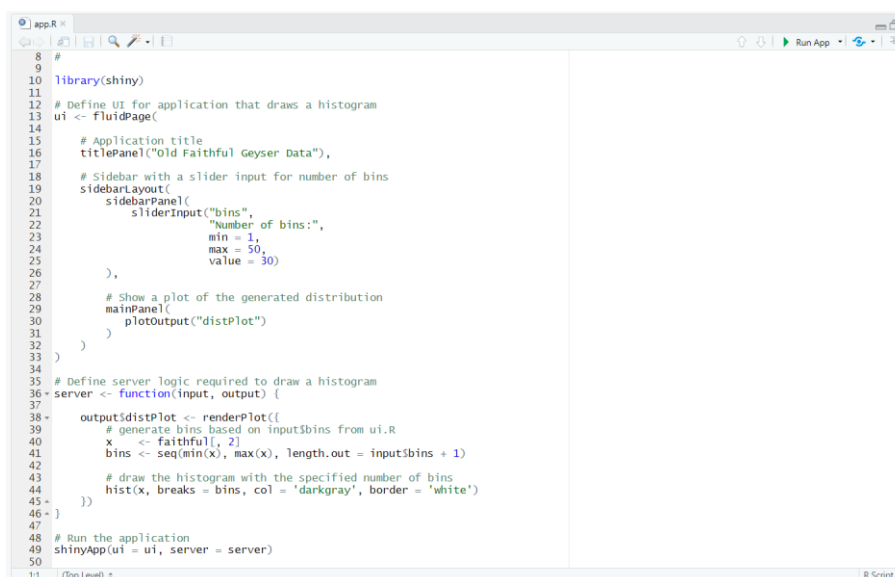
Power BI je vyvíjen společností Microsoft. Jedná se o soubor softwarových služeb a aplikací spolupracujících na přeměně nesouvisejících zdrojů dat v ucelené, vizuálně poutavé a interaktivní informace v podobě dashboardů. Power BI je složen ze tří základních prvků – desktopová aplikace Power BI Desktop sloužící pro tvorbu dashboardů a zpracování dat, online služba Power BI service, pomocí které lze sdílet vytvořené reporty a vizualizace, a mobilní verze Power BI mobile apps pro prohlížení publikovaných aplikací v mobilním prostředí (Hart et al. 2022).

4 APLIKACE SHINYVISION

V rámci bakalářské práce vznikla webová aplikace nesoucí název ShinyVision. Aplikace má za cíl vizualizovat prostorové a neprostorové data pomocí vhodných metod a umožnit uživateli tyto data prohlížet v tabelárním zobrazení. ShinyVision byla tvořena v prostředí RStudio pomocí programovacího jazyka R a pro úpravu vzhledu některých prvků byly použity kaskádové styly CSS.

4.1 Implementace Shiny do RStudio

Prostředí RStudio v základní verzi neobsahuje framework Shiny. Je nutné ho instalovat v rámci balíčku CRAN knihovny *Shiny* (Chang et al. 2021b). Po instalaci balíčku byl založen nový Shiny projekt. Následně je vždy vytvořena ukázková aplikace Shiny, pomocí které je možné otestovat správnost chodu balíčku Shiny. V RStudio je umožněno spouštět aplikace na lokálním prostředí pomocí tlačítka „Run App“ a také publikovat aplikace na serverovou část *shinyapps.io* nebo *RStudio Connect* (Obr. 4.1).



```
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15   # Application title
16   titlePanel("Old Faithful Geyser Data"),
17
18   # Sidebar with a slider input for number of bins
19   sidebarLayout(
20     sidebarPanel(
21       sliderInput("bins",
22                 "Number of bins:",
23                 min = 1,
24                 max = 50,
25                 value = 30)
26     ),
27
28     # Show a plot of the generated distribution
29     mainPanel(
30       plotOutput("distPlot")
31     )
32   )
33
34 # Define server logic required to draw a histogram
35 server <- function(input, output) {
36
37   output$distPlot <- renderPlot({
38     # generate bins based on input$bins from ui.R
39     x <- faithful[, 2]
40     bins <- seq(min(x), max(x), length.out = input$bins + 1)
41
42     # draw the histogram with the specified number of bins
43     hist(x, breaks = bins, col = 'darkgray', border = 'white')
44   })
45
46 }
47
48 # Run the application
49 shinyApp(ui = ui, server = server)
50
```

Obr. 4.1 Ukázková Shiny aplikace po založení projektu.

Kód aplikace je uložen ve složce projektu v souboru *app.R*. Dále je umožněno v adresáři projektu vytvořit složku s názvem *www*, do které jsou vkládány přídatné soubory renderovány ve webovém prohlížeči společně s aplikací Shiny, např. kaskádové styly, obrázky, datový soubor JSON.

4.2 Layout aplikace

Začátkem tvorby aplikace bylo navržení rozvržení ovládacích prvků, tzv. layoutu. K tomu byla využita knihovna *shinydashboard* (Chang et al. 2021a) umožňující jednoduše vytvářet interaktivní dashboardy.

Layout aplikace je složen ze tří hlavních částí. První částí je hlavička, v R kódu funkce *dashboardHeader()*, ve které je umístěné logo aplikace, tlačítko na ovládní bočního panelu a informativní nadpis.

Druhá část představuje interaktivní boční panel, funkce *dashboardSidebar()*, obsahující menu se strukturovanými záložkami. Při kliknutí na určitou záložku

je zobrazena stránka s obsahem. Záložky, funkce `menuItem()` a `menuSubItem()`, jsou odlišovány unikátním identifikátorem pro určení zobrazení konkrétní stránky s obsahem. Jednotlivým záložkám je přiřazena ikona z internetového balíčku ikon *Font Awesome* (Fonticons 2022) funkcí `icon()` z knihovny *Shiny* a funkcí `HTML()` z knihovny *htmltools* (Cheng et al. 2021) umožňující zapsání HTML elementů v R kódu (Obr. 4.2).

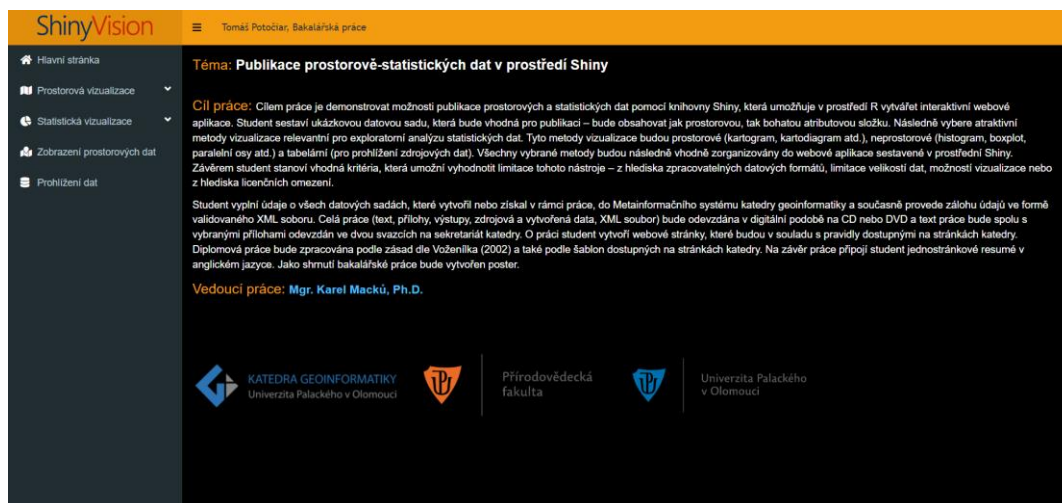
```

dashboardSidebar(
  width = "250px",
  minified = FALSE,
  collapsed = FALSE,
  sidebarMenu(
    menuItem("Hlavní stránka", tabName = "maintab", icon = icon("home")),
    menuItem("Prostorová vizualizace", icon = icon("map")),
    menuSubItem("Kartogram", tabName = "tabKarto"),
    menuSubItem("Kartodiagram", tabName = "tabKartodia")
  ),
  menuItem("Statistická vizualizace", icon = icon("chart-pie"),
    tags$div(class = "menuNadpis", HTML('<i class="fa fa-bar-chart" style="color:#d8e1e6;"></i> &nbsp;Statistické grafy')),
    menuSubItem("Histogram", tabName = "tabHisto"),
    menuSubItem("Boxplot", tabName = "tabBoxplot"),
    menuSubItem("Korelační diagram", tabName = "tabScatter"),
    menuSubItem("Korelační matice", tabName = "tabKorelMat"),
    menuSubItem("Paralelní osy", tabName = "tabParaOsy"),
    tags$div(class = "menuNadpis", HTML('<i class="fas fa-project-diagram" style="color:#d8e1e6;"></i> &nbsp;Vzájemné vztahy')),
    menuSubItem("Liniový graf", tabName = "tabTimeline")
  ),
  menuItem("Zobrazení prostorových dat", tabName = "tabGeodata", icon = icon("map-marked")),
  menuItem("Prohlížení dat", tabName = "tabData", icon = icon("database"))
),
)

```

Obr. 4.2 Ukázka kódu bočního panelu.

Ve třetí části jsou pomocí funkce `dashboardBody()` nadefinovány stránky s obsahem. Do téhle funkce je také vkládán HTML element hlavičky `tags$head()` v němž jsou importovány externí soubory ze složky *www* – CSS styly a soubor obsahující JavaScript kód. Obsah určité stránky je zapisován funkcí `tabItem()` odlišený unikátním identifikátorem ze záložek bočního panelu.

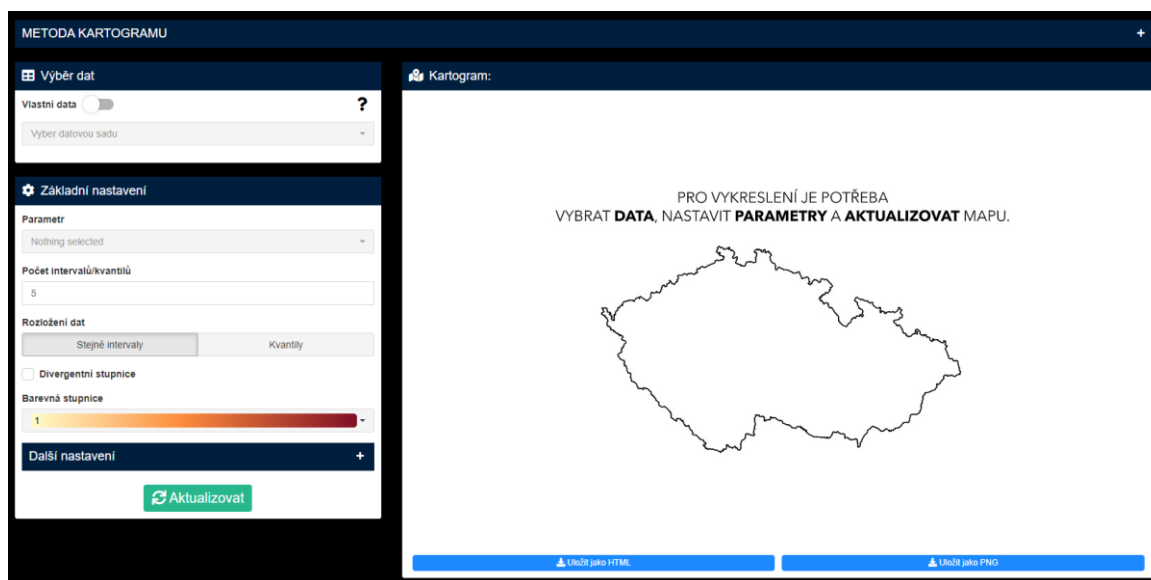


Obr. 4.3 Layout aplikace ShinyVision.

4.3 Uživatelské rozhraní

Kompozice stránky byla rozvržena pomocí funkcí `fluidRow()` a `column()` z knihovny *Shiny*. Tyto funkce umožňují rozdělit prvky do sloupců a řádků s přesně definovanou šířkou. Pro tvorbu UI u všech vizualizací byla využita knihovna *shinydashboard* (Chang et al. 2021a) s rozšiřující knihovnou *shinydashboardPlus* (Granjon et al. 2021). Funkce `box()` vytváří jednotlivé sekce (boxy), do kterých lze vkládat uživatelské prvky, například uživatelské vstupy, a uživateli je usnadněná orientace v jednotlivých nastavení vizualizace.

Prostorové a statistické vizualizace jsou vždy složeny ze sekce informativní, kdy při kliknutí na ikonu „plus“ je otevřen panel s popisem dané vizualizace. Sekce výběru dat slouží k nastavení jednak předdefinovaných dat, tak i výběru vlastních dat. V této sekci je umístěna ikona otazníku, po kliknutí na ikonu je otevřen panel informující o struktuře dat. Tato funkce byla vytvořena kombinací jazyků R, JavaScript knihovny *jQuery* (Resig a The jQuery Team 2021) a kaskádových stylů CSS. Sekce základního nastavení slouží uživateli ke konfiguraci jednotlivých vstupů pro vykreslení vizualizace. Sekce obsahuje podsekcí „další nastavení“ pro doplňující nastavení vizualizace, např. výběr barev nebo úprava vyskakovacích oken. Hlavní sekce, pojmenovaná po vybrané vizualizaci a vybrané datové sadě, slouží pro umístění vykreslené vizualizace.



Obr. 4.4 Uživatelské rozhraní u metody kartogramu.

Vstupní prvky

Pro implementaci vstupních uživatelských prvků byly využity knihovny *Shiny*, *shinyWidgets* (Perrier et al. 2022) a *colourpicker* (Attali a Griswold 2021). Seznam použitých vstupních prvků je k dispozici v tabulce 4.1.

Tab. 4.1 Seznam vstupních prvků

vstupní prvek	popis	knihovna
actionBtnn	tlačítko pro vyvolání akce	shinyWidgets
awesomeCheckbox	zaškrťovací tlačítko	shinyWidgets
colourInput	výběr HEX kódů barev	colourpicker
fileInput	výběr souborů z lokálního úložiště	Shiny
materialSwitch	přepínatelné tlačítko	shinyWidgets
numericInput	výběr číselných hodnot	Shiny
pickerInput	výběr prvků ze seznamu položek	shinyWidgets
radioGroupButtons	skupina interaktivních tlačítek	shinyWidgets
textInput	vstup vlastního textu	Shiny

4.4 Výběr dat

Funkcionalita výběru dat obsahuje možnost vybrat si předdefinovaná data z výběru ze seznamu dostupných datových sad pomocí funkce `pickerInput()`, nebo po přepnutí tlačítka pro výběr vlastních dat lze nahrát data ze svého úložiště funkcí `fileInput()`.

Již při přepnutí tlačítka výběru dat dochází k reaktivitě. Na straně serveru je v rámci observeru vykonána podmínka skrývající `pickerInput` a zobrazující `fileInput` a naopak (Obr. 4.5). Observer je jeden ze stavebních kamenů reaktivního programování. Blok kódu uvnitř observeru je vykonáván okamžitě při změně vstupu a nevrací žádnou hodnotu oproti jiným reaktivním operacím (Wickham 2021b).

```
observe({
  if(input$data_switch == FALSE){
    show(id = "data_sada_k")
    hide(id = "file_karto")
  } else {
    show(id = "file_karto")
    hide(id = "data_sada_k")
  }
})
```

Obr. 4.5 funkce `observe()` s podmínkou pro skrytí a zobrazení výběrů.

Načtení dat probíhá na straně serveru pomocí funkce `reactive()`. Oproti observeru, reaktivní funkce vrací hodnotu, proto je vždy potřeba deklarovat proměnnou uchovávající právě hodnotu z reaktivní funkce. Pro odlišení vlastních či předdefinovaných dat byla použita rozvětvená podmínka na základě přepínatelného tlačítka vracejícího hodnoty `TRUE` nebo `FALSE`.

Shapefile

Pro zpracování výběru vlastních dat ve formátu ESRI Shapefile byla použita myšlenka kódu (Moraga 2019) a modifikována dle potřeb (Obr. 4.6). Uživatel je povinen vybrat alespoň čtyři základní Shapefile soubory s příponou `.shp`, `.prj`, `.dbf` a `.shx`. Do proměnné `soubor` jsou uloženy hodnoty z výběru `fileInput` – názvy souborů, a do proměnné `tempdirname` vstupují cesty k těmto souborům. V cyklu `for()` dochází k připojení názvu a cesty k souborům. K přečtení výsledného Shapefile souboru slouží

funkce `st_read()` z knihovny `sf` (Pebesma et al. 2022) a funkce `paste()` sloužící v tomto případě jako sloučení vybraných Shapefile souborů do jednoho souboru s příponou `.shp` pro potřeby funkce `st_read()`. Pomocí funkce `st_transform()` je přetransformován souřadnicový systém na WGS84. Výsledná hodnota je uložena do deklarované proměnné na začátku reaktivní funkce.

```
soubor <- input$file_karto
tempdirname <- dirname(soubor$datapath[1])

for (i in 1:nrow(soubor)) {
  file.rename(
    soubor$datapath[i],
    paste0(tempdirname, "/", soubor$name[i])
  )
}
shp <- st_read(paste(tempdirname,
                    soubor$name[grep(pattern = "*.shp$", soubor$name)],
                    sep = "/"))
shp <- st_transform(shp, "+proj=longlat +datum=WGS84 +no_defs")
shp
```

Obr. 4.6 Nahrávání Shapefile souboru.

JSON/GeoJSON

Zpracování vlastního výběru souborů JSON je o poznání jednodušší, než u souboru Shapefile. Uživateli je umožněno vybírat soubory s příponou `.json` nebo `.geojson`. Funkcí `file_ext()` z knihovny `xfun` (Xie et al. 2022b) je extrahována přípona souboru vstupující do podmínky pro odlišení datového formátu. Po splnění podmínky je soubor přečten funkcí `st_read()` a přetransformován na souřadnicový systém WGS84.

```
#geojson
if (file_ext(soubor) == "json" || file_ext(soubor) == "geojson") {
  shp <- st_read(input$file_karto$datapath)
  shp <- st_transform(shp, "+proj=longlat +datum=WGS84 +no_defs")
  shp
}
```

Obr. 4.7 Nahrávání GeoJSON souboru.

List Excel

Uživateli je umožněno nahrát Excel soubor s příponou `.xlsx` nebo `.xls` a vybrat si určitý list. Po vybrání souboru je spuštěn observer obsahující funkci `updatePickerInput()`. Tato funkce provádí aktualizaci výběru hodnot na základě reaktivní funkce. Ve vlastnosti „choices“ jsou přiřazené hodnoty z funkce `excel_sheets()` z knihovny `readxl` (Wickham et al. 2022) ukládající názvy listů a uživatel si tak může vybrat specifický list Excel souboru (Obr. 4.8). Pro správnost fungování bylo zapotřebí vytvořit pomocnou proměnnou obsahující názvy listů a opatřit čtení souboru podmínkou vykonávající blok kódu pouze při shodném názvu listů ve výběru prvků a ve funkci pro čtení excel souboru `read_excel()`.

```

#Update názvů listů z excel souboru
observe({
  req(input$file_kd)
  updatePickerInput(session, "sheets_kd",
                    choices = excel_sheets(input$file_kd$datapath)
  )
})

```

Obr. 4.8 Aktualizace výběru názvu listů.

Soubor CSV

V tabelární vizualizaci je umožněno nahrát CSV soubor (Comma Separated Values). Po nahrání souboru pomocí funkce `fileInput()` pro výběr dat je uživateli zobrazen vstup vlastního textu (`textInput()`) pro zvolení oddělovače hodnot. Data ze souboru jsou načtena funkcí `read.csv()`.

Předdefinovaná data

Uživatel má možnost si zvolit z předdefinovaných dat. Pro odlišení datových sad a jednotlivých roků byly použity hodnoty z přepínatelného tlačítka a hodnoty z výběru datové sady. U metody kartogramu jsou nahrávány data z Shapefile souboru (Obr. 4.9), u ostatních vizualizací s přednastavenými daty je načítán Excel list.

```

#Přednastavená data
} else if (input$data_switch == FALSE & input$data_sada_k == "Kraje2010"){
  shp <- st_read("www/data/dataKraje2010.shp")
  shp <- st_transform(shp, "+proj=longlat +datum=WGS84 +no_defs")
  shp
} else if (input$data_switch == FALSE & input$data_sada_k == "Kraje2020"){
  shp <- st_read("www/data/dataKraje2020.shp")
  shp <- st_transform(shp, "+proj=longlat +datum=WGS84 +no_defs")
  shp
} else if (input$data_switch == FALSE & input$data_sada_k == "Okresy2010"){
  shp <- st_read("www/data/dataOkresy2010.shp")
  shp <- st_transform(shp, "+proj=longlat +datum=WGS84 +no_defs")
  shp
}

```

Obr. 4.9 Ukázka nahrávání předdefinovaných dat

4.5 Prostorové vizualizace

K sestrojení prostorových vizualizací byla použita primárně knihovna *Leaflet* (Cheng et al. 2022). Leaflet je JavaScript knihovna sloužící k tvorbě webových interaktivních map, v prostředí R je implementována jako přídatný balíček.

4.5.1 Metoda kartogramu

Leaflet knihovna umožňuje vytvořit interaktivní kartogram kombinací plošných znaků a vhodně zvolené barevné stupnice. Uživatel si po zvolení datové sady vybere jeden z parametrů. Parametry jsou automaticky načítány do vstupu prvků pomocí funkce `updatePickerInput()` do níž vstupují hodnoty z reaktivní funkce pro načtení názvů parametrů z vybrané datové sady (Obr. 4.10).

```

#Načtení názvů parametrů
parametr_karto_input <- reactive({
  names(data_karto())
})

#Update Input parametrů
observe({
  updatePickerInput(session, "param",
                    choices = parametr_karto_input())
  updatePickerInput(session, "popup_k",
                    choices = parametr_karto_input())
})

```

Obr. 4.10 Načtení názvů parametrů do vstupu prvků.

Barevné stupnice

Knihovna Leaflet je založena na používání barevných stupnic ze souboru barevných palet Color Brewer 2.0 (Brewer 2013). Uživatel má možnost vybrat si z osmi kontinuálních a z pěti divergentních barevných stupnic. Vizuální podoba stupnic ve výběru prvků je řešena HTML elementy a CSS styly, kdy je obarveno pozadí div elementu na základě barevné palety.

Při výběru kontinuální barevné stupnice je na základě číselné hodnoty přiřazen název barevné palety, například uživatel si vybere osmou barevnou stupnici a pomocí podmínky reaktivní funkce vrátí hodnotu „Blues“. Hodnota vstupuje do funkce `colorBin()` nebo `colorQuantile()` na základě toho, zda si uživatel vybral rozložení dat pomocí stejných intervalů nebo kvantilů. Funkce předělá číselné hodnoty vybraného parametru na barevnou stupnici s uživatelským nastavením rozložení dat a počtem intervalů či kvantilů.

Při zaškrtnutí tlačítka pro zobrazení divergentních stupnic je uživateli zobrazen seznam výběru divergentních stupnic. Oproti kontinuálním barevným stupnicím je zde řešen problém přechodu ze záporných hodnot do kladných hodnot stupnice. Při výběru stupnice jsou pomocí podmínek vytvořeny vektory obsahující HEX kódy krajních barev kladných a záporných stupnic. Vektory vstupují do funkce `colorRampPalette()` interpolující hodnoty z vektoru do nové kladné a záporné barevné stupnice. Pomocí výrazu za funkcí je určen počet barev záporných, od 0 po minimální hodnotu parametru, a počet barev kladných, od 0 po maximální hodnotu parametru. Ve výrazu je také zahrnut případ, kdy prostřední hodnota divergentní stupnice není rovna 0. Vzniklé stupnice jsou spojené do jedné a pomocí funkce `colorNumeric()` jsou převedeny hodnoty parametru na barevnou spojitou divergentní stupnici (Obr. 4.11).

```

#Rozložení dat, barevné palety.
paleta_karto <- reactive({

  #Divergentní stupnice
  if(input$divergent_k) {
    zaporne <- colorRampPalette(colors = div_barvy_zaporne(), space = "Lab")(if(min(data_karto()[[input$param]]) < 0){
      0 - min(data_karto()[[input$param]])
    } else {
      0 + min(data_karto()[[input$param]])
    })

    kladne <- colorRampPalette(colors = div_barvy_kladne(), space = "Lab")(0 + max(data_karto()[[input$param]]))
    stupnice <- c(zaporne, kladne)
    colorNumeric(palette = stupnice, domain = data_karto()[[input$param]])
  } else {
    #Barevná stupnice
    if(input$rozlozeni_dat == "Stejně intervaly"){
      colorBin(barvy_k(), data_karto()[[input$param]], bins = input$interval, pretty = FALSE)
    } else if (input$rozlozeni_dat == "Kvantily"){
      colorQuantile(barvy_k(), data_karto()[[input$param]], n = input$interval)
    }
  }
})

```

Obr. 4.11 Vytváření barevných stupnic.

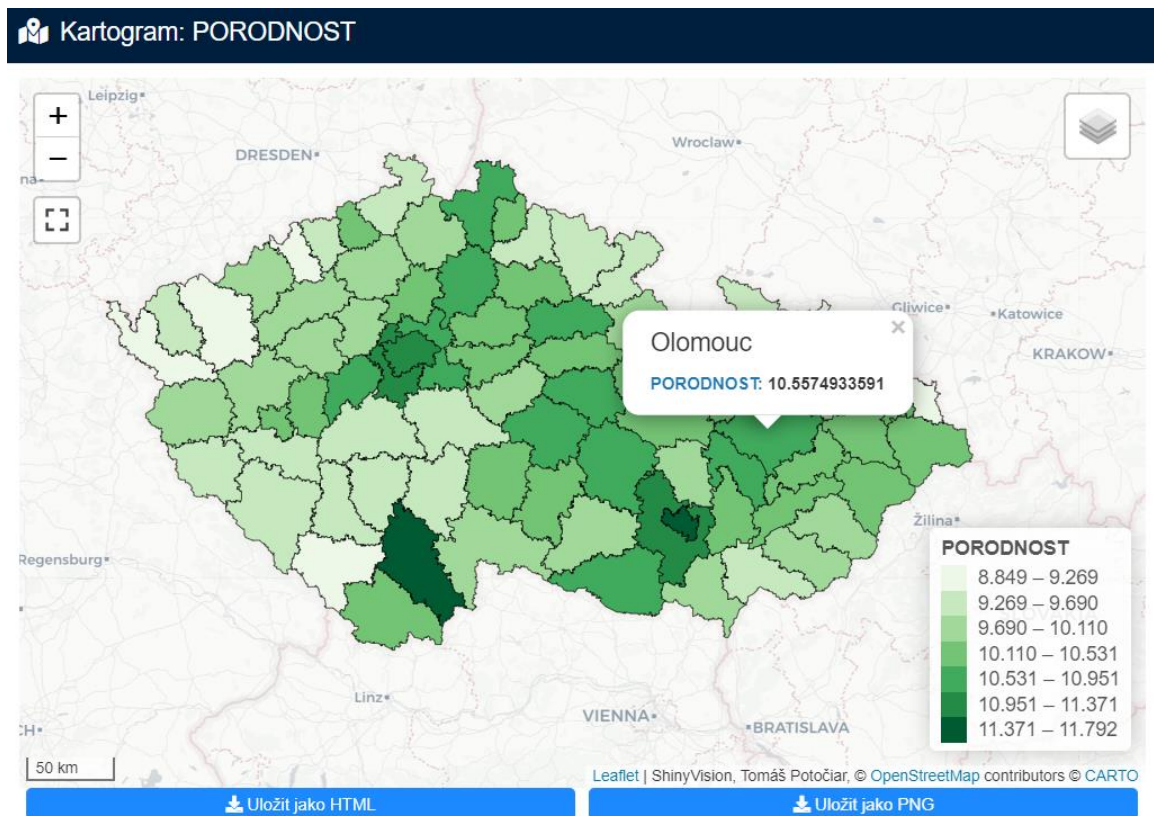
Leaflet mapa

Interaktivní mapa je umístěna v samostatném boxu. V části uživatelského rozhraní je funkcí `leafletOutput()` vytvořen element pro renderování Leaflet mapy. Na serverové straně byla vytvořena reaktivní funkce s „událostí“ `eventReactive()`. Tahle reaktivní funkce neaktualizuje hodnoty automaticky při změně vstupních hodnot, ale až při spuštění události (eventu). Událost této funkce je nastavena na akční tlačítko aktualizace, kdy při kliknutí na tlačítko je vykonán blok kódu obsahující zpracování Leaflet mapy.

Mapa je vytvářena funkcí `leaflet()`. Pro metodu kartogramu byla využita rozšiřující funkce `addPolygons()` umožňující přidat do mapy plošné znaky. Do této funkce vstupují zpracovaná data z reaktivní funkce. Barva polygonů je určena na základě vybrané barevné stupnice a parametru. Informace ve vyskakovacím okně zobrazujícím se po kliknutí na určitý polygon jsou nastaveny podle vybraného parametru pro vykreslení hodnot a parametru pro název jednotky. Do mapového pole je dále přidána legenda mapy, měřítko mapy, tlačítko pro zapnutí zobrazení na celou obrazovku a ovládací panel pro přepínání podkladových map. Funkce pro vkládání podkladových map, `addProviderTiles()`, pracuje s Leaflet extenzí obsahující velké množství podkladových map a dlaždic (Seelmann a GitHub contributors 2021).

U mapy byla také vytvořena funkcionalita přiblížení na zájmové území. Pomocí funkce `st_bbox()` bylo vypočítáno ohraničení zájmové oblasti, tzv. bounding box, a při vykreslení je mapa na něj přiblížena. Tohle přiblížení zajišťuje funkce `fitBounds()`.

Pro vykreslení mapy byla použita funkce `renderLeaflet()` odkazující na výstup mapy odlišený identifikátorem zvoleným v uživatelském rozhraní. Na stejném principu je také vykreslen pomocný text a nadpis mapy pomocí renderovací funkce `renderText()`. Vykreslenou mapu lze stáhnout jako HTML soubor obsahující totožnou Leaflet mapu i s veškerou funkcionalitou, nebo jako obrázek ve formátu PNG (Obr. 4.12). Funkcionalita stahování map byla implementována na základě kódu uživatele Jaccar (Jaccar 2019).



Obr. 4.12 Kartogram porodnosti vykreslený pomocí Leaflet knihovny.

4.5.2 Metoda kartodiagramu

Pro vytvoření kartodiagramu byla použita knihovna *Leaflet* s nadstavbou *leaflet.minicharts* (Bachelier et al. 2021) umožňující přidat do mapy interaktivní diagramy pomocí funkce `addMinicharts()`. Podmínkou pro umístění diagramů do mapy je přítomnost informace o souřadnicích X a Y ve tvaru souřadnicového systému WGS84 v Excel souboru datové sady. Souřadnice slouží pro ukotvení diagramů na určité místo.

Po výběru datové sady a XY souřadnic jsou uživatelem zvoleny parametry vstupující do diagramů. Výběr prvků `pickerInput()` obsahuje argument pro vícenásobný výběr (`multiple = TRUE`) a tak je umožněno uživateli vybrat různý počet parametrů, jenž jsou uloženy do výsledného vektoru. Dále je uživatelem vybrán typ diagramu, kruhový nebo sloupcový, a nastavena předdefinovaná paleta barev z knihovny *RColorBrewer* (Neuwirth 2022). V sekci dalšího nastavení jsou uživatelem nastaveny násobky šířky, výšky a názvy jednotlivých diagramů. Veškeré vybrané prvky vstupují do reaktivní funkce pro vykreslení mapy po kliknutí na aktualizací tlačítko.

Výpočet velikosti kruhových diagramů.

Při výběru kruhového typu diagramu se na straně serveru vypočítává velikost jednotlivých diagramů. Velikost je proporcionálně měněna na základě hodnoty daného diagramu. Obecná formulace výpočtu zní:

$$\text{násobek velikosti} * (\text{druhá odmocnina ze součtu hodnot parametrů} / \text{druhá odmocnina ze součtu maximálních hodnot parametrů}) \quad (1)$$

Výpočet je implementován do R kódu v reaktivní funkci vstupující do argumentu funkce `addMinicharts()` (Obr 4.13).

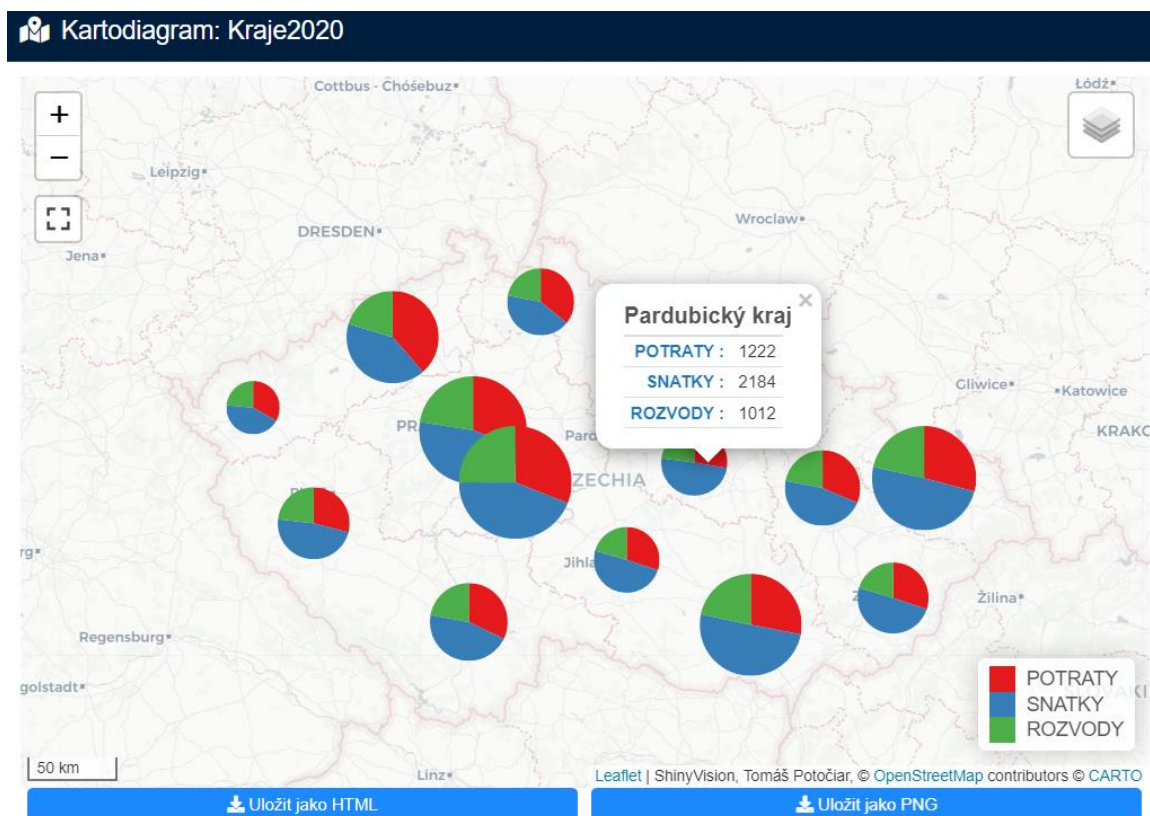

```

#Výpočet šířky kruhového diagramu
sirka_diagram <- reactive({
  if(input$typ_diagram == "Kruhový") {
    input$diagram_w * (sqrt(rowSums(data_kd[, input$param_kd])) /
      sqrt(rowSums(data_kd[, input$param_kd] %>% summarise_if(is.numeric, max))))
  } else {
    input$diagram_w
  }
})

```

Obr. 4.13 Kód pro výpočet šířky kruhového diagramu.

Funkce přidávající do mapy diagramy obsahuje vlastnost zobrazující v diagramech popisky hodnot parametru. Jelikož lze tuto vlastnost použít pouze při výběru jednoho parametru, byla vytvořena podmínka omezující vlastnost na základě počtu parametrů. Funkcionalita Leaflet mapy a její vykreslení probíhá na stejném principu jako u metody kartogramu (viz podkapitola 4.5.1).



Obr. 4.14 Kartodiagram krajů zobrazující potraty, sňatky a rozvody.

4.6 Statistické vizualizace

Neprostorové, neboli statistické vizualizace jsou v aplikaci zobrazovány interaktivními grafy. Statistické vizualizace byly rozděleny do tří sekcí na základě interpretace grafů – statistické grafy (histogram, boxplot), vzájemné vztahy (korelační diagram, korelační matice, paralelní osy) a časový vývoj (liniový graf). Grafy byly převážně tvořeny knihovnou *ggplot2* (Wickham et al. 2021) umožňující vizualizovat data a pro přidání interaktivity byla použita rozšiřující knihovna *ggiraph* (Gohel et al. 2022). Vstupní data pro statistické vizualizace jsou v podobě Excel listu.

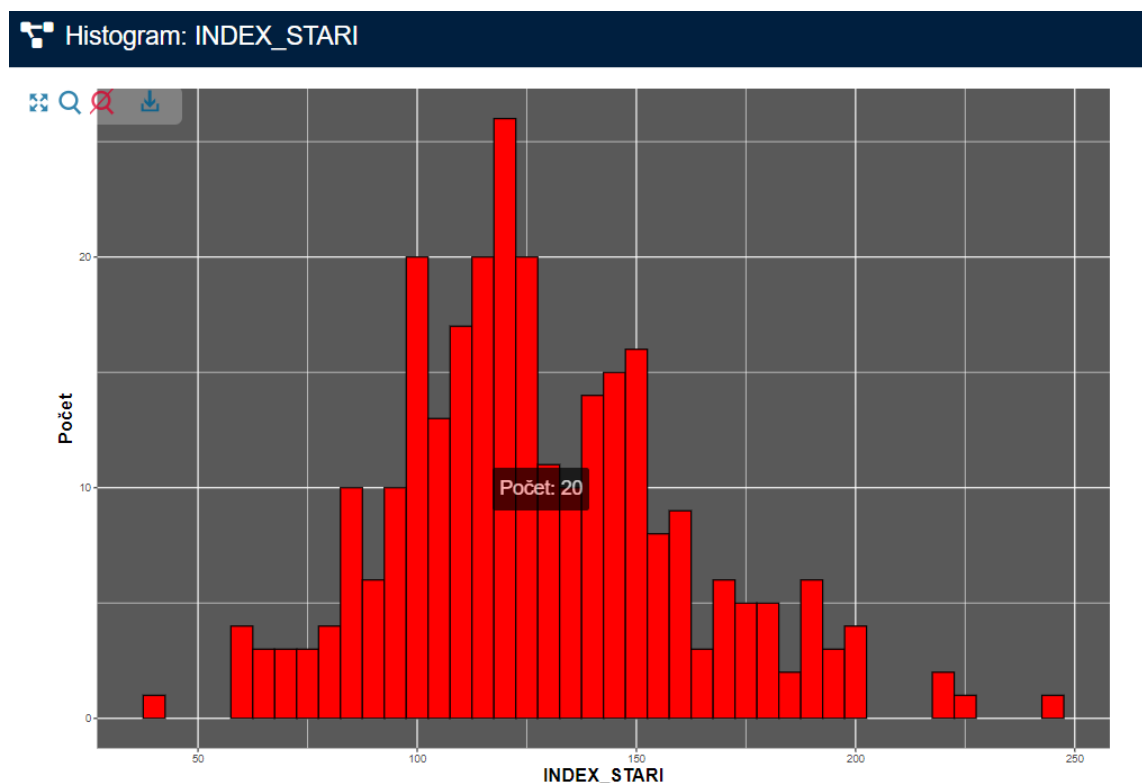
4.6.1 Histogram

Zobrazení interaktivního histogramu je založeno na knihovnách *ggplot2* a *ggiraph*. Po výběru dat je uživatelem zvolen parametr pro rozdělní hodnot na ose x. Uživatel ve vstupu hodnot zadá číselnou hodnotu určující mocnost sloupců, tzv. binů. Výšku sloupce určuje počet hodnot spadající do rozmezí sloupce.

V serverové části jsou do funkce `ggplot()` vloženy data a je určen parametr pro osu x. Interaktivitu zajišťuje vlastnost funkce `geom_histogram_interactive()` a to zejména zobrazení informace o počtu hodnot po najetí kurzorem na sloupec, tzv. tooltip. Ve funkci `ggplot()` jsou dále nastaveny popisky jednotlivých os a vzhled grafu podle uživatelského nastavení barev v boxu „další nastavení“.

Objekt vzniklý z funkce `ggplot()` je převeden do objektu funkce `girafe()` pro schopnost interagovat s grafem při vykreslení. Objektu je přiřazený list s nastavením, obsahující funkci pro konfiguraci panelu s nástroji, který umožňuje stahování grafů nebo pohybování se v grafu. List obsahuje také konfiguraci stylu tooltipu či vypnutí funkcionality vybírání dat.

Vykreslení grafu probíhá až po vykonání události, tedy po kliknutí na aktualizací tlačítko, pomocí renderovací funkce výstupního objektu `renderGirafe()`. Zároveň s grafem je vykreslen nadpis renderovací funkcí textu.



Obr. 4.15 Interaktivní histogram indexu stáří.

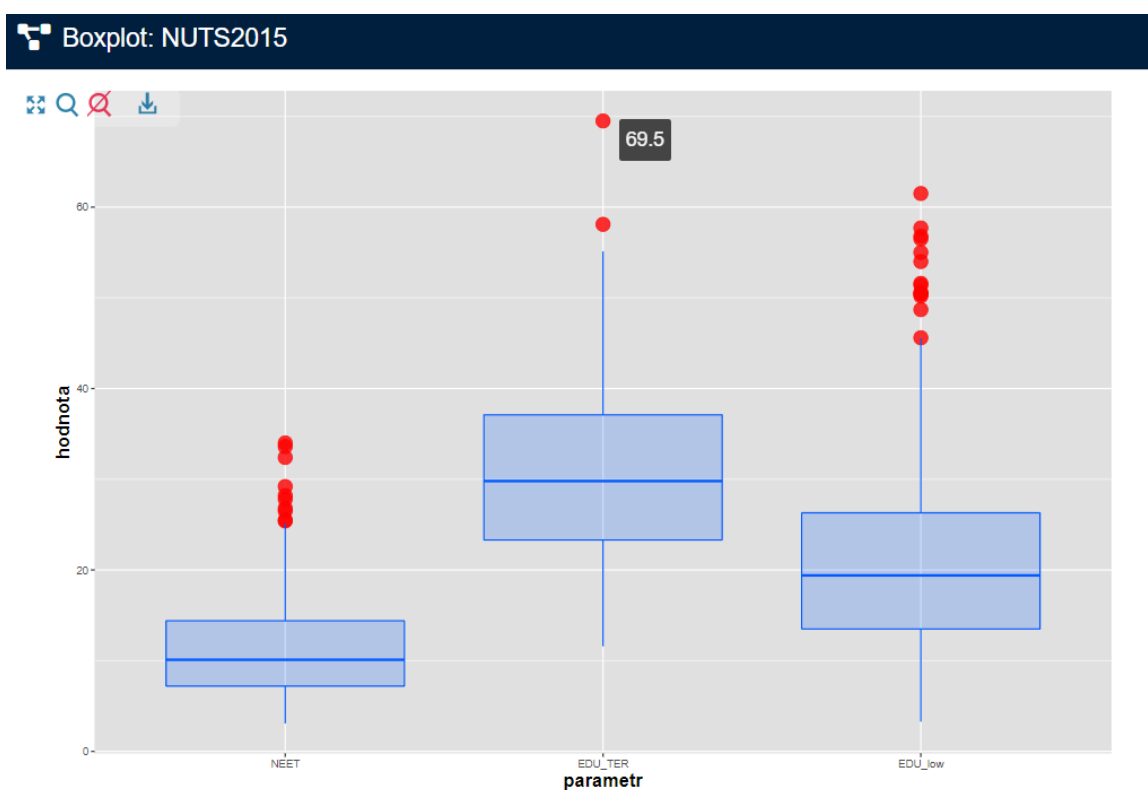
4.6.2 Boxplot

Při použití boxplot grafu má uživatel možnost výběru kvalitativní či kvantitativní funkcionality vizualizace. Pokud uživatel používá kvantitativní data, vybere si různý počet parametrů představující různý počet boxů („krabic“) na ose x. Osa y zobrazuje hodnoty parametrů. Při použití kvalitativních dat je uživateli umožněno vybrat na ose x

parametr, podle kterého lze data seskupit a na ose y je vybrán parametr porovnávající hodnoty mezi skupinovým parametrem.

Vstup dat do funkcí `ggplot()` a `girafe()` byl upraven podmínkou odlišující zpracování kvalitativních a kvantitativních dat. Pro kvantitativní data byla použita funkce `melt()` z knihovny *reshape2* (Wickham 2020) upravující strukturu dat do vhodné podoby. Funkce vytváří tabulku se sloupcem `variable` představujícím názvy vybraných parametrů na ose x a se sloupcem `value` obsahujícím číselné hodnoty parametrů.

Ve funkci `geom_boxplot_interactive()` jsou boxům a odlehlým hodnotám přiřazeny barvy a velikosti bodů dle výběru uživatele. Dále jsou nastaveny názvy os funkcí `labs()` a upraven vzhled grafu, tj. barva pozadí a styl popisek os. Převedený interaktivní objekt, umožňující při přejetí kurzorem na odlehlé hodnoty zobrazit informaci o hodnotě a graf uložit jako PNG obrázek, je vykreslen renderovací funkcí `renderGirafe()` do určeného boxu v uživatelském rozhraní.

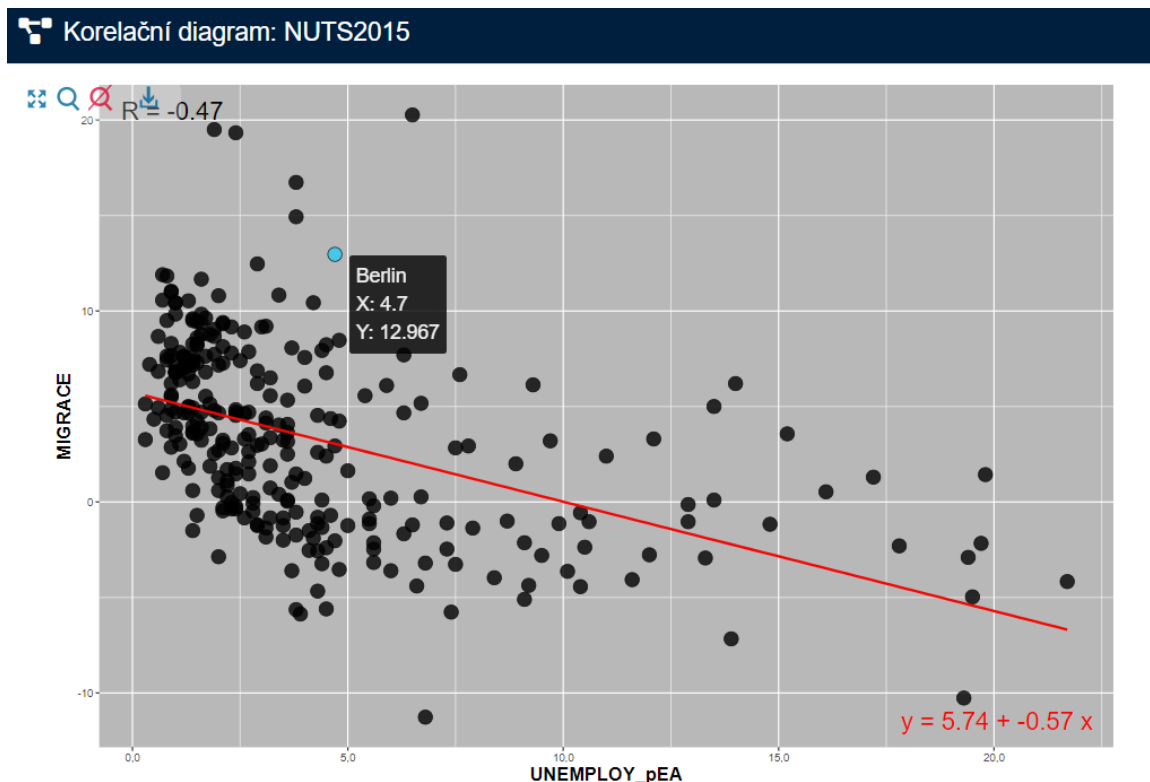


Obr. 4.16 Interaktivní boxplot vybraných parametrů

4.6.3 Korelační diagram

Funkcionalitu korelačního diagramu zajišťují knihovny *ggplot2* a *ggiraph*, spolu s výpočty korelačních koeficientů a lineárních rovnic. Po výběru dat jsou uživatelem zvoleny parametry os X a Y. Uživatel má možnost z výběru tří metod korelace – Pearsonova, Spearmanova, Kendallova. Korelační koeficient je vypočítán pomocí funkce `cor()` z knihovny *stats* (R Core Team 2022). Do grafu je umístěn popisek funkcí `annotate()` s vypočítanou hodnotou korelace. Uživateli je umožněno proložit vykreslený graf liniemi, přesněji lineární funkcí, lokální regresí nebo zobecněným generálním modelem (*generalized additive model*). Při výběru lineární funkce je umístěn v grafu popisek jejího výpočtu pomocí funkcí `coef()` a `lm()`.

Samotné body jsou vykresleny funkcí `geom_point_interactive()` v níž jsou nastavené atributy `id`, představující zvolený parametr pro název bodu jako identifikátor, a `tooltip`, zobrazující název, hodnoty na ose X a hodnoty na ose Y při najetí kurzorem na bod. Grafu jsou přiřazeny barevné hodnoty zadávající uživatel v sekci „další nastavení“ a upraveny popisky jednotlivých os. Objekt je převeden do funkce `girafe()` obsahující konfiguraci interaktivních prvků v grafu a nastavení barvy při „hover“ efektu (přejetí kurzorem na bod). Následně je graf spolu s nadpisem vykreslen do odpovídajícího boxu.



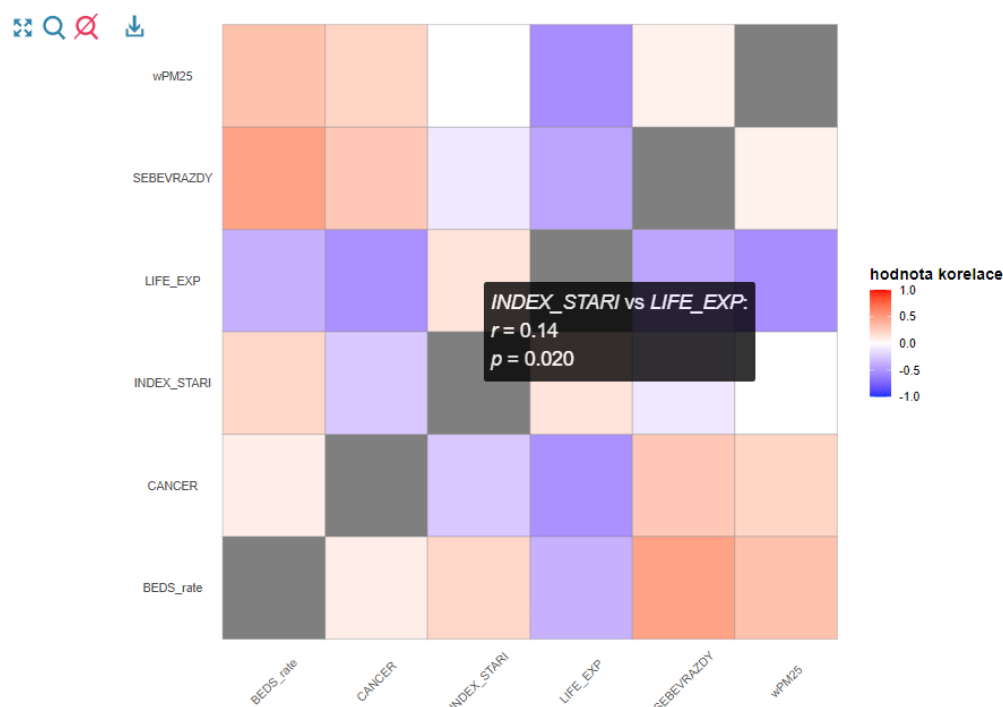
Obr. 4.17 Korelační diagram s výpočtem lineární funkce.

4.6.4 Korelační matice

Pro sestavení interaktivního grafu korelační matice byla využita knihovna `ggiraphExtra` (Moon 2020) rozšiřující knihovny `ggplot2` a `ggiraph`. Uživatelem jsou vybrány parametry načtené z nahraných dat vstupující do matice pro určení korelace. Lze vybírat ze tří metod korelace – Pearsonova, Spearmanova, Kendallova.

Oproti vizualizacím tvořeným ze základní knihovny `ggplot2`, vybrané prvky vstupují do funkce `ggCor()`, v níž jsou nastavené potřebné atributy, např. parametry, metoda korelace nebo barevná stupnice korelace určená ze záporné, nulové a kladné hodnoty. Interaktivitou grafu je zde při najetí na čtverec zobrazení informací ohledně porovnávaných parametrů, hodnoty korelace a hodnoty p určené pro další statistické operace. Také je uživateli umožněno graf uložit ve formě obrázku.

Grafu je nastaveno jednoduché téma pro přehlednost grafu pomocí funkce `theme_clean2()`. Legendě korelace byla upravená velikost a zaměněn popisek do vhodné podoby. Po kliknutí na aktualizací tlačítko je graf vykreslen renderovací funkcí do odpovídajícího boxu spolu s nadpisem.

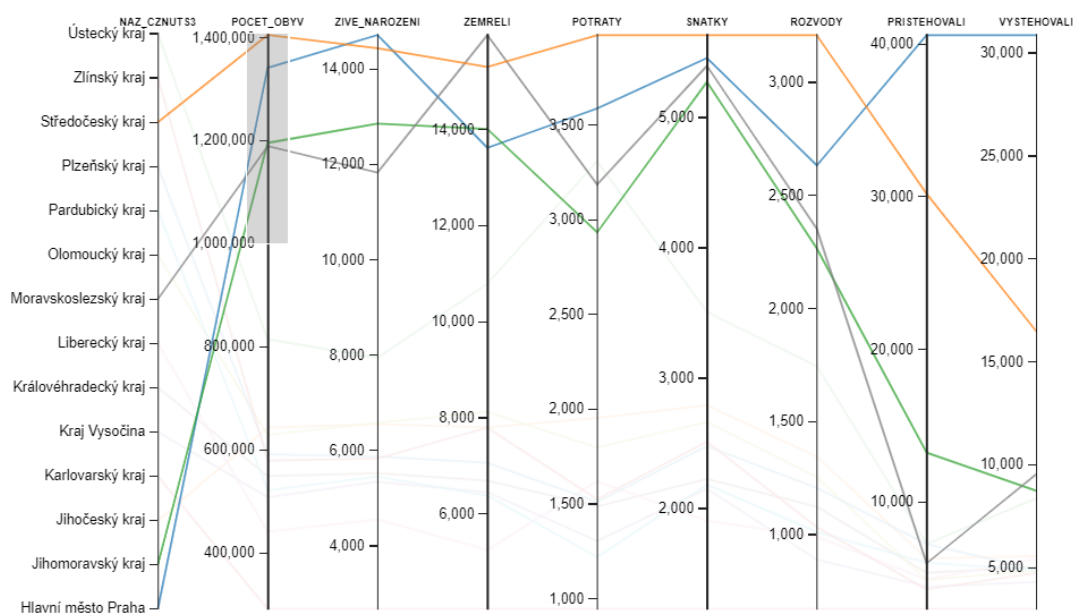


Obr. 4.18 Interaktivní graf korelační matice vybraných parametrů.

4.6.5 Paralelní osy

Graf paralelních os byl tvořen knihovnou *parcoords* (Bostock et al. 2019). Po výběru dat a parametrů pro porovnání na základě jednotlivých os je umožněno uživateli vybrat parametr pro seskupení dat a lépe se tak orientovat v grafu. V případě potřeby data neseskupovat, uživatel vybere možnost „NONE“.

Vybrané prvky vstupují do funkce `parcoords()` v níž jsou nastavené potřebné atributy, např. barevné schéma os s podmínkou při seskupení dat, interaktivita vybírání dat nebo logika vybírání dat. V grafu lze vybírat jednotlivé osy pomocí tzv. brush módu. Toho lze docílit zaznačením části osy ve vertikální podobě. Osy spadající do vyznačené části jsou zvýrazněné oproti osám mimo vyznačenou oblast a lze si tak filtrovat potřebné osy pro porovnání. Vertikálním osám lze také měnit pozici v grafu přesunutím osy na určité místo. Do této funkcionality spadá logika vybírání dat „zároveň“ a „nebo“ při vyznačení více než jedné osy. Graf je vykreslen funkcí `renderParcoords()` po kliknutí na aktualizací tlačítko odkazující se na reaktivní funkci s událostí dle referenčního identifikátoru.



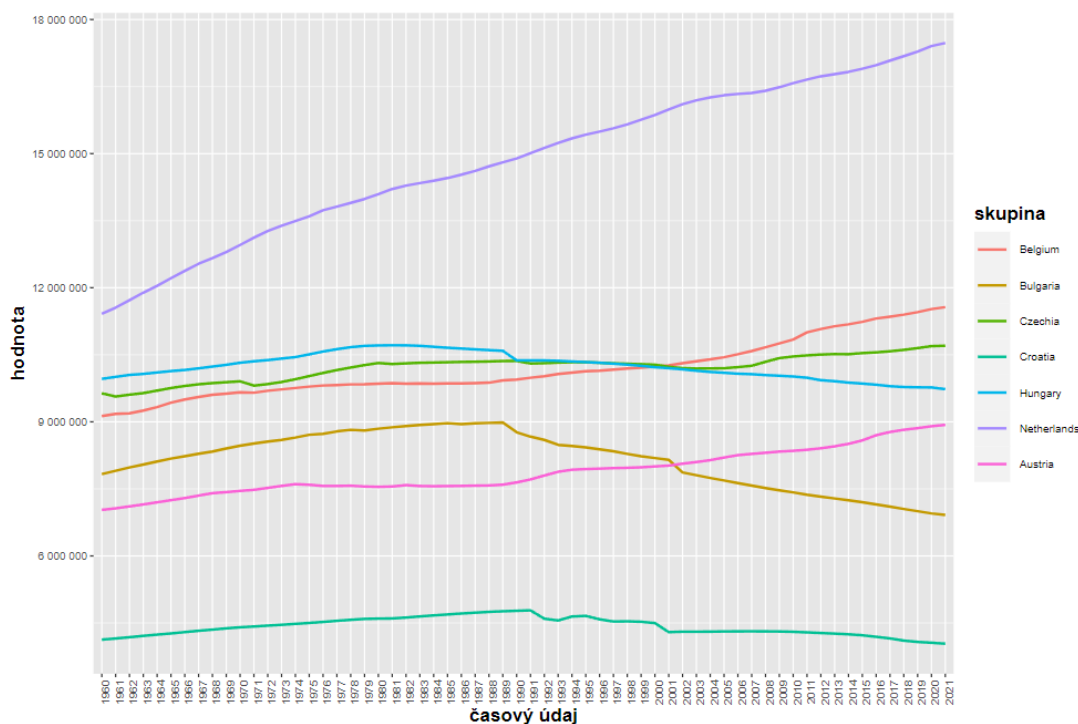
Obr. 4.19 Graf paralelních os – vybrané kraje nad 1 milion obyvatel.

4.6.6 Liniový graf

Pro statistickou vizualizaci časového vývoje byl použit liniový graf vytvořený za pomoci knihoven *ggplot2* a *ggiraph*. Uživatelem je zvolen časový parametr na ose X, reprezentující chronologický vývoj vybrané skupiny parametrů na ose Y. Časový parametr může být různého typu, např. hodiny, roky nebo datum a pro správné zobrazení linií musí být vhodně seřazen (od nejstaršího po nejnovější apod.).

Vybrané data jsou upraveny do vhodné struktury funkcí `melt()`, kde do atributu `id.vars` vstupuje časový parametr a do atributu `measure.vars` vstupuje skupina vybraných parametrů. Ve funkci `ggplot()` je pro osu X nastaven časový parametr, pro osu Y skupina parametrů a seskupení na základě názvů parametrů. Pomocí funkce `geom_line_interactive()` jsou hodnoty v grafu spojeny, liniím nastavena symbologie dle výběru uživatele a přidána informace o názvu parametru při přejetí na linii kurzorem.

V grafu jsou dále upraveny popisky os a legendy ve funkci `theme()` umožňující konfigurovat vzhled grafu. Vzniklý objekt vstupuje do funkce `girafe()`, ve které je nakonfigurována interaktivita grafu – při přejetí kurzorem na linii se ostatní linie zesvětlí pomocí funkce `opts_hover_inv()`. Nastavený graf je po aktivaci aktualizací tlačítka vykreslen renderovací funkcí `renderGirafe()` spolu s dynamickým nadpisem.



Obr. 4.20 Časový vývoj populace vybraných států Evropy.

4.7 Tabelární vizualizace

Vizualizace dat v podobě tabulky je k dispozici v sekci „prohlížení dat“. Vytvoření interaktivní tabulky s daty bylo realizováno přídatným balíčkem *DT* (Xie et al. 2022a). Balíček implementuje JavaScript knihovnu *DataTables* a její funkcionalitu do prostředí R.

Uživateli je umožněno vybrání předdefinovaných dat nebo nahrání vlastních dat ve formátu CSV, Shapefile, Excel List, JSON nebo GeoJSON. Po výběru jsou pomocí observeru a funkce `updateAwesomeCheckboxGroup()` zobrazeny parametry datové sady ve formátu přepínatelných tlačítek. Vybrané parametry vstupují do funkce `datatable()`. Ve funkci je přidána extenze v podobě akčních tlačítek, které jsou nakonfigurovány v atributu `options`. Tlačítko „Copy“ slouží ke zkopírování hodnot do schránky, tlačítko „Print“ umožňuje přejít do režimu pro tisk a následně vytisknout zobrazenou tabulku a pomocí tlačítka „Download“ lze stáhnout tabulku s hodnotami v podobě CSV, Excel nebo PDF.

Tabulce je nastavena maximální výška, aktivován horizontální posuvník při zobrazení většího počtu parametrů a pomocí JavaScriptu nastaveny vhodné barvy pozadí a hlavičky zobrazující názvy parametrů a umožňující seřadit hodnoty vzestupně či sestupně. Atribut `dom` umožňuje zobrazit či skrýt určité prvky tabulky pomocí jednopísmenových identifikátorů zapsaných kombinovaně ve formě textu. Například písmeno „f“ zobrazí filtrovací panel, písmeno „t“ je pro zobrazení tabulky, písmeno „p“ slouží pro zobrazení počtu stránek atd. (SpryMedia 2022). Interaktivní tabulka je vykreslena renderovací funkcí `renderDataTable()` do výstupu `dataTableOutput()` v uživatelském rozhraní.

Tabulka s daty												
											Search: <input type="text"/>	
											Copy	
											Print	
											Download	
NAZ_LAU1	POCET_OBYV	MUZI	ZENY	ZIVE_NAROZENI	ZEMRELI	POTRATY	SNATKY	ROZVODY	PRISTEHOVALI	VYSTEHOVALI		
50	Praha	1341370	657729	683641	14713	13621	3587	5455	2633	40582	30867	
4	Brno-město	381702	185325	196377	4501	4652	1019	1717	680	10322	9112	
43	Ostrava-město	318583	154744	163839	3225	4390	883	1427	616	4158	5816	
26	Karviná	244888	120016	124872	2245	3551	749	1147	505	2636	4875	
41	Olomouc	235662	114770	120892	2488	2803	650	993	494	2872	2588	
5	Brno-venkov	225631	111703	113928	2503	2358	513	986	489	4933	3278	
13	Frydek-Místek	214544	105863	108681	2189	2706	405	891	443	2711	2267	
9	České Budějovice	196276	96115	100161	2258	2160	639	906	445	3017	2416	
48	Pízeň-město	194807	95424	99383	1965	2247	492	803	382	4999	4177	
75	Zlín	191329	93627	97702	1891	2518	463	825	349	1995	2060	
51	Praha-východ	187302	92575	94727	2032	1542	485	706	486	7483	4212	
44	Pardubice	176150	87736	88414	1834	2020	384	763	347	3566	2684	
42	Opava	176062	86261	89801	1762	2193	460	797	336	1633	1661	
32	Liberec	175561	86248	89313	1872	1994	630	744	401	2954	2993	
27	Kladno	166733	81790	84943	1648	2027	577	673	361	3228	2487	
16	Hradec Králové	164378	80293	84085	1778	1946	364	639	300	2688	2368	

Show 25 entries

Previous 1 2 3 4 Next

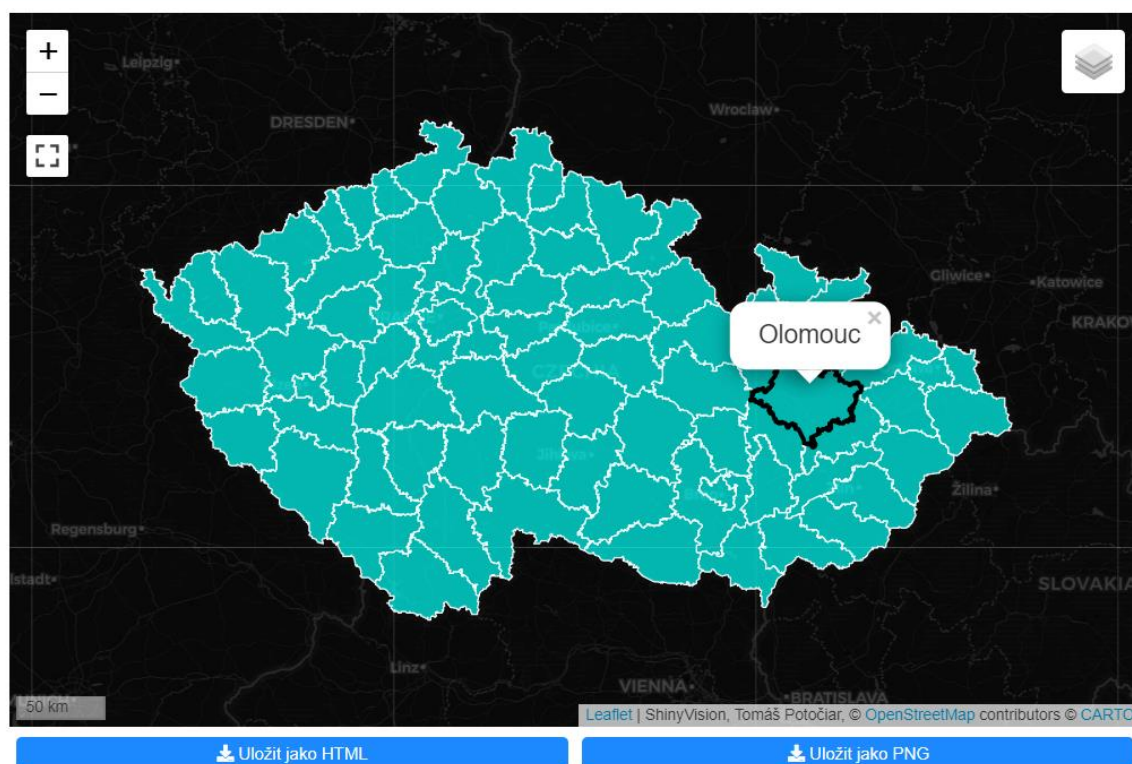
Obr. 4.21 Interaktivní tabulka zobrazující okresy ČR.

4.8 Prohlížení prostorových dat

Aplikace umožňuje prohlížet prostorová data pomocí Leaflet mapy. Lze zobrazit bodové, liniové i polygonové prvky. Funkcionalita slouží k čistě prohlížení dat, přičemž lze nastavit libovolnou symbologii a parametr zobrazující se ve vyskakovacím okně. Na základě výběru geometrie pomocí skupiny tlačítek s upravenými CSS styly je nutné pro zobrazení interaktivní mapy zvolit soubor Shapefile nebo GeoJSON, jež je přetransformován do souřadnicového systému WGS84. Díky transformaci a nastavení bounding boxu kolem zobrazených prvků lze prohlížet data, v jakémkoliv souřadnicovém systému, umístěné v mapě na jakémkoliv místě na Zemi. V reaktivní funkci s událostí jsou vytvořeny podmínky sloužící k odlišení vykonání bloku kódu pro určitý typ geometrie.

Zobrazení bodových prvků v mapě je umožněno funkcí `addCircleMarkers()`. Uživateli je umožněno nastavit barvu výplně, velikost bodu, šířku ohraničení, barvu ohraničení, průhlednost výplně a průhlednost ohraničení. Liniové prvky v mapě jsou zobrazovány pomocí funkce `addPolylines()` u nichž lze upravit barvu linie, šířku linie a průhlednost linie. Polygonové prvky jsou do mapy přidávány funkcí `addPolygons()` a úprava symbologie je stejná jako u bodových prvků bez možnosti změny velikosti prvku.

Všechny typy geometrie vstupují do vykreslovací funkce `renderLeaflet()` a uživateli je umožněno vykreslené mapy stahovat ve formátu interaktivní mapy HTML nebo PNG obrázku.



Obr. 4.22 Leaflet mapa okresů ČR s upravenou symbologií.

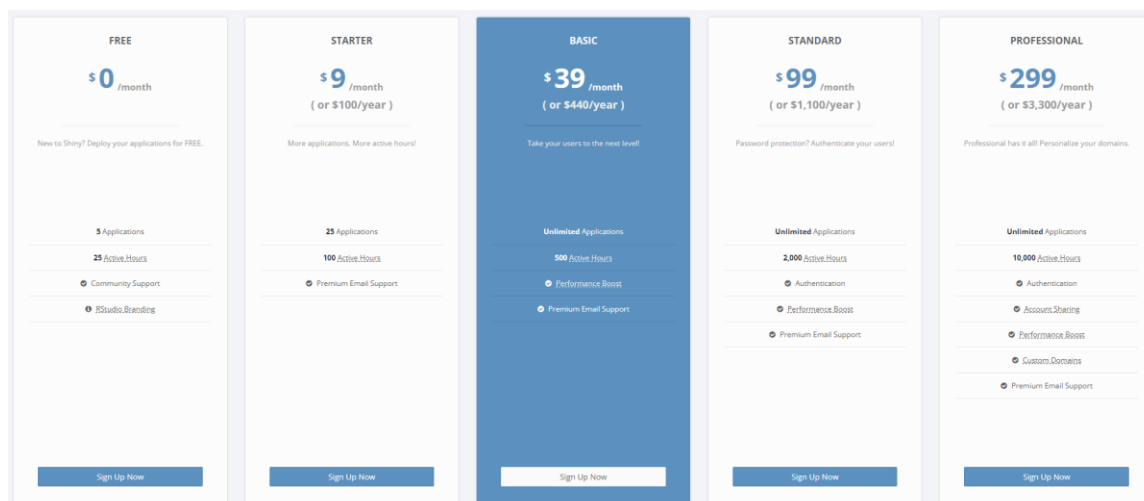
5 LIMITY A OMEZENÍ

Při vytváření aplikace byly zkoumány limity a omezení frameworku Shiny z různých hledisek a oblastí. Vyhodnocení limitů a omezení probíhalo na základě zkušeností, úvahy a potřeb tvůrce aplikace. Nevyřešené časově náročné problémy byly zařazeny mezi limity a omezení s možností budoucího řešení.

5.1 Licenční omezení shinyapps.io

Aplikace ShinyVision je publikována prostřednictvím cloudové platformy *shinyapps.io* vyvíjené společností RStudio. Pro demonstraci možností vizualizace knihovny *Shiny* byl využit licenční plán „FREE“ umožňující publikovat pod jedním účtem maximálně pět aplikací zcela zdarma. Při používání tohoto plánu je povoleno maximálně 25 aktivních hodin za období (měsíc). Jedná se o dobu, kdy je aplikace využívána uživateli, tj. aplikace je aktivní. Při neaktivitě je aplikace přeprnuta do stavu „neaktivní“ a doba aktivních hodin není přičítána. Jestliže dojde k překročení limitu aktivních hodin, aplikace nejsou dostupné uživatelům do dalšího platebního cyklu (shinyapps.io team 2021).

Další licenční omezení je ve velikosti alokované operační paměti aplikacím. S licenčním plánem „FREE“ lze nastavit maximální velikost paměti na 1024 MB. Toto omezení se promítlo i ve funkcionalitě aplikace ShinyVision. Při pokusu o uložení vykreslené mapy byl překročen limit využití operační paměti, aplikace vyvolá chybovou hlášku a uživatele odpojí od serveru. Z tohoto důvodu byla funkcionalita ukládání Leaflet map zakázána pomocí funkce `disabled()` z knihovny *shinyjs* (Attali 2021) odkazována na akční tlačítko.



Obr. 5.1 Seznam licenčních plánů platformy shinyapps.io (RStudio).

5.2 Velikost importovaných dat

V aplikaci ShinyVision byla testována maximální velikost importovaných dat v závislosti na rychlosti načtení dat a výpočetní rychlosti aplikace. Pro testování byly využity obce ČR z datové sady *ArcČR 3.3* ve formátu Shapefile s celkovou velikostí přibližně 90 MB.

Data na lokálním zařízení byly načteny v relativně dobrém času (přibližně 10 sekund), ovšem v publikované aplikaci je rychlost načítání dat odvíjena od rychlosti uploadu internetu uživatele, takže doba byla o poznání delší. Problém nastal při vykreslení obcí v mapě. Bylo téměř nemožné se pohybovat v mapě jak na lokálním zařízení, tak i v publikované verzi z důvodu výpočetní rychlosti aplikace a velkého počtu lomových bodů.

Limit velikosti importovaných dat byl nastaven na 30 MB pomocí globálního nastavení aplikace `options(shiny.maxRequestSize = 30*1024^2)`. Takle velikost byla dostačující vzhledem k času stráveného čekáním na nahrání dat a výpočetní rychlosti aplikace umožňující bezproblémově ovládat funkcionalitu Leaflet mapy.

5.3 Datové formáty

V aplikaci ShinyVision byly použity datové formáty umožňující importovat data do aplikace a následně je upravovat a zpracovávat. Pro prostorové vizualizace slouží primárně formáty Shapefile, JSON a GeoJSON, u metody kartodiagramu je používán list Excelu. Pro statistické a tabelární vizualizace lze využít formátů Excel listu (.xlsx, .xls) nebo formátu CSV s možností výběru jakéhokoliv oddělovače.

Jedním z limitů formátu Shapefile je nutnost výběru všech povinných souborů s příponami .shp, .shx, .dbf a .prj. Při vynechání jednoho souboru je import dat chybně proveden. Z toho důvodu musela být vytvořena funkce sjednocující vybrané povinné soubory do jednoho SHP souboru (Obr. 4.6).

Ukládat vykreslené mapy lze ve formátech interaktivní webové aplikace HTML nebo statického obrázku PNG. Grafy jsou ukládány pouze ve statickém formátu PNG. U tabelární vizualizace lze zobrazené data uložit ve formátu Excel listu, CSV nebo PDF. Omezením PDF souboru je velikost A4 a tím možnost zobrazit jen určitý počet parametrů.

5.4 Struktura importovaných dat

Při importování dat uživatelem je nutné zachovat přesně definované struktury pro správné zobrazení grafu či mapy. Při odlišné struktuře může dojít k chybnému vykreslení vizualizace a následné chybné interpretaci výsledků. V lepším případě je zobrazena chybová hláška s popisem chyby a zabránění tak chybné interpretaci. Informace o struktuře dat jsou zobrazeny po kliknutí na ikonu otazníku ve formátu vzorové tabulky a informativního textu.

5.5 Možnosti vizualizace

Pomocí knihovny Shiny lze vytvořit mnoho vizualizací mnoha různými způsoby. Limitace z tohoto hlediska je spíše v možnostech prostorové vizualizace, jelikož jazyk R je zaměřený z větší části na práci se statistickými daty a umožňuje tvořit širokou škálu vizualizací díky velkému počtu dostupných a propracovaných knihoven, zejména knihovny *ggplot2*.

Interaktivní prostorové vizualizace lze tvořit pomocí knihovny *Leaflet* nebo na základě přístupového tokenu z webového portálu Mapbox mohou být využity knihovny *mapdeck* (Cooley 2020) a *mapboxer* (Kueth 2020). Existují i další knihovny, ale jsou spíše zaměřené pro tvorbu statických map bez možností interakce nebo propojení s Shiny.

Omezení možností vizualizace je částečně způsobeno používáním vlastním výběrem dat, tj. musí být brán ohled na jakékoliv vstupní data, čímž je výrazně zvýšená náročnost tvorby vizualizace, někdy i kvůli tomuhle problému dochází k porušení korektnosti vizualizace (viz kapitola 7), jenž by v případě přesně určených vstupních dat nemuselo dojít.

5.6 Interaktivita

U některých vizualizací dochází k poklesu interaktivity z pohledu požadavků tvůrce aplikace a limitů knihovny. U statistických vizualizací sestrojených pomocí knihovny *ggplot2* nelze uložit interaktivní verzi grafu, ale pouze jako statický obrázek. U grafu paralelních os funkcionální stahování grafu není vůbec dostupná. V tomhle grafu byla řešena také vlastnost zobrazení informací po přejetí kurzorem na určitou osu, ale ve výsledku byla zjištěna nepodpora této funkcionality a zařazena mezi limity knihovny.

U časového vývoje pomocí liniového grafu je při přejetí kurzorem na linii zobrazován pouze název linie, přičemž bylo uvažováno i o zobrazení aktuální hodnoty parametru. Podobné omezení je u boxplot grafu, kde lze zobrazit přesnou hodnotu pouze odlehlých bodů po přejetím kurzorem, nikoliv u hodnoty mediánu a dalších prvků.

6 VÝSLEDKY

Hlavním výsledkem bakalářské práce je aplikace ShinyVision zaměřená na vizualizaci dat. Aplikace byla napsána v jazyku R a hlavní roli tvořila knihovna *Shiny* (Chang et al. 2021b). Aplikace byla umístěna do interaktivního layoutu (Obr. 4.3) a jednotlivým vizualizacím bylo vytvořeno strukturované uživatelské rozhraní (Obr. 4.4). V rámci aplikace jsou rozlišovány prostorové, statistické a tabelární vizualizace.

Výstupy prostorových vizualizací byly řešeny pomocí knihovny *Leaflet* (Cheng et al. 2022), mezi něž patří metoda kartogramu (Obr. 4.12), metoda kartodiagramu (Obr. 4.14) a prohlížení prostorových dat pomocí bodových, liniových či prostorových znaků (Obr. 4.22). Tyto vizualizace obsahují interaktivní prvky a možnost ukládat vykreslené mapy ve zvolených formátech.

Většina statistických vizualizací byla řešena knihovnou *ggplot2* (Wickham et al. 2021). Statistické vizualizace byly rozděleny do tří sekcí – statistické grafy, do nichž patří histogram (Obr. 4.15) a boxplot (Obr. 4.16), vzájemné vztahy obsahující korelační diagram (Obr. 4.17), korelační matice (Obr. 4.18), paralelní osy (Obr. 4.19) a pro sekci časový vývoj byl použit liniový graf (Obr. 4.20). Statistické vizualizace obsahují interaktivní prvky vytvořené pomocí knihovny *ggiraph* (Gohel et al. 2022) pokud se jedná o graf z knihovny *ggplot2*.

Tabelární vizualizace slouží pro prohlížení zdrojových dat. Pro vytvoření interaktivní tabulky s daty byla použita JavaScript knihovna *DT* implementována do R (Xie et al. 2022a). Uživatel je schopen zobrazit, vyfiltrovat a uložit vybrané data v několika formátech (Obr 4.21).

Veškeré vizualizace pracují s vytvořenou funkcionalitou výběru dat (kapitola 4.4) a s principem reaktivních funkcí pro aktualizaci parametrů z vybrané datové sady (Obr. 4.10). Uživatelům je pak umožněno nastavení vizualizace dle potřeby a vždy po kliknutí na aktualizací tlačítko je graf, tabulka či mapa vykreslena.

Na závěr práce byla určena kritéria limitů a omezení knihovny Shiny a věci s tvorbou aplikace ShinyVision spojených. Maximální velikost importovaných dat je omezena na 30 MB, chybí některé prvky grafů a map pro úplnou korektnost, licenční omezení *shinyapps.io* pro publikování aplikací Shiny je dosti výrazné – kvůli použití licenčního plánu „FREE“ a překračování limitu alokované paměti musela být omezena funkcionalita ukládání map. Naopak omezení datových formátů je minimální spolu s možnostmi vizualizace a implementovanou interaktivitou.

Aplikace ShinyVision je zveřejněná v rámci cloudové platformy *shinyapps.io* na tomto odkaze: **<https://tomaspotociar.shinyapps.io/shinyvision>**

7 DISKUZE

Při tvorbě aplikace nastalo několik problémů, jejichž vyřešení bylo velmi časově náročné, nebo dokonce nemožné na základě limitací knihoven apod. Nicméně problémy nejsou příliš razantní a potenciál aplikace pro další využití je značný.

Problémy

Jedním z nevyřešených problémů je uživatelské nastavení vlastních intervalů u metody kartogramu. Bylo zamýšleno tak, že si uživatel zvolí počet intervalů stupnice a podle toho mu bude zobrazena možnost si nastavit vlastní hodnoty intervalů. Tím by byla i vyřešena nekorektnost stupnic, jenž je momentálně nastavena na základě automatického rozložení hodnot na stejné intervaly. Problém se nepodařilo vyřešit, avšak jeho vyřešení by přineslo značné výhody. Nejedná se pravděpodobně o limitaci knihovny, nýbrž o schopnost vyřešení problému programátorsky.

Dalším problémem odvíjejícím se od problému s intervaly je nekorektnost barevných stupnic na základě kartografických pravidel. Při vyřešení problému s uživatelskou konfigurací intervalů by mohlo řešení být implementováno i na vlastní konfiguraci barevné stupnice. Nyní má uživatel možnost vybrat si pouze z několika předdefinovaných stupnic a mohou nastat situace, kdy při zobrazení určitých dat žádná ze stupnic nemusí být vhodná k sestrojení správného kartogramu.

U metody kartodiagramu je potřeba vždy mít údaje o poloze k umístění diagramu v mapě v podobě souřadnic X a Y ve tvaru systému WGS84. Toto může být někdy problémem, jelikož k vygenerování souřadnic je většinou potřeba jiného GIS softwaru. Problém by šel vyřešit vytvořením „generátoru souřadnic bodů“, kdy by si uživatel naklikal v mapě pozice diagramů na základě počtu záznamů v datové sadě a nemusel by souřadnice generovat mimo aplikaci Shiny. Dále má uživatel možnost si vybrat pouze z dvou typů diagramu, kruhového a sloupcového, a nelze zobrazit stupnici gradaci kruhových diagramů. Zde se jedná o limitace knihovny *leaflet.minicharts* (Bachelier et al. 2021).

U statistických vizualizací nastal problém u zobrazení dodatečných informací při přejetí kurzorem na prvek, zejména u paralelních os, liniového grafu a boxplotu. Z části je to zapříčiněno limitací knihovny a z části univerzálností dat, tj. uživatel má možnost vybrat si vlastní data a proto se ve funkcích musí brát ohled na různé případy způsobující chyby při vykreslování.

Publikované aplikaci, skrze cloudovou platformu *shinyapps.io*, musela být omezena funkcionalita ukládání map z důvodu licenčního omezení této platformy. Při pokusu o uložení hodnota alokované paměti překročila maximální limit a uživatel byl od aplikace odpojen. Problém by byl vyřešen použitím licenčního plánu „BASIC“ nebo použitím jiného způsobu ukládání map využívajícím méně paměti.

Do problémů by se také dala zařadit rozsáhlost kódu. Aplikace ShinyVision má přes 3800 řádků a i přes odsazování a komentování bloků kódu byla orientace náročná.

Potenciál aplikace

Aplikace ShinyVision je vhodná pro komplexní zkoumání datových sad, jednak v prostorové složce, tak i ve statistické a tabelární. Využití si najde hlavně v akademické sféře a u datových analytiků, jejichž práce je založená na důkladném poznání dat. Aplikace má také potenciál pro pokračování tvorby a vylepšení funkčnosti nedorozřešených problémů. Například knihovna Leaflet umožňuje spolu s Shiny dynamické zobrazování prvků v mapě pomocí tzv. proxy, jenž by posunula úroveň vizualizací o něco výše.

Potenciál je mnohem větší u aplikací s konkrétním řešeným problémem. Aplikace ShinyVision je zaměřená spíše na obecnou vizualizaci dat, tj. po výběru dat lze zobrazit jakýkoliv parametr nastavený dle potřeby. Náročnost řešení obecné aplikace je mnohonásobně vyšší než u aplikace s konkrétním řešeným problémem.

Ukázka dalších aplikací

- *Shanghai - Airbnb* – <https://tomasmichal.shinyapps.io/Shanghai>
- *Mapování strachu v Olomouci* – <https://ptmapping.shinyapps.io/PTMapping>

Aplikace vznikly v rámci studia na katedře geoinformatiky v Olomouci. Jedná se o výstupy k projektu a praktickým úkolům.

Více ukázkových aplikací lze najít v oficiální webové galerii RStudio na odkaze <https://shiny.rstudio.com/gallery>.

8 ZÁVĚR

Cílem bakalářské práce bylo demonstrovat možnosti publikace prostorových a statistických dat pomocí knihovny *Shiny* umožňující v prostředí R vytvářet interaktivní webové aplikace. Dále bylo nutné sestavit ukázkovou datovou sadu obsahující prostorovou a bohatou atributovou složku. Následně jsou stanoveny kritéria pro vyhodnocení limitace knihovny.

Výsledkem práce je aplikace nesoucí název ShinyVision umožňující zobrazovat data pomocí prostorových, statistických a tabelárních vizualizací. Uživateli je umožněno vybrat si předdefinované sestavené datové sady *ukazatelé kvality života* pro NUTS2, *demografické údaje ČR* v podrobnosti na kraje, okresy a ORP nebo *populace vybraných států Evropy*. V opačném případě si uživatel může nahrát data vlastní. Po výběru dat a nastavení vstupních hodnot je interaktivní vizualizace vykreslena v odpovídající sekci (boxu). Prostorové vizualizace obsahují metodu kartogramu, kartodiagramu a metodu bodových, liniových a plošných znaků. Statistické vizualizace jsou demonstrovány na histogramu, boxplotu, korelačním diagramu, korelační matici, paralelních osách a liniovém grafu. Pro prohlížení zdrojových dat slouží tabelární vizualizace. Grafy, mapy a tabulky je možné ukládat v různých formátech.

Na základě tvorby a zkoumání řešené problematiky byly stanoveny kritéria pro vyhodnocení limitací z hlediska licenčního omezení shinyapps.io, maximální velikosti importovaných dat, omezení datových formátů a z hlediska možností vizualizací spolu s možností interaktivity. Aplikace je zveřejněna formou cloudové platformy shinyapps.io a dostupná na odkaze <https://tomaspotociar.shinyapps.io/shinyvision/>.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

APPSILON, 2022. *Python Dash vs. R Shiny – Which To Choose in 2022 and Beyond* [online] [vid. 2022-03-21]. Dostupné z: <https://appsilon.com/dash-vs-shiny/>

ATKINS, Aron, Jonathan MCPHERSON, JJ ALLAIRE a RSTUDIO, 2021. *rsconnect: Deployment Interface for R Markdown Documents and Shiny Applications* [online] [vid. 2021-10-15]. Dostupné z: <https://cran.r-project.org/web/packages/rsconnect/index.html>

ATTALI, Dean, 2021. *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds* [online] [vid. 2022-01-26]. Dostupné z: <https://cran.r-project.org/package=shinyjs>

ATTALI, Dean a David GRISWOLD, 2021. *colourpicker: A Colour Picker Tool for Shiny and for Selecting Colours in Plots* [online] [vid. 2021-10-21]. Dostupné z: <https://cran.r-project.org/package=colourpicker>

BACHELIER, Veronique, Jalal-Edine ZAWAM, Benoit THIEURMEL, Francois GUILLEM a RTE, 2021. *leaflet.minicharts: Mini Charts for Interactive Maps* [online] [vid. 2021-11-10]. Dostupné z: <https://cran.r-project.org/package=leaflet.minicharts>

BOSTOCK, Mike, Kai CHANG, Xing YUN, Kenton RUSSELL, Anobel ODISHO a Mark ALBRECHT, 2019. *parcoords: „Htmlwidget“ for „d3.js“ Parallel Coordinates Chart* [online] [vid. 2022-01-15]. Dostupné z: <https://cran.r-project.org/package=parcoords>

BREWER, Cynthia, 2013. *COLORBREWER 2.0 color advice for cartography* [online] [vid. 2021-10-21]. Dostupné z: <https://colorbrewer2.org/>

CHANG, Winston, Barbara BORGES RIBEIRO a RSTUDIO, 2021a. *shinydashboard: Create Dashboards with „Shiny“* [online] [vid. 2021-09-20]. Dostupné z: <https://cran.r-project.org/package=shinydashboard>

CHANG, Winston, Joe CHENG, JJ ALLAIRE, Carson SIEVERT a Barret SCHLOERKE, 2021b. *shiny: Web Application Framework for R* [online] [vid. 2021-09-20]. Dostupné z: <https://cran.r-project.org/package=shiny>

CHENG, Joe, Bhaskar KARAMBELKAR, Yihui XIE, Hadley WICKHAM, Vladimir AGAFONKIN a Kenton RUSSELL, 2022. *leaflet: Create Interactive Web Maps with the JavaScript „Leaflet“ Library* [online] [vid. 2021-09-21]. Dostupné z: <https://cran.r-project.org/package=leaflet>

CHENG, Joe, Carson SIEVERT, Barret SCHLOERKE, Winston CHANG, Yihui XIE, Jeff ALLEN a RSTUDIO, 2021. *htmltools: Tools for HTML* [online] [vid. 2022-01-17]. Dostupné z: <https://cran.r-project.org/package=htmltools>

CLOVERDX, 2021. *How much data will the world produce in 2021?* [online] [vid. 2022-03-15]. Dostupné z: <https://www.cloverdx.com/blog/how-much-data-will-the-world-produce-in-2021>

COOLEY, David, 2020. *mapdeck: Interactive Maps Using „Mapbox GL JS“ and „Deck.gl“* [online] [vid. 2022-04-15]. Dostupné z: <https://cran.r-project.org/package=mapdeck>

FONTICONS, 2022. *Font Awesome* [online] [vid. 2021-09-21]. Dostupné z: <https://fontawesome.com/>

FREE SOFTWARE FOUNDATION, 2022. *GNU Operating Systems* [online] [vid. 2022-03-15]. Dostupné z: <https://www.gnu.org/>

GATTO, Malu A C, 2015. Making research useful: Current challenges and good practices in data visualisation.

GOHEL, David, Panagiotis SKINTZOS, Mike BOSTOCK, Speros KOKENES, Eric SHULL, Lee THOMASON a Lee THOMASON, 2022. *ggiraph: Make „ggplot2“ Graphics Interactive* [online] [vid. 2021-12-02]. Dostupné z: <https://cran.r-project.org/package=ggiraph>

GRANJON, David, RINTERFACE, ALMASAEED STUDIO, Guang YANG, Winston CHANG a Victor PERRIER, 2021. *shinydashboardPlus: Add More „AdminLTE2“ Components to „shinydashboard“* [online] [vid. 2021-11-23]. Dostupné z: <https://cran.r-project.org/package=shinydashboardPlus>

GUNELIUS, Susan, 2014. *The Data Explosion in 2014 Minute by Minute - Infographic* [online] [vid. 2022-03-15]. Dostupné z: <https://www.newstex.com/blog/the-data-explosion-in-2014-minute-by-minute-infographic>

HART, Michele, Michael BLYTHE, Alex BUCK, Mona BERDUGO, Kesem SHARABI, Maggie SPARKMAN, Kent SHARKEY a V-CHMCCL, 2022. *What is Power BI?* [online] [vid. 2022-03-15]. Dostupné z: <https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>

HOLTZ, Yan, 2018a. Boxplot. *data-to-viz.com* [online] [vid. 2022-03-21]. Dostupné z: <https://www.data-to-viz.com/caveat/boxplot.html>

HOLTZ, Yan, 2018b. Correlogram. *data-to-viz.com* [online] [vid. 2022-03-21]. Dostupné z: <https://www.data-to-viz.com/graph/correlogram.html>

HOLTZ, Yan, 2018c. Histogram. *data-to-viz.com* [online] [vid. 2022-03-21]. Dostupné z: <https://www.data-to-viz.com/graph/histogram.html>

HOLTZ, Yan, 2018d. Line Chart. *data-to-viz.com* [online] [vid. 2022-03-21]. Dostupné z: <https://www.data-to-viz.com/graph/line.html>

JACCAR, 2019. *Download leaflet map as a png when shiny app is deployed* [online] [vid. 2022-03-10]. Dostupné z: <https://stackoverflow.com/questions/59519438/download-leaflet-map-as-a-png-when-shiny-app-is-deployed>

KUETHE, Stefan, 2020. *mapboxer: An R Interface to „Mapbox GL JS“* [online] [vid. 2022-03-15]. Dostupné z: <https://cran.r-project.org/package=mapboxer>

MOON, Keon-Woong, 2020. *ggiraphExtra: Make Interactive „ggplot2“. Extension to „ggplot2“ and „ggiraph“* [online] [vid. 2022-02-10]. Dostupné z: <https://cran.r-project.org/package=ggiraphExtra>

MORAGA, Paula, 2019. 15.9.3 Uploading shapefile in server(). In: *Geospatial Health Data, Modeling and Visualization with R-INLA and Shiny* [online]. B.m.: Chapman and Hall/CRC. ISBN 9780367357955. Dostupné z: <https://www.paulamoraga.com/book-geospatial/sec-shinyexample.html#uploading-shapefile-in-server>

NEUWIRTH, Erich, 2022. *RColorBrewer: ColorBrewer Palettes* [online] [vid. 2021-10-04]. Dostupné z: <https://cran.r-project.org/package=RColorBrewer>

PEBESMA, Edzer, Roger BIVAND, Etienne RACINE a Michael SUMNER, 2022. *sf: Simple Features for R* [online] [vid. 2021-10-10]. Dostupné z: <https://cran.r-project.org/package=sf>

PERRIER, Victor, Fanny MEYER, David GRANJON, Ian FELLOWS, Wil DAVIS a Spencer MATTHEWS, 2022. *shinyWidgets: Custom Inputs Widgets for Shiny* [online] [vid. 2021-11-10]. Dostupné z: <https://cran.r-project.org/package=shinyWidgets>

R CORE TEAM, 2022. *The R Stats Package* [online] [vid. 2022-02-15]. Dostupné z: <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html>

RESIG, John a THE JQUERY TEAM, 2021. *jQuery* [online] [vid. 2022-03-20]. Dostupné z: <https://jquery.com/>

RIBECCA, Severino, 2019a. *Parallel Coordinates Plot* [online] [vid. 2022-03-16]. Dostupné z: https://datavizcatalogue.com/methods/parallel_coordinates.html

RIBECCA, Severino, 2019b. *Scatterplot* [online] [vid. 2022-03-16]. Dostupné z: <https://datavizcatalogue.com/methods/scatterplot.html>

RSTUDIO, 2022a. *RStudio* [online] [vid. 2021-09-20]. Dostupné z: <https://www.rstudio.com/products/rstudio/>

RSTUDIO, 2022b. *RStudio Documentation, RStudio Connect* [online] [vid. 2021-09-20]. Dostupné z: <https://docs.rstudio.com/rsc/>

SEELMANN, Stefan a GITHUB CONTRIBUTORS, 2021. *Leaflet-providers* [online] [vid. 2021-10-15]. Dostupné z: <https://github.com/leaflet-extras/leaflet-providers>

SHINY FROM RSTUDIO, 2020. *Share your apps, Lesson 7* [online] [vid. 2022-03-20]. Dostupné z: <https://shiny.rstudio.com/tutorial/written-tutorial/lesson7/>

SHINYAPPS.IO TEAM, 2021. *shinyapps.io user guide* [online] [vid. 2021-10-08]. Dostupné z: <https://docs.rstudio.com/shinyapps.io/>

SPRYMEDIA, 2022. *DataTables - dom* [online] [vid. 2021-11-27]. Dostupné z: <https://datatables.net/reference/option/dom>

STREAMLIT INC., 2022. *Streamlit documentation* [online] [vid. 2022-03-15]. Dostupné z: <https://docs.streamlit.io/>

TECHTARGET CONTRIBUTOR, 2019. *Web application (Web app)* [online] [vid. 2022-03-14]. Dostupné z: <https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>

THE R FOUNDATION, 2022. *What is R?* [online] [vid. 2022-03-14]. Dostupné z: <https://www.r-project.org/about.html>

VOŽENÍLEK, Vít a Jaromír KAŇOK, 2011. *Metody tematické kartografie: vizualizace prostorových jevů*. Olomouc: Univerzita Palackého v Olomouci. ISBN 978-80-244-2790-4.

WICKHAM, Hadley, 2020. *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package* [online] [vid. 2022-02-02]. Dostupné z: <https://cran.r-project.org/package=reshape2>

WICKHAM, Hadley, 2021a. *Mastering Shiny: build interactive apps, reports, and dashboards powered by R* [online]. ISBN 9781492047384 1492047384. Dostupné z: <https://mastering-shiny.org/>

WICKHAM, Hadley, 2021b. Reactive building blocks. In: *Mastering Shiny: build interactive apps, reports, and dashboards powered by R* [online]. ISBN 9781492047384 1492047384. Dostupné z: <https://mastering-shiny.org/>

WICKHAM, Hadley, Jennifer BRYAN a RSTUDIO, 2022. *readxl: Read Excel Files* [online] [vid. 2021-10-28]. Dostupné z: <https://cran.r-project.org/package=readxl>

WICKHAM, Hadley, Winston CHANG, Lionel HENRY, Thomas Lin PEDERSEN a RSTUDIO, 2021. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics* [online] [vid. 2021-11-07]. Dostupné z: <https://cran.r-project.org/package=ggplot2>

XIE, Yihui, Joe CHENG, Xianying TAN a JJ ALLAIRE, 2022a. *DT: A Wrapper of the JavaScript Library „DataTables“* [online] [vid. 2021-11-27]. Dostupné z: <https://cran.r-project.org/package=DT>

XIE, Yihui, Wush WU, Daijiang LI, Xianying TAN, Salim BRÜGGEMANN a Christophe DERVIEUX, 2022b. *xfun: Supporting Functions for Packages Maintained by „Yihui Xie“* [online] [vid. 2021-11-15]. Dostupné z: <https://cran.r-project.org/package=xfun>

PŘÍLOHY

SEZNAM PŘÍLOH

Volné přílohy

Příloha 1 CD

Příloha 2 Poster

Popis struktury CD

Adresáře:

- aplikace
- data
- text_prace
- web