

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103003/B/2022/36122167902279940

Vizualizácia dát v systéme R

Bakalárska práca

2022

Patrik Jakubík

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Vizualizácia dát v systéme R
Bakalárska práca

Študijný program:	Hospodárska informatika
Študijný odbor:	Hospodárska informatika
Školiace pracovisko:	Katedra operačného výskumu a ekonometrie FHI
Vedúci záverečnej práce:	Ing. Adriana Lukáčiková, PhD.

Bratislava 2022

Patrik Jakubík

Čestné vyhlásenie

Čestne vyhlasujem, že som bakalársku prácu s názvom: Vizualizácia v systéme R, vypracoval samostatne pod vedením svojho školiteľa a že som uviedol/a všetku použitú literatúru.

Dátum:

.....

Podpis študenta

Pod'akovanie

Pod'akovanie patrí vedúcej bakalárskej práce Ing. Adriane Lukáčikovej PhD. za jej pomoc a ochotu pri tvorbe tejto práce.

Abstrakt

JAKUBÍK, Patrik: *Vizualizácia dát v systéme R.* – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie FHI. – Ing. Adriana Lukáčiková, PhD. – Bratislava: FHI, 2022, počet strán (46)

Cieľom tejto bakalárskej práce je predstaviť jednotlivé možnosti vizualizácie dát v systéme R. Tento softvér patrí k vedúcim nástrojom v oblasti štatistiky, ekonometrie a dátovej vedy, pretože disponuje veľkým množstvom funkcionalít.

Práca je rozdelená do štyroch kapitol. Obsahuje 16 obrázkov. Prvá kapitola, teda teoretická časť, je venovaná pojmu vizualizácia dát, charakteristike jazyka R, jeho histórii a tiež konkrétnym technológiám, ktoré sa využívajú na tvorbu grafov v jazyku R. V druhej kapitole si určíme konkrétne ciele. Tretia kapitola sa zaoberá metódami a postupmi, ktoré budeme v práci využívať. Priblížime si vývojové prostredie RStudio a postup vizualizácie v ňom. Praktická časť obsahuje rôzne spôsoby vizualizácií na mnou vybraných skutočných dátach. Táto časť obsahuje popis jednotlivých zhotovených grafov a tiež kód, ktorým boli tieto grafy vytvorené.

Kľúčové slová

Vizualizácia dát, graf, systém R, ggplot2

Abstract

JAKUBÍK, Patrik: *Data visualization in the R system*. – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Operations Research and Econometrics FHI. - Ing. Adriana Lukáčiková, PhD. – Bratislava: FHI, 2022, number of pages (46)

The aim of this bachelor thesis is to present the various possibilities of data visualization in the system R. This software is one of the leading tools in the field of statistics, econometrics and data science, because it has a large number of functionalities.

The thesis is divided into four chapters. It contains 16 figures. The first chapter, the theoretical part, is devoted to the concept of data visualization, the characteristics of the R language, its history and also the specific technologies that are used to create graphs in the R language. In the second chapter, we set specific goals. The third chapter deals with the methods and procedures that we will use in the thesis. We will get acquainted with the RStudio development environment and the visualization process in it. The practical part contains various ways of visualization on my selected actual data. This part contains a description of the individual graphs created, as well as the code with which these graphs were created.

Keywords

Data visualization, graph, R system, ggplot2

Zoznam obrázkov

Obrázok 1 Vývojové prostredie RStudio	18
Obrázok 2 Skladaný stĺpcový graf	22
Obrázok 3 Spojnicový graf	24
Obrázok 4 Spojnicové grafy	25
Obrázok 5 Plošný graf 1.....	26
Obrázok 6 Plošný graf 2.....	28
Obrázok 7 Kruhový graf	30
Obrázok 8 Bodový graf.....	31
Obrázok 9 Bublinový graf.....	33
Obrázok 10 Schéma krabicového grafu	33
Obrázok 11 Krabicový graf.....	35
Obrázok 12 Husľový graf	36
Obrázok 13 Stromová mapa.....	38
Obrázok 14 Graf hustoty	39
Obrázok 15 Ridgeline graf.....	40
Obrázok 16 Chorepletova mapa.....	42

Obsah

Úvod.....	10
1 Súčasný stav riešenej problematiky doma a v zahraničí.....	11
1.1 Vizualizácia dát	11
1.2 Jazyk R.....	12
<i>1.2.1 História jazyka R.....</i>	<i>12</i>
<i>1.2.2 Jazyk R</i>	<i>12</i>
1.3 Vizualizácia dát v systéme R.....	13
<i>1.3.1 Základný modul pre tvorbu grafov.....</i>	<i>13</i>
<i>1.3.2 The lattice.....</i>	<i>14</i>
<i>1.3.3 Plotly</i>	<i>14</i>
<i>1.3.4 Ggplot2.....</i>	<i>14</i>
2 Cieľ práce	17
3 Metodika práce a metódy skúmania	18
3.1 RStudio.....	18
4 Výsledky práce a diskusia	21
4.1 Stĺpcový graf.....	21
4.2 Spojnicový graf.....	23
4.3 Plošný graf	25
4.4 Kruhový graf	28
4.5 Bodový graf.....	30
4.6 Bublinový graf.....	31
4.7 Krabicový graf.....	33
4.8 Husľový graf.....	35
4.9 Stromový graf.....	37
4.10 Graf hustoty.....	38
4.11 Ridgeline graf	39

4.12 Choropletova mapa	41
Záver	43
Zoznam použitej literatúry	45

Úvod

Spôsob, akým analyzujeme údaje, sa v posledných rokoch dramaticky zmenil. S príchodom osobných počítačov a internetu enormne vzrástol objem údajov, ktoré máme k dispozícii. V dnešnej dobe je veda o dátach tzv. Data Science považovaná za jednu z najrýchlejšie rastúcich a najperspektívnejších oblastí. Nástup osobných počítačov (a najmä dostupnosť monitorov s vysokým rozlíšením) mal tiež vplyv na to, ako sú dáta chápané a prezentované. Obrázok skutočne povie viac ako tisíc slov a ľudia sú prispôsobení na získavanie užitočných informácií z vizuálnych prezentácií. Moderná analýza údajov sa čoraz viac spolieha na grafické prezentácie, aby odhalila význam a sprostredkovala výsledky. Dnešní dátoví analytici potrebujú pristupovať k údajom zo širokej škály zdrojov (systémy na správu databáz, textové súbory, štatistické balíky, tabuľky a pod.). Potrebujú zlučovať a analyzovať dáta pomocou najnovších metód a prezentovať zistenia zmysluplným a graficky príťažlivým spôsobom.

Existuje veľa možností a spôsobov ako vizualizovať dáta. Na trhu je dostupné veľké množstvo rôznych softvérov na analyzovanie a následnú vizualizáciu údajov. Táto bakalárska práca sa venuje konkrétne vizualizácii dát v systéme R.

Prvá časť práce je zameraná na pojem vizualizácia údajov. Jej začiatky a dôležitosť z historického aj súčasného hľadiska. Ďalej sa zaoberá už spomínanému programovaciemu jazyku R, a tiež špecifikácii a využitiu rôznych balíčkov jazyka R tzv. packages, ktoré slúžia na tvorbu grafov. Ďalšia kapitola je venovaná vývojovému prostrediu RStudio. Pozrieme sa na daný jazyk a softvér pomocou, ktorého budeme spracovávať a vizualizovať vybrané dáta. Posledná časť obsahuje rôzne typy grafických výstupov so stručným popisom konkrétneho typu grafu a tiež kód, ktorým boli tieto vizualizácie vytvorené. Všetky tieto vizualizácie boli vytvorené pomocou reálnych dát z rôznych oblastí.

1 Súčasný stav riešenej problematiky doma a v zahraničí

1.1 Vizualizácia dát

S príchodom moderných technológií sa stávala vizualizácia údajov stále jednoduchšou. Niekoľko kliknutí myšou dokáže viac-menej okamžite zmeniť obrovskú tabuľku nespracovaných čísel na vizuálne prítiažlivý a ľahko interpretovateľný diagram. Vytváranie grafov poskytuje vysokorýchlostnú skratku na vytvorenie porozumenia. Mnohé zo súčasných vizualizačných techník boli vynájdené počas priemyselnej revolúcie, pričom táto oblasť urobila veľké pokroky hlavne v polovici 19. storočia.

Pravdepodobne prvé vizualizácie údajov boli v oblasti kartografie. Mapy, ktoré sa pôvodne používali na účely navigácie, vlastníctva pôdy a všeobecnej ľudskej zvedavosti, existujú v tej či onej forme už najmenej desaťtisíc rokov.

*Vizualizácia dát je odbor, ktorý sa zaoberá digitálnym spracovaním a prezentáciou abstraktných dát.*¹ Kvantitatívne údaje sa získavajú meraním, analýzou alebo štatistickými metódami. Hlavným cieľom je spracovať komplexné dáta a zobrazit' ich vo vizuálnej forme, ktorá je pre človeka zrozumiteľnejšia ako numerické vyjadrenie. Vďaka vizualizácii majú ľudia možnosť získať lepší prehľad o téme, odhaliť zákonitosti, atribúty a vzťahy medzi dátami, čo pomáha pochopiť výsledky skryté v abstraktných údajoch. Mozog človeka spracováva vizuálne informácie 60 000-krát rýchlejšie ako text a až 90 percent informácií prenášaných do mozgu odpovedá vizuálnym údajom.

*Vizualizácia údajov je čiastočne umenie a čiastočne veda.*² Vizualizácia dát musí v prvom rade presne sprostredkovať dané údaje. Nesmie zavádzať ani skresľovať. Nepresné údaje vo vizualizácii môžu negatívne ovplyvniť záver a viesť k nekorektnej interpretácii výsledkov. Ak je napríklad jedno číslo dvakrát väčšie ako druhé, ale vo vizualizácii vyzerajú približne rovnako, potom je vizualizácia nesprávna. Jeden zo znakov dobrej vizualizácie je estetickosť. Ak vizualizácia obsahuje rušivé farby alebo rôzne iné nevyvážené vizuálne prvky, ktoré rozptyľujú pozornosť, potom bude pre diváka ťažšie túto vizualizáciu preskúmať a pochopiť. Dobré vizuálne prezentácie majú tendenciu zlepšiť posolstvo vizualizácie.

¹ DETKO, Marek. Interaktivní vizualizace dat a metadat [elektronický zdroj]. 16.5.2017 Dostupné na: <https://www.cognito.cz/design/interaktivni-vizualizace-dat-a-metadat>

² WILKE, Claus O. Fundamentals of Data Visualization - A Primer on Making Informative and Compelling Figures, 1. vyd. USA: O'Reilly Media, 2019. s. 1 ISBN 978-14-9203-108-6

1.2 Jazyk R

1.2.1 História jazyka R

Jazyk R je dialekt jazyka S. Jazyk S vyvinul John Chambers, Rick Becker a ďalší v Bell Labs. Jazyk S bol spustený v roku 1976 ako interné prostredie pre štatistickú analýzu. Verzia 4 jazyka S bola vydaná v roku 1998 a používa sa dodnes. *Kľúčovou myšlienkou vytvorenia jazyka S a neskôr jazyka R bolo poskytnúť jazyk, ktorý je vhodný na interaktívnu analýzu dát, ako aj na písanie dlhších programov.*³

Jazyk R sa začal používať pomerne neskôr po vyvinutí jazyka S. Jedným z kľúčových obmedzení jazyka S bolo, že bol dostupný iba v komerčnom balíku S-PLUS. Jazyk R vytvorili Ross Ihaka a Robert Gentleman na katedre štatistiky na univerzite v Aucklande v roku 1991. V roku 1993 bol jazyk R predstavený verejnosti. V roku 1995 Martin Mächler významne prispel tým, že presvedčil Rossa a Roberta, aby použili GNU (General Public License), a toto spôsobilo, že jazyk R bol deklarovaný ako open source projekt. To bolo kritické, pretože to umožnilo, aby zdrojový kód celého systému R bol prístupný každému, kto si s ním chcel pohrať. Nakoniec bola v roku 2000 pre verejnosť uvoľnená verzia R 1.0.0.

1.2.2 Jazyk R

R je jazyk a prostredie pre štatistické výpočty a grafiku. Ako už bolo spomenuté, ide o projekt GNU, ktorý je podobný jazyku a prostrediu S. Existuje niekoľko dôležitých rozdielov, ale veľa kódu napísaného pre S beží bez zmien pod R. Existuje mnoho dobrých dôvodov, prečo na analýzu dát používať práve systém R. To ukazuje aj fakt, že R je najpoužívanejším nástrojom v data science.

R poskytuje širokú škálu štatistických (lineárne a nelineárne modelovanie, klasické štatistické testy, analýza časových radov, klasifikácia, zhukovanie, ...) a grafických techník a je vysoko rozšíriteľný. Jednou zo silných stránok R je jednoduchosť, s akou možno vytvárať dobre navrhnuté grafy, vrátane matematických symbolov a vzorcov, ak je to potrebné.

Jazyk R beží na širokej škále platforiem UNIX a podobných systémoch (vrátane FreeBSD a Linux), Windows a MacOS.

³ KUMAR, Mohit. R Overview and history [elektronický zdroj]. 12.8.2020 Dostupné na: <https://medium.com/@ArtisOne/r-overview-and-history-75ecb036d0df>

Jedna pekná vlastnosť, ktorú R zdieľa s mnohými populárnymi open source projektmi, sú časté vydania. Pravidelný cyklus vydávania naznačuje aktívny vývoj softvéru a zaisťuje, že chyby sú včas odstraňované. Ďalšou kľúčovou výhodou, ktorú má R oproti mnohým iným štatistickým balíkom, sú jeho sofistikované grafické možnosti. R má schopnosť vytvárať grafiku „publikačnej kvality“.⁴ Dôsledkom existencie veľkej a rastúcej komunity používateľov je, že R obsahuje veľké množstvo metód pre riešenie takmer všetkých problémov v rôznych oblastiach štatistiky, ekonometrie a data science. Väčšina týchto metód je distribuovaná formou balíkov, ktoré sú uložené na jednom z centrálnych dátových skladov. Základný grafický systém R umožňuje kontrolu nad v podstate každým aspektom grafu.

R si zachoval pôvodnú filozofiu S, ktorá spočíva v tom, že poskytuje jazyk, ktorý je užitočný pre interaktívnu prácu, ale zároveň obsahuje výkonný programovací jazyk na vývoj nových nástrojov.⁵ To umožňuje používateľovi, ktorý používa existujúce nástroje a aplikuje ich na dáta, stať sa pomaly, ale isto vývojárom, ktorý vytvára nové nástroje.

1.3 Vizualizácia dát v systéme R

Systém R poskytuje množstvo rôznych nástrojov pre vizualizáciu dát. Jednou z najvýkonnejších funkcií R je jeho schopnosť vytvárať širokú škálu grafov na rýchlu a jednoduchú vizualizáciu údajov. Grafy je možné replikovať, upravovať a dokonca publikovať pomocou niekoľkých príkazov. Jedným z najkomplexnejších a tiež najpoužívanejších nástrojov na vizualizáciu dát v systéme R je balík ggplot2. Pomocou tohto balíčka boli vytvorené takmer všetky grafy v tejto práci.

1.3.1 Základný modul pre tvorbu grafov

Základný modul tzv. Base modul je štandardnou súčasťou systému R a nevyžaduje sa inštalácia žiadnych dodatočných balíkov. Tento základný modul obsahuje funkciu plot(), ktorá dokáže vytvárať širokú škálu grafov. Jeho účelom je rýchle, ale nie veľmi komplexné zobrazenie. Pristupuje zložitým spôsobom pri nastavovaní jednoduchých požiadaviek a nie je možné efektívne meniť parametre grafov. Najväčšou výhodou je rýchlosť zobrazenia, takže je veľmi efektívny pri cyklickom zobrazovaní dát alebo vizualizácii dátových súborov.

⁴ FIELD, Andy a kol. Discovering Statistics Using R, 1. vyd. USA: SAGE Publications Ltd, 2012. s. 294 ISBN 978-14-4620-046-9

⁵ PENG, Roger. R Programming for Data Science, 5. vyd. USA: lulu.com, 2016. s. 194 ISBN 978-6505-682-6

1.3.2 The lattice

The lattice je balík, vyvinutý Deepayanom Sarkarom, používa grid grafiku na implementáciu mriežkového grafického systému Cleveland12 a predstavuje značné zlepšenie oproti základnej grafike. Poskytuje možnosť jednoducho vytvárať podmienené grafy a niektoré detaily vykresľovania (napr. legendy) sa spravia automaticky. Lattice grafike však chýba formálny model, čo môže sťažiť jej rozšírenie.

1.3.3 Plotly

Plotly je populárny javascriptový vizualizačný nástroj s rozhraním R. Akýkoľvek graf vytvorený pomocou balíka plotly využíva knižnicu JavaScript plotly.js. Je to skvelý nástroj, ak chcete vytvárať interaktívnu grafiku pre dokumenty HTML. Syntaxou je podobný ggplot2. *Má dokonca funkciu ggplotly(), ktorá prepája balíček ggplot2 a plotly nenáročným spôsobom a dokáže previesť mnohé grafiky ggplot2 na ich interaktívne ekvivalenty.*⁶

1.3.4 Ggplot2

Balík ggplot2 napísal Hadley Wickham v roku 2009. Zámerom balíka ggplot2 je poskytnúť komplexný, na gramatike založený systém na generovanie grafov jednotným a koherentným spôsobom, ktorý používateľom umožňuje vytvárať nové a inovatívne vizualizácie údajov. *Na rozdiel od väčšiny ostatných grafických balíkov má ggplot2 základnú gramatiku založenú na Grammar of Graphics, ktorá umožňuje zostavovať grafy kombináciou nezávislých komponentov.*⁷ Vďaka tomu je ggplot2 výkonný. Namiesto obmedzovania sa na sady preddefinovanej grafiky, môžete vytvoriť novú grafiku, ktorá je prispôbená špecifickému problému.

Ggplot2 je navrhnutý tak, aby fungoval iteratívne. Graf sa skladá zo série vrstiev. Vrstvy sa dajú predstaviť ako plastové priehľadné fólie, na ktorých je niečo vytlačené. To „niečo“ môže byť text, dátové body, čiary, pruhy alebo čokoľvek iné. Začína sa s vrstvou, ktorá zobrazuje nespracované údaje. Potom sa pridajú vrstvy anotácií a štatistických súhrnov.

⁶ WICKHAM, Hadley a kol. Ggplot2: Elegant Graphics for Data Analysis, 2. vyd. Nemecko: Springer, 2016. s. 6 ISBN 978-03-8798-140-6

⁷ WICKHAM, Hadley a kol. Ggplot2: Elegant Graphics for Data Analysis, 2. vyd. Nemecko: Springer, 2016. s. 6 ISBN 978-03-8798-140-6

Všetky grafy sa skladajú z dát (informácií, ktoré chceme vizualizovať), a mapovania (popisu toho, ako sú premenné údajov mapované na estetické atribúty). Existuje päť komponentov mapovania:

- Layers - vrstva je súbor geometrických prvkov a štatistických transformácií. Geometrické prvky (geoms), predstavujú to, čo skutočne vidno na grafe: body, čiary, mnohoúhelníky atď. Štatistické transformácie (stats), sumarizujú údaje: napríklad zoskupovanie a počítanie pozorovaní na vytvorenie histogramu.
- Scales - mapujú hodnoty v dátovom priestore na hodnoty v estetickom priestore. To zahŕňa použitie farby, tvaru alebo veľkosti. Tiež vykresľujú legendu a osi. Pre každý objekt je možné definovať v akej škále má byť reprezentovaný
- Súradnicový systém - popisuje, ako sú súradnice údajov mapované do roviny grafu. Poskytuje tiež osi a mriežky, ktoré pomáhajú pri čítaní grafu. Bežne používame karteziánsky súradnicový systém, ale je k dispozícii množstvo ďalších, vrátane polárnych súradníc a projekcií máp.
- Facets - definujú, ako sú dáta rozdelené a zobrazené.
- Theme - téma ovláda jemnejšie body zobrazenia, ako je veľkosť písma a farba pozadia.

Postup vizualizácie:

Každý graf ggplot2 má tri kľúčové komponenty:

- Data – každá tvorba grafu začína dátami, ktoré sú do funkcie ggplot() vkladané vo formáte databázovej tabuľky tzv. data.frame
- Súbor estetických mapovaní (aesthetic mapping) - definuje estetické atribúty dát (veľkosť, pozíciu, farbu, tvary)
- Layer - aspoň jedna vrstva, ktorá popisuje spôsob vykreslenia každého pozorovania. Vrstvy sa zvyčajne vytvárajú pomocou funkcie geom().

Prvý krok pri vytváraní vizualizácie je volanie funkcie `ggplot()`. Táto funkcia má dva parametre. Prvý parameter `data` obsahuje dátový súbor v podobe tabuľky, ktorý sa vizualizuje. Druhý parameter `mapping` prijíma základné požiadavky, podľa ktorých sa má vizualizácia riadiť. Tieto požiadavky sa nachádzajú vo funkcii `aes()`. Prvými dvoma argumentmi tejto funkcie sú premenné osí `x` a `y`. Ďalej sa do vytváraného grafu pridávajú ďalšie komponenty pomocou znamienka `+`. Ako typické vykresľované geometrické objekty môžeme spomenúť napríklad body, čiary alebo stĺpce. Následne v rámci funkcie

`geom_*()` voláme funkciu `aes()`, ktorou docielime konkretizovanie jedinečných vlastností charakteristických len pre danú vrstvu vo viacvrstvovom grafe. Myšlienkou `ggplot2` je teda grafické prepojenie vstupných dát a špecifických estetických objektov, ktoré sa vyznačujú geometrickou štruktúrou a je možné ich modifikovať v mierke a štýle.

2 Cieľ práce

V súčasnej dobe je stále viac nástrojov a technológií, vďaka ktorým môžeme lepšie spracovávať a využívať potenciál získaných dát. Hlavným cieľom bakalárskej práce je popísať a priblížiť grafické spracovanie dát v systéme R. Vizualizované dáta majú pre človeka väčšiu pridanú hodnotu ako dáta prezentované len číselným vyhodnotením alebo písaným textom.

Na to, aby sme splnili hlavný cieľ, musíme splniť aj niekoľko čiastkových cieľov. Prvým čiastkovým cieľom je stručné popísanie základov jazyka R a vysvetlenie základnej syntaxe jazyka.

Ďalším čiastkovým cieľom je oboznámenie s vývojovým prostredím RStudio a vysvetlenie jeho základných funkcionalít, ktoré budú potrebné na vizualizáciu dát v tejto bakalárskej práci.

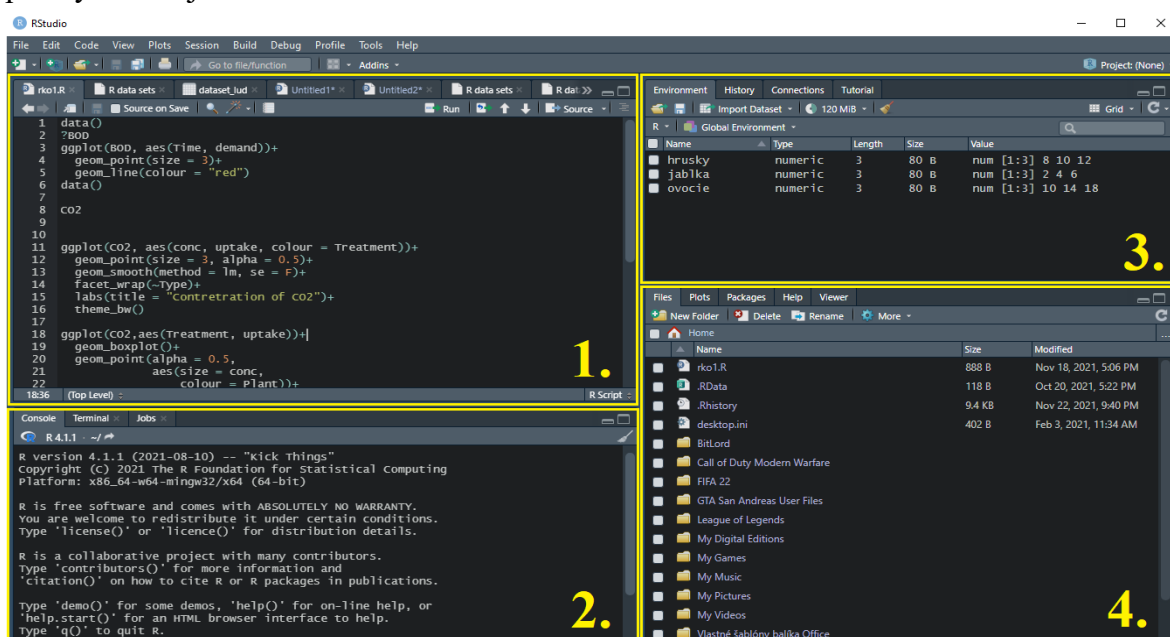
Tretím čiastkovým cieľom je špecifikovať a priblížiť využitie jednotlivých balíkov tzv. packages, ktoré slúžia na rôzne typy vizualizácií v systéme R. Až po splnení všetkých týchto čiastkových cieľov môžeme splniť hlavný cieľ, ktorým je v podstate vytvorenie návodu na tvorbu rozličných typov grafov pomocou systému R. Dobre vytvorený graf musí spĺňať nejaké podmienky, musí byť prehľadný, ľahko čitateľný a presne sprostredkovať vybrané dáta.

3 Metodika práce a metódy skúmania

Ako už bolo spomenuté, na tvorbu tejto bakalárskej práce sme zvolili vývojové prostredie RStudio. Tento softvér je v súčasnej dobe považovaný ako jeden z najlepších pre prácu s jazykom R. Predstavuje súbor nástrojov a techník, ktoré zjednodušujú a rozširujú základné funkcionality jazyka R. Rozhodli sme sa pre použitie daného softvéru, pretože disponuje množstvom výhod oproti základnému prostrediu R.

3.1 RStudio

RStudio je integrované vývojové prostredie (IDE), ktoré pridáva moderné funkcie, ako je zvýrazňovanie syntaxe a refaktoring kódu do R. Sila RStudio je, že spája všetky funkcie, ktoré potrebujete, na jednom mieste. Program RStudio je možné spustiť na pracovnej ploche alebo cez webový prehliadač. Desktopová verzia je dostupná pre platformy Windows, Mac OS X a Linux a správa sa podobne na všetkých platformách s malými rozdielmi v klávesových skratkách. Rovnako ako R, aj RStudio je open source projekt. Jeho stanoveným cieľom, ktorý už spĺňa, je vyvinúť výkonný nástroj, ktorý podporuje postupy a techniky potrebné na vytváranie dôveryhodných a vysokokvalitných analýz. Pri tvorbe tejto práce som používal výhradne program RStudio na vytváranie všetkých grafov, ktoré sa nachádzajú v tejto bakalárskej práci. RStudio sa skladá zo všeobecného menu a štyroch panelov ako je možné vidieť na obrázku 1. Každý z týchto panelov obsahuje niekoľko záložiek. Používateľ si môže nakonfigurovať rozmiestenie a veľkosť daných panelov. Pod priloženým obrázkom sú stručne popísané všetky štyri panely a záložky, ktoré jednotlivé panely obsahujú.



Obrázok 1 Vývojové prostredie RStudio (zdroj: vlastné spracovanie)

V prvom paneli sa nachádza editor, to znamená priestor, v ktorom sa vytvárajú, upravujú a načítavajú rôzne scripty. Kód, ktorý do neho napíšete, môžete uložiť a opakovane spúšťať. V editore môžete súčasne editovať ľubovoľné množstvo súborov rôznych typov (R script, R markdown, textové dokumenty a mnohé ďalšie). Po ukončení rôznych úprav má užívateľ možnosť spustiť daný script pomocou tlačidla run, ktoré sa nachádza v pravom hornom rohu tohto panelu. Keď sa v scripte nachádzajú nejaké chyby, tak chybové hlásky sa potom zobrazujú v druhom paneli (konkrétne v konzole).

Druhý panel predstavuje samotnú konzolu vývojového prostredia. Tá je určená pre interaktívnu prácu s R. Kód, ktorý do nej zapíšete, RStudio okamžite vyhodnotí a výsledky vypíše do konzoly alebo zobrazí ako graf.

Tretí panel obsahuje záložky enviroment, history a connections. V záložke enviroment sa nachádzajú všetky objekty (dáta, funkcie a pod.), ktoré boli použité vo zvolenom prostredí v R. Vďaka takémuto zoznamu, v ktorom sa nachádzajú už definované premenné si vie používateľ jednoducho skontrolovať, či sa premenná v systéme už nachádza, a tým predísť duplicite a prepísaniu hodnôt premenných. Táto funkcia je veľkou výhodou RStudio. Enviroment tiež umožňuje importovať niektoré typy dát a mazať predtým vytvorené objekty. Záložka history zobrazuje kód, ktorý bol v minulosti spustený v konzole. Umožňuje ho tiež uložiť a presunúť späť do konzoly alebo do editora. Môžete tiež vyhľadať konkrétny riadok kódu pomocou textovej oblasti s ikonou vyhľadávania. To môže byť užitočné, keď viete, že ste vykonali zmenu na objekte, ale máte problém presne lokalizovať, kde v potenciálne dlhom zozname príkazov boli zmeny vykonané. A záložka connections poskytuje spôsob pripojenia k zdrojom údajov. Tieto typy riešení na ukladanie údajov sa používajú, keď sú datasety príliš veľké na to, aby sa dali jednoducho uložiť lokálne do vášho počítača. Ak máte informácie o pripojení k databáze, ktorú potrebujete pre svoj projekt, môžete ju jednoducho pripojiť pomocou tejto záložky.

V poslednom, štvrtom paneli sa nachádza viac záložiek. Konkrétne Files, Plot, Packages, Help a Viewer. V záložke Files sa zobrazujú súbory a adresáre, ktoré sú v aktuálnom projekte alebo adresári a umožňujú s nimi robiť základné operácie (mazať ich, premenovávať a pod.). Záložka plot zobrazuje grafy, ktoré boli v R vykreslené. Táto záložka sa otvorí až vo chvíli, keď je nejaký graf vytvorený. Ďalej v záložke packages je možné vidieť zoznam inštalovaných balíkov. Zároveň umožňujú aj balíky načítať, inštalovať, aktualizovať a odstraňovať. Záložka help je nastavená ako webový prehliadač. Obsahuje dokumentáciu k funkciám, dátam a balíkom. Posledná záložka v štvrtom paneli je viewer.

Toto je oblasť, kde sa objaví výstup z niektorých pokročilejších funkcií reportovania RStudia, ako sú dokumenty R markdown. Takéto dokumenty sa používajú na prezentáciu kódu a výsledkov spoločne reprodukovateľným spôsobom, ktorý sa používa na zdieľanie výsledkov. V tejto práci je pre nás dôležitá hlavne záložka Plot, vďaka ktorej budeme vizualizovať rôzne typy grafov.

4 Výsledky práce a diskusia

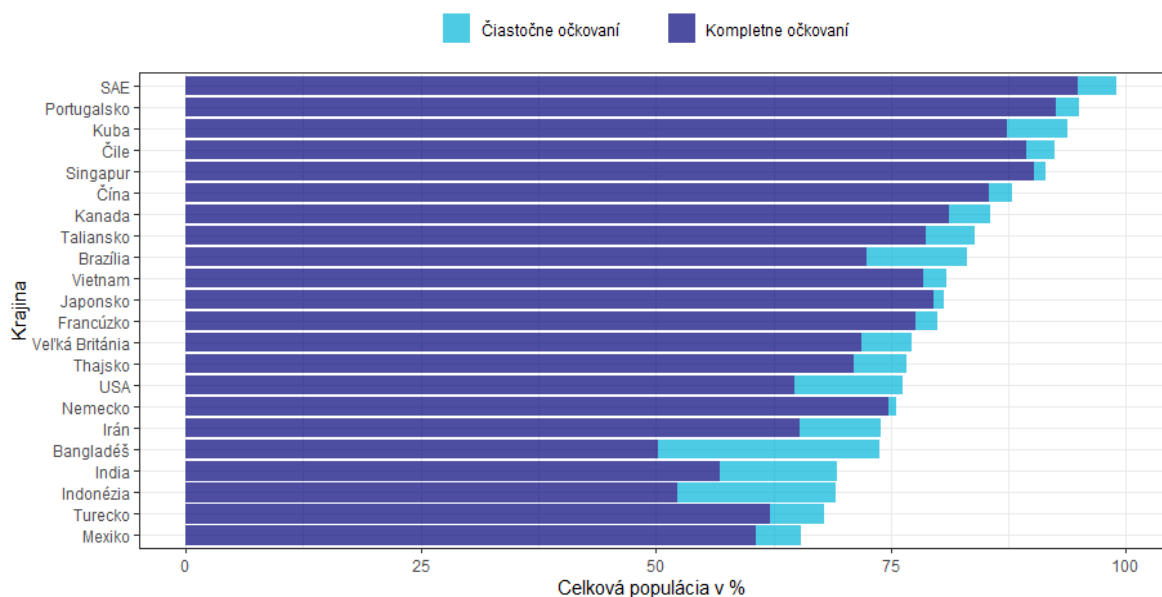
4.1 Stĺpcový graf

Stĺpcové grafy sú jedny z najčastejšie používaných grafov. Užitočné sú na zobrazenie zmien údajov za časové obdobie alebo na znázornenie porovnania medzi položkami. V stĺpcových grafoch sú kategórie usporiadané zvyčajne pozdĺž vodorovnej osi a hodnoty pozdĺž zvislej osi, ale samozrejme to môže byť aj naopak. Jednotlivé stĺpce by sa v stĺpcovom grafe nemali dotýkať, pretože je to hlavný spôsob ako odlíšiť stĺpcový graf od histogramu. Rozlišujeme viac druhov stĺpcových grafov ako napríklad skupinový alebo skladaný. Skupinový stĺpcový graf umožňuje zobrazenie ďalšej dimenzie dát, ktorou sú podskupiny jednotlivých hlavných kategórií. Skladaný stĺpcový graf umožňuje to isté, avšak hodí sa pre prípady, kedy podskupiny dohromady tvoria celú hlavnú kategóriu - stĺpce tu skladáme nad seba.

```
vakcinacia <- read_excel("vakcinacia.xlsx")
vakcinacia %>%
  ggplot(aes(x = reorder(Krajina,+Pocet), y = Pocet, fill =
Typ)) +
  geom_bar(stat = "identity",
           position = "stack",
           alpha = 0.7) +
  coord_flip() +
  scale_fill_manual(values = c("#00b5d9", "#000278")) +
  theme_bw() +
  labs(x = "Krajina",
       y = "Celková populácia v %",
       title = "") +
  theme(legend.position="top") +
  theme(legend.title = element_blank())
```

Daný kód zobrazuje postup tvorby skladaného stĺpcového grafu na obr. 2. Kód bol vytvorený pomocou balíčka ggplot2. Vstupné parametre daného grafu boli spracované a prevedené na požadovaný formát v programe Microsoft Excel a následne importované do RStudio príkazom `<- read_excel("vakcinacia.xlsx")`. Príkazom `vakcinacia %>%` sa vloží konkrétny dátový súbor do funkcie `ggplot()`. Parameter

mapping, konkrétne (`aes(x = reorder(Krajina, +Pocet), y = Pocet, fill = Typ)`) graficky prepája dátový súbor s osami x a y. Na x-ovú os bude vykreslená premenná *Krajina* a na y-ovú os bude vykreslená premenná *Pocet*. Funkcia `aes()` obsahuje tiež parameter `fill = Typ`, ktorý zafarbí stĺpce podľa typu zaočkovanosti. Ďalej je vrstva `geom_bar()`, ktorá je používaná pri tvorbe stĺpcového grafu rôzneho typu. Predvolený argument `stat` je `stat_bin()`, ktorý sa používa na rozdelenie údajov do tried. Keď chceme vytvoriť stĺpcový graf, musí sa použiť argument `stat = "identity"`, ktorý vyžaduje špecifikáciu premennej uloženej do argumentu y. Následne máme argument `position = "stack"`, ktorý nám definuje, že ide o skladaný graf. Príkaz `scale_fill_manual()` umožňuje voľbu vlastných farieb pre jednotlivé stĺpce, ktoré budú farebne rozlíšené podľa typu zaočkovanosti. Pomocou funkcie `coord_flip()` vymeníme osy x a y. Vrstva `theme_bw()` sa vzťahuje na tému grafického výstupu a špecifikuje mriežkovanie, farbu pozadia a pod. Pomocou vrstvy `labs()` určujeme popisy osí x a y a názov grafu.



Obrázok 2 Skladaný stĺpcový graf celkovej zaočkovanosti populácie v jednotlivých krajinách (zdroj: ourworldindata.org , vlastné spracovanie)

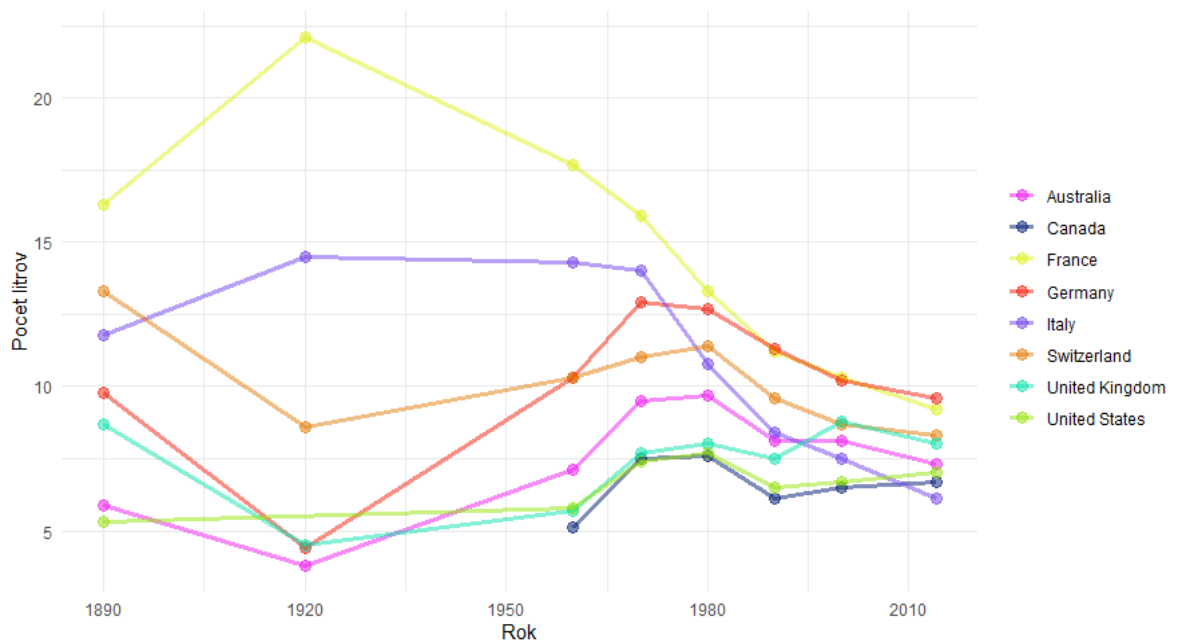
Skladaný stĺpcový graf na obrázku 2 zobrazuje percentuálnu zaočkovanosť proti ochoreniu COVID-19 k 1.3.2022 podľa jednotlivých krajín. Na grafe je farebne odlišené koľko percent ľudí je kompletne očkovaných a koľko je čiastočne očkovaných. Ako je vidno na obrázku najväčšiu percentuálnu zaočkovanosť majú Spojené arabské emiráty a to vyše 99 percent.

4.2 Spojnicový graf

Spojnicový graf slúži najčastejšie na zobrazenie súvisiacich dát závislých od času, a preto je ideálny pre grafickú vizualizáciu hodnôt v priebehu času. Časová premenná sa zvyčajne aplikuje na vodorovnú os a konkrétne hodnoty na zvislú os. Graf môže zobrazovať vývoj jednej alebo viacerých premenných, ktoré sa dajú rozlíšiť rôznymi druhmi čiar a farbami.

```
alcohol %>%  
  ggplot(aes(Year, Alcohol,  
             colour = Entity)) +  
  geom_point(size = 3, alpha = 0.5) +  
  geom_line(size = 1.3, alpha = 0.5) +  
  scale_color_manual(values = c("#EE17F1", "#001A73",  
                                "#DDF117", "#F41D0B", "#7041EC", "#E37C07", "#07E3AB",  
                                "#93E307")) +  
  theme_minimal() +  
  labs(x = "Rok",  
       y = "Pocet litrov") +  
  theme(legend.title = element_blank())
```

Kódom zobrazeným vyššie bol zhotovený graf na obrázku 3. Pomocou príkazu `alcohol %>%` sme vložili dátový súbor s názvom `alcohol` do funkcie `ggplot()`. Následne máme funkciu `aes()`, ktorej argumentmi určíme aké hodnoty budú vykresľované na osiach `x` a `y`. V tomto prípade bude pre os `x` prislúchajúca premenná `Year` a pre os `y` premenná `Alcohol`. Parametrom `colour = Entity` dosiahneme, že farby jednotlivých bodov a čiar budú rozlíšené podľa konkrétnych krajín. Ďalej sú vrstvy `geom_point()` a `geom_line()`, vďaka ktorým vykreslíme jednotlivé body a čiary v grafe. Argumentmi `size` a `alpha` určíme veľkosť a priehľadnosť daných bodov a čiar. Pomocou príkazu `theme_minimal()` sme zvolili vhodné pozadie a mriežkovanie vizualizácie. Nakoniec sme pomenovali jednotlivé osy `x` a `y` funkciou `labs()` a odstránili názov legendy, pretože by bol zbytočný.



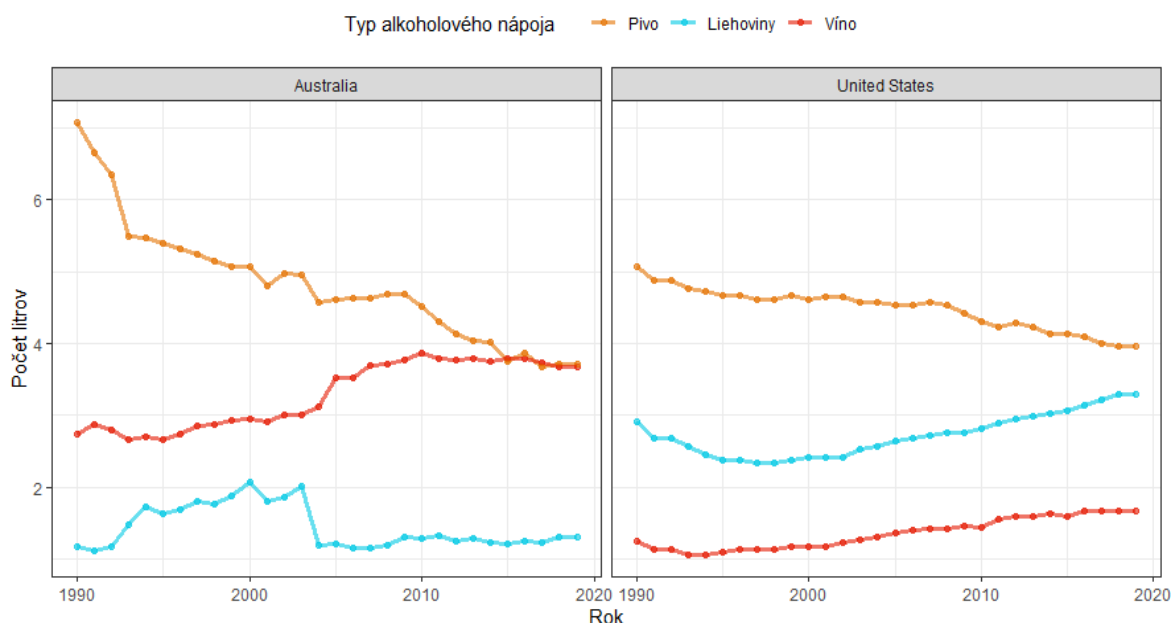
Obrázok 3 Spojnicový graf priemernej spotreby alkoholu na obyvateľa (zdroj: ourworldindata.org , vlastné spracovanie)

Na obrázku 3 sa nachádza spojnicový graf, ktorý zobrazuje priemernú spotrebu alkoholu na obyvateľa meranú v litroch čistého alkoholu na osobu a rok. Daná vizualizácia obsahuje údaje o spotrebe alkoholu v ôsmich krajinách od roku 1890 po rok 2019.

```
spirit %>%
  ggplot(aes(Year, Alcohol, color = Type)) +
  geom_point(alpha = 0.6)+
  geom_line(size = 1.3, alpha = 0.6)+
  scale_color_manual(labels = c("Pivo", "Liehoviny",
    "Víno"),
                    values = c("#E57301", "#01CAE5",
    "#E51901"))+
  facet_wrap(~Entity)+
  labs(x = "Rok",
    y = "Počet litrov",
    color = "Typ alkoholového nápoja")+
  theme_bw()+
  theme(legend.position="top")
```

Týmto kódom vytvoríme spojnicový graf podobný ako na obrázku 3 s tým rozdielom, že tento bude obsahovať spotrebu alkoholu rozdelenú podľa typu alkoholického

nápoja vo vybraných krajinách. Rozdiel oproti predošlému kódu je vo funkcii `aes()` taký, že body a čiary sú farebne odlíšené podľa typu alkoholického nápoja. Tiež jednotlivé body a čiary majú iné hodnoty parametrov `size` a `alpha` vo funkciách `geom_point()` a `geom_line()`. Funkciou `scale_color_manual()` sme zmenili farby bodov a čiar podľa typu alkoholického nápoja a tiež určili popisy legendy. Následne sa v kóde nachádza príkaz `facet_wrap(~Entity)`, ktorý priradí osobitný graf každej krajine zvlášť. Nakoniec sme pomocou príkazu `theme(legend.position="top")` určili konkrétnu polohu legendy grafu.



Obrázok 4 Spojnicové grafy priemernej spotreby alkoholu podľa typu (zdroj: ourworldindata.org , vlastné spracovanie)

Vizualizácia na obrázku 4 zobrazuje porovnanie spotreby rôznych druhov alkoholických nápojov na obyvateľa a to konkrétne piva, vína a liehovín v Austrálii a Spojených štátoch amerických počas rokov 1990 až 2019. Z grafu môžeme vidieť, že v oboch krajinách sa vypije najviac litrov piva. A tiež, že spotreba piva má od roku 1990 klesajúcu tendenciu, ale spotreba vína a liehovín naopak rastie aj v Austrálii aj v USA.

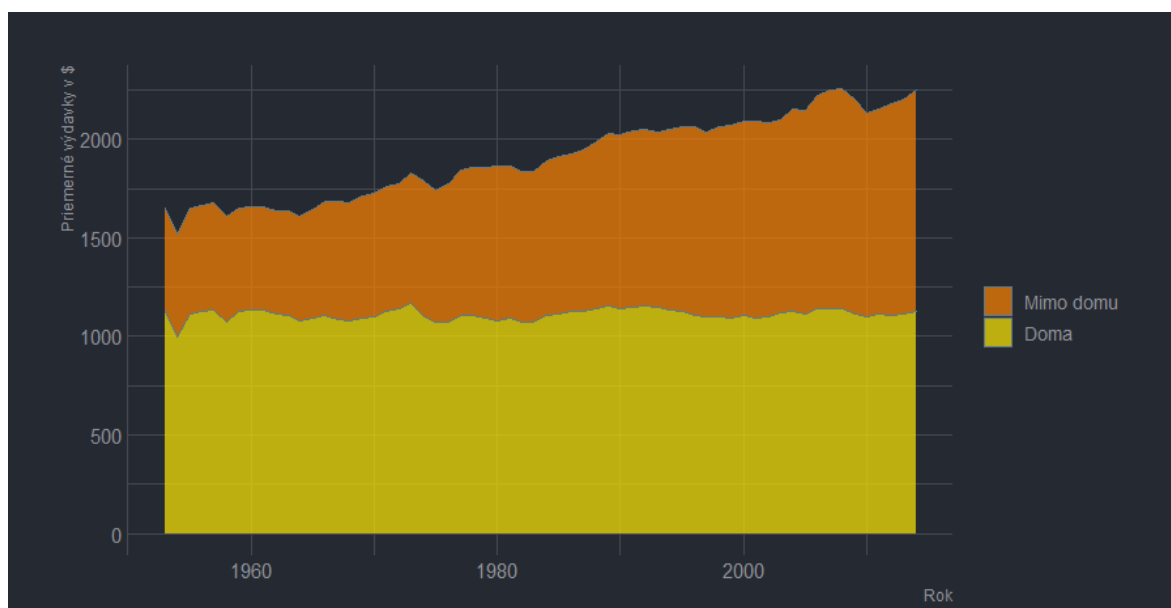
4.3 Plošný graf

Plošný graf je spojený graf, ktorý má vyplnenú plochu pod krivkou. Pre viac dátových radov sa často využíva skladaný plošný graf, ktorý zobrazuje dátové rady vykreslené s vyplnenými plochami naskladanými jedna na druhú. Takýto graf môže ukázať, ako sa v priebehu času menia jednotlivé vzťahy. Skladané stĺpcové grafy vo vynikajúcom formáte

umožňujú používateľom vidieť zmeny v sérii údajov a ich zastúpenie v celkovej oblasti grafu. Ponúka jednoduchú prezentáciu, ktorá je na prvý pohľad ľahko interpretovateľná.

```
food %>%
  ggplot(aes(Year, Number1))+
  geom_area(aes(fill = Type),
            alpha = 0.7)+
  scale_fill_manual(labels = c("Mimo domu", "Doma"),
                    values = c("#FF8300",
"#FFE800"))+
  theme_ft_rc()+
  labs(x = "Rok",
       y = "Priemerné výdavky v €")+
  theme(legend.title = element_blank(),
       text = element_text(size = 13))
```

Tento kód zobrazuje tvorbu skladeného plošného grafu použitím balíčka ggplot2, ktorý sa konštruje pomocou geometrickej vrstvy volanej funkciou `geom_area()`. V rámci tejto funkcie sme parametrom `fill = Type` docielili, že jednotlivé plochy tohto konkrétneho grafu budú farebne oddelené podľa premennej `Type`. Príkaz `text = element_text(size = 13)`, ktorý sa nachádza na konci kódu slúži na zmenu veľkosti písma v grafe.

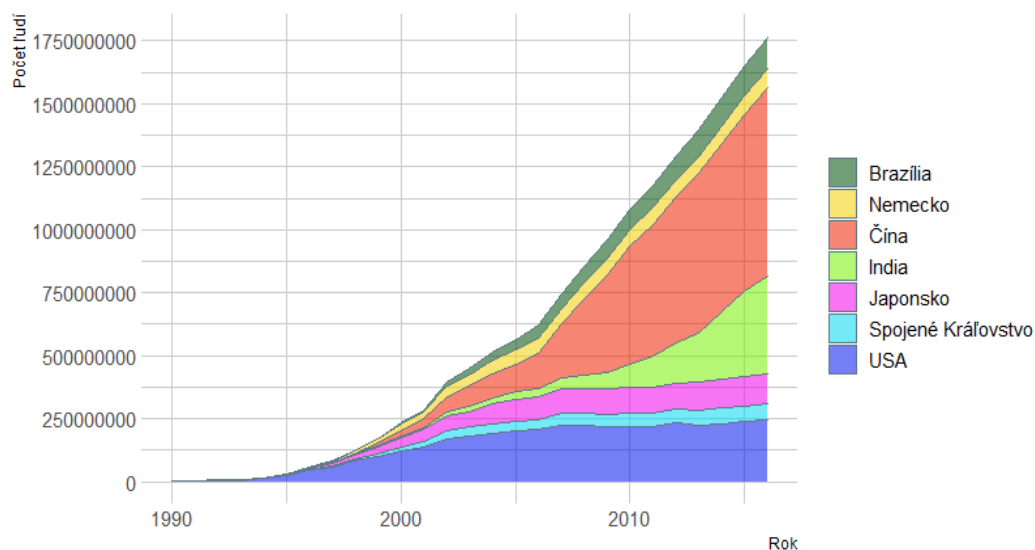


Obrázok 5 Priemerné ročné výdavky na jedlo na osobu v USA (zdroj: ourworldindata.org , vlastné spracovanie)

Na grafe na obrázku 5 môžeme vidieť priemerné ročné výdavky na osobu v Spojených štátoch amerických farebne rozdelené na výdavky vynaložené na jedlo doma a mimo domova (ako sú reštaurácie, kaviarne, vysoké školy, práca atď.) v rokoch 1953 až 2014.

```
internet %>%
  ggplot(aes(Year, Number))+
  geom_area(aes(fill = Entity),
            alpha = 0.55)+
  scale_fill_manual(labels = c("Brazília", "Nemecko",
                                "Čína",      "India", "Japonsko",  "Spojené Kráľovstvo",
                                "USA"),
                    values = c("#00500A", "#F3CF01",
                                "#F02101", "#78F001", "#F001ED", "#01D7F0",
                                "#0116F0"))+
  scale_y_continuous(breaks = seq(0,1750000000,250000000))+
  theme_ipsum()+
  labs(x = "Rok",
        y = "Počet ľudí")+
  theme(legend.title = element_blank(),
        text = element_text(size = 14))
```

Týmto kódom vytvoríme podobný plošný skladaný graf ako na obrázku 5, s tým rozdielom, že zobrazujeme viac dátových radov. Ako je možné vidieť daný kód je takmer totožný ako ten predošlý. Pridali sme sem funkciu `scale_fill_manual()`, vďaka ktorej sme zvolili vlastné farby grafu a tiež jednotlivé texty nachádzajúce sa v legende. Následne sme pomocou príkazu `scale_y_continuous()` definovali postupnosť čísiel zobrazených na osi y, s tým že čísla 0 až 1750000000 predstavujú rozsah danej osi a číslo 2500000000 predstavuje prírastok. V tomto prípade sme vybrali tému s názvom `theme_ipsum`, ktorá nie je súčasťou základného balíka R a na jej použitie sme museli pridať knižnicu `library(hrbrthemes)`.



Obrázok 6 Počet ľudí používajúcich internet (zdroj: ourworldindata.org , vlastné spracovanie)

Na grafe zobrazenom na obrázku 6 sú znázornené počty ľudí, ktorí používajú internet farebne rozdelené podľa jednotlivých siedmich krajín s najväčším počtom užívateľov od roku 1990 po rok 2017. Podľa grafu môžeme skonštatovať, že najviac ľudí, ktorí používajú internet na nachádza jednoznačne v Číne, čo je vzhľadom na veľkosť populácie Číny logický výsledok.

4.4 Kruhový graf

Kruhový graf je spôsob štruktúrovaného grafického znázornenia matematických alebo štatistických dát. V kruhovom grafe je dĺžka oblúka každého rezu a následne jeho stredový uhol a kruhový výsek úmerný množstvu, ktoré predstavuje. Plocha kruhu predstavuje celý objem súboru dát. Jeho jednotlivé časti zobrazujú podiel čiastkových hodnôt na celkovom objeme dát. Obvykle sú jednotlivé časti odlíšené farbou. Kruhové grafy sú používané v obchodnom svete a masmédiách. V odborných kruhoch sa odporúča presnejšie zobrazenie využitím stĺpcového, bodového, alebo líniového grafu.

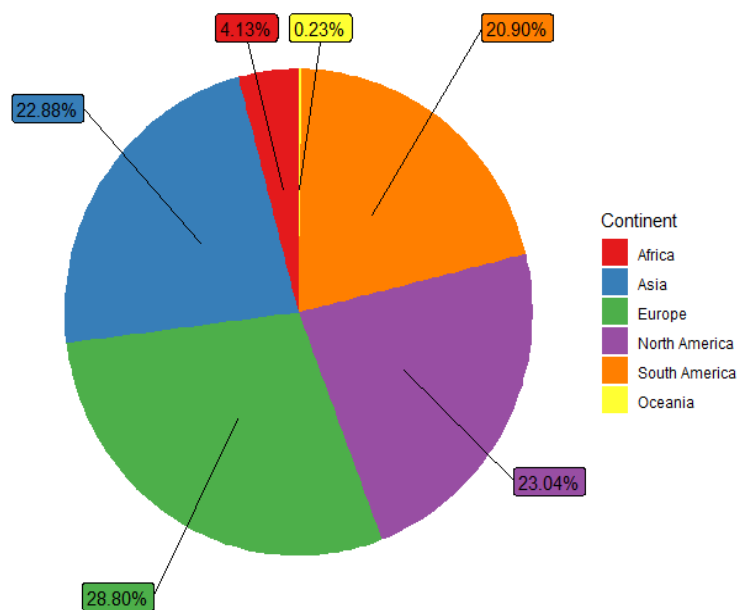
```
death %>%
  mutate(csum = rev(cumsum(rev(Pocet))),
         pos = Pocet/2 + lead(csum, 1),
         pos = if_else(is.na(pos), Pocet/2, pos),
         percentage = Pocet/sum(Pocet)) %>%
  ggplot(aes(x = "", y = Pocet, fill =
    fct_inorder(Continent)))+
  geom_col(width = 1)+
```

```

scale_fill_brewer(palette = "Set1")+
geom_label_repel(aes(y = pos,
                    label =
glue::glue("{scales::percent(percentage)}"),
                    fill = Continent),
size = 4,
nudge_x = 1,
show.legend = FALSE)+
coord_polar(theta = "y")+
labs(fill = "Continent")+
theme_void()

```

Týmto kódom sme vytvorili kruhový graf na obrázku 7. Tak isto ako v predošlých vizualizáciách sme použili balíček ggplot2. V argumente `aes()` vykresľujeme do grafu len premennú `y`, čo predstavuje počet smrť na ochorenie covid-19, ktoré budú farebne odlišené podľa jednotlivých kontinentov. Toto dosiahneme argumentom `fill = fct_inorder(Continent)`. Funkcia `fct_inorder` spôsobí, že sa jednotlivé počty smrť zobrazia v poradí v akom sú usporiadané kontinenty v našom datasete. Ďalej sa nachádza grafická vrstva `geom_col()`, ktorá sa používa pri tvorbe stĺpcových grafov, ale funkcia `coord_polar(theta = "y")` nám pretvorí stĺpcový graf na kruhový. V kóde sa tiež vyskytuje funkcia `mutate()`, ktorá slúži na pridávanie nových premenných, ale zároveň zachováva už existujúce. Vrstva `geom_label_repel()` nachádzajúca sa v balíčku `ggrepel` nám umožňuje vložiť jednotlivé popisy do grafu na vhodné pozície. Farebná škála grafu je určená jednou z veľkého množstva existujúcich farebných paliet a to konkrétne funkciou `scale_fill_brewer(palette = "Set1")`, ktorá používa paletu s názvom `Set1`. Nakoniec máme vrstvu `theme_void()`, ktorá je ideálna na vykreslenie kruhového grafu.



Obrázok 7 Graf počtu úmrtí na ochorenie Covid-19 k 20.3.2022 podľa jednotlivých kontinentov
(zdroj: ourworldindata.org, vlastné spracovanie)

Obrázok 7 zobrazuje kruhový graf celkových potvrdených úmrtí na ochorenie Covid-19 ku dňu 20.3.2022 v jednotlivých kontinentoch. Najväčší počet úmrtí je v Európe a najmenej ľudí zomrelo jednoznačne v Oceánii.

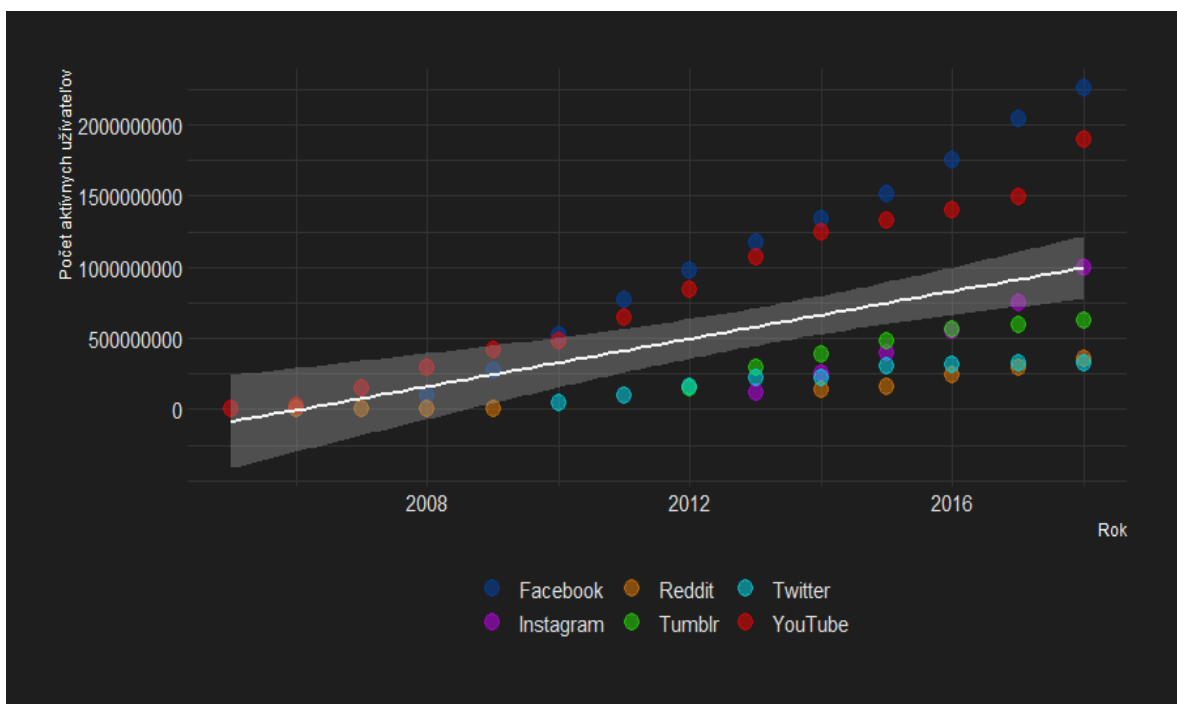
4.5 Bodový graf

Bodový graf má vždy dve osi s hodnotami, jednu množinu číselných údajov zodpovedajúcu k vodorovnej a druhú množinu číselných hodnôt k zvislej osi. Graf zobrazuje body v priesečníkoch číselných hodnôt x a y a hodnoty spája do jednotlivých údajových bodov. Podľa konvencie predstavuje os x ľubovoľné hodnoty, ktoré nezávisia od inej premennej, ktorá sa označuje ako nezávislá premenná. Hodnoty y sú umiestnené na zvislej osi a predstavujú závislú premennú.

```
social %>%
  ggplot(aes(Year, Number))+
  geom_point(alpha = 0.5, size = 4,aes(col = Entity))+
  geom_smooth(method = "lm", col = "#F2F4F4")+
  scale_color_manual(values = c("#0044AD", "#C800F5",
    "#E77C03", "#22E703", "#00E1F3", "#FF0000"))+
  scale_y_continuous(breaks = seq(0,2000000000,500000000))+
  theme_modern_rc()+
  labs(x = "Rok",
    y = "Počet aktívnych užívateľov")+
```

```
theme(legend.title = element_blank(),
      text = element_text(size = 14),
      legend.position = "bottom")
```

Na vytvorenie bodového grafu sa používa geometrická vrstva `geom_point()`. V rámci tejto funkcie sme použili už známe parametre `alpha` a `size` a tiež sme pridali tretiu premennú s názvom `Entity`, podľa ktorej budú farebne rozlíšené jednotlivé body v grafe. Táto požiadavka je definovaná parametrom `col = Entity`. Ďalej sme použili vrstvu `geom_smooth()`, ktorá vykreslí krivku pozorovaných údajov. Zvolili sme typ krivky zodpovedajúcej lineárnemu modelu, ktorá sa definuje pomocou argumentu `method = "lm"`.



Obrázok 8 Počet aktívnych užívateľov sociálnych sietí (zdroj: ourworldindata.org , vlastné spracovanie)

Graf na obrázku 8 zobrazuje celosvetový počet ľudí používajúcich rôzne sociálne siete od roku 2005 po rok 2019. Z grafu je možné vyčítať, že v súčasnosti sú najviac využívanými sociálnymi sieťami facebook, youtube a instagram.

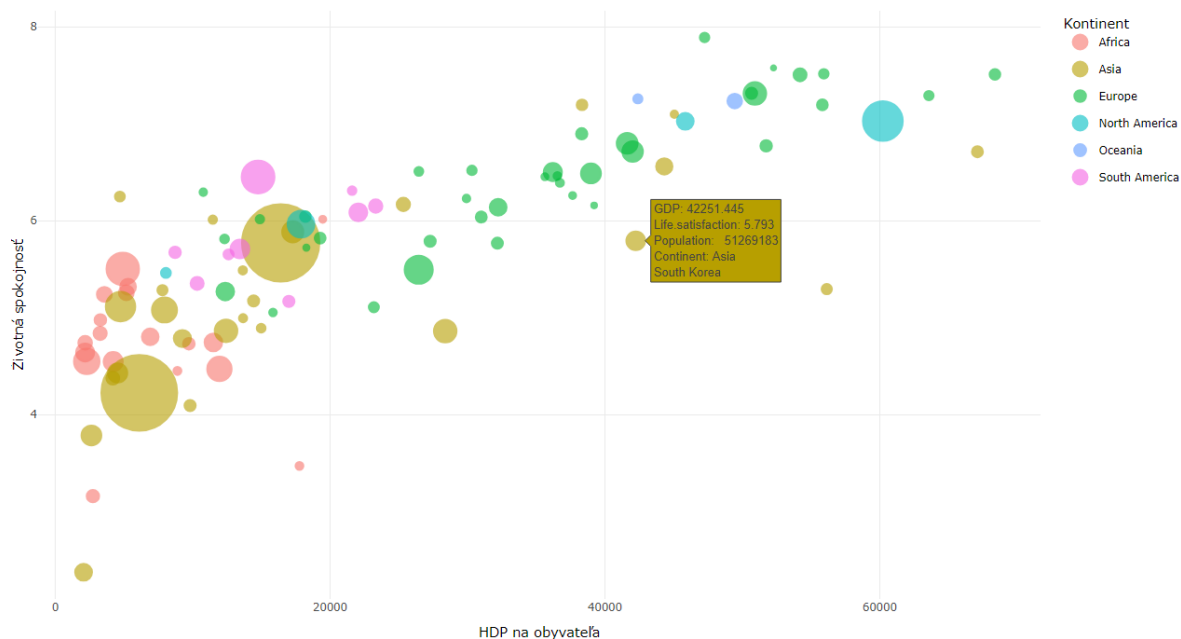
4.6 Bublinový graf

Bublinový graf je variant bodového grafu, v ktorom sa údajové body bublinami a ďalšia dimenzia údajov predstavuje veľkosť bublín. Rovnako ako v bodovom grafe bublinový graf neobsahuje os kategórií – vodorovné aj zvislé osi sú osi hodnôt. Okrem

hodnôt x a y vykreslených v bodovom grafe bublinový graf zobrazuje hodnoty x, hodnoty y a hodnoty z (veľkosť).

```
options(scipen = 999)
plot <- ggplot(gdp,aes(GDP, Life.satisfaction, size =
  Population, col = Continent, text = Entity))+
  geom_point(alpha = 0.6)+
  scale_color_discrete(name = "Kontinent",
    labels = c("Afrika", "Ázia",
      "Európa", "Severná Amerika", "Oceánia", "Južná
      Amerika"))+
  scale_size(range = c(1.5, 22),
    name = "Population")+
  labs(x = "HDP na obyvateľa",
    y = "Životná spokojnosť")+
  guides(size = "none")+
  theme_minimal()
ggplotly(plot)
```

Bublinový graf je obdobou bodového grafu a preto aj kód, ktorým sa takýto graf vytvára funguje na rovnakom princípe. Rozdiel je v tom, že sa nám tu vyskytuje ďalšia premenná, ktorá reprezentuje veľkosť jednotlivých bublín v grafe. V našom prípade je to veľkosť populácie konkrétnych krajín, ktoré sú farebne odlíšené podľa kontinentov, na ktorých sa nachádzajú. Na vykreslenie bublín používame geometrickú vrstvu `geom_point()` ako v normálnom bodovom grafe. Kvôli lepšej vizualizácii tu máme funkciu `scale_size(range = c(1.5, 22))`, ktorou určujeme veľkostný rozsah bublín. Následne sa v kóde nachádza už dobre známa funkcia `labs()`, ktorou si pomenujeme jednotlivé osy. Funkciou `guides(size = "none")` odstránime jednu z legend. Nakoniec použijeme funkciu `ggplotly()` z balíčka `plotly`, ktorá slúži na prevedenie grafu vytvoreného pomocou balíčka `ggplot2` do interaktívnej podoby. Týmto sme dosiahli, že keď prejdeme kurzorom na ľubovoľnú bublinu v grafe, tak nám vypíše jednotlivé informácie o danej krajine (názov krajiny, HDP na obyvateľa, životnú spokojnosť, veľkosť populácie a kontinent).

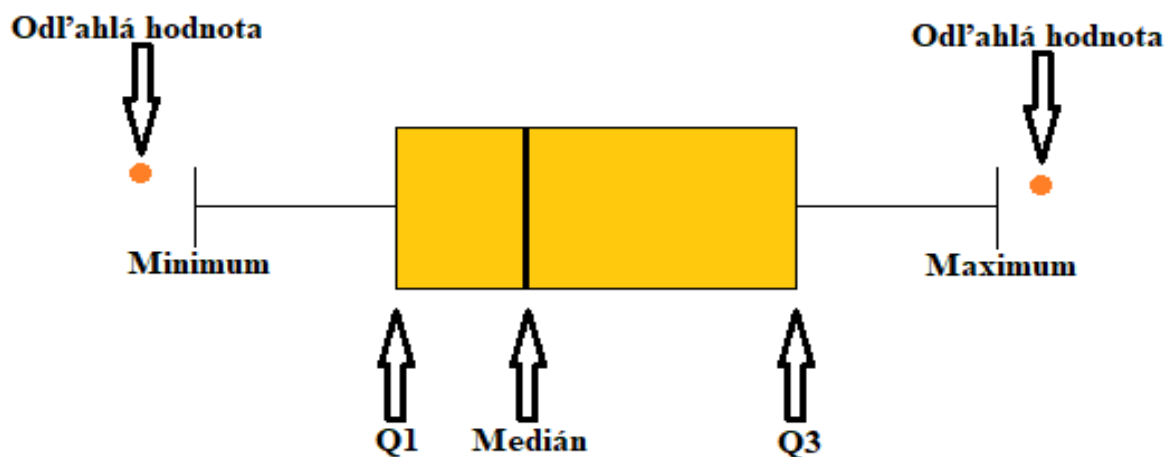


Obrázok 9 vzťah životnej spokojnosti a HDP na obyvateľa za rok 2020 (zdroj: ourworldindata.org, vlastné spracovanie)

Graf na obrázku 9 znázorňuje vzťah životnej spokojnosti a HDP na obyvateľa za rok 2020. Na vertikálnej osi je zobrazený celoštátny priemer spokojnosti so životom, na škále od 0 do 10, kde 10 je najvyššia možná životná spokojnosť. Horizontálna os zobrazuje HDP na obyvateľa upravené o infláciu a cenové rozdiely medzi krajinami.

4.7 Krabicový graf

Krabicový graf alebo box plot je jeden zo spôsobov grafickej vizualizácie numerických dát pomocou ich kvartilov. Často je vhodná a účelná súbežná vizuálna analýza viacerých krabičkových grafov naraz konštruovaných buď pre rôzne štatistické znaky jedného súboru, alebo pre jeden štatistický znak, sledovaný v rôznych štatistických súboroch. To umožňuje rýchle a prehľadné pozorovanie analyzovaných vlastností. Box ploty môžu byť vykreslené vodorovne alebo zvisle.



Obrázok 10 Schéma krabicového grafu (vlastné spracovanie)

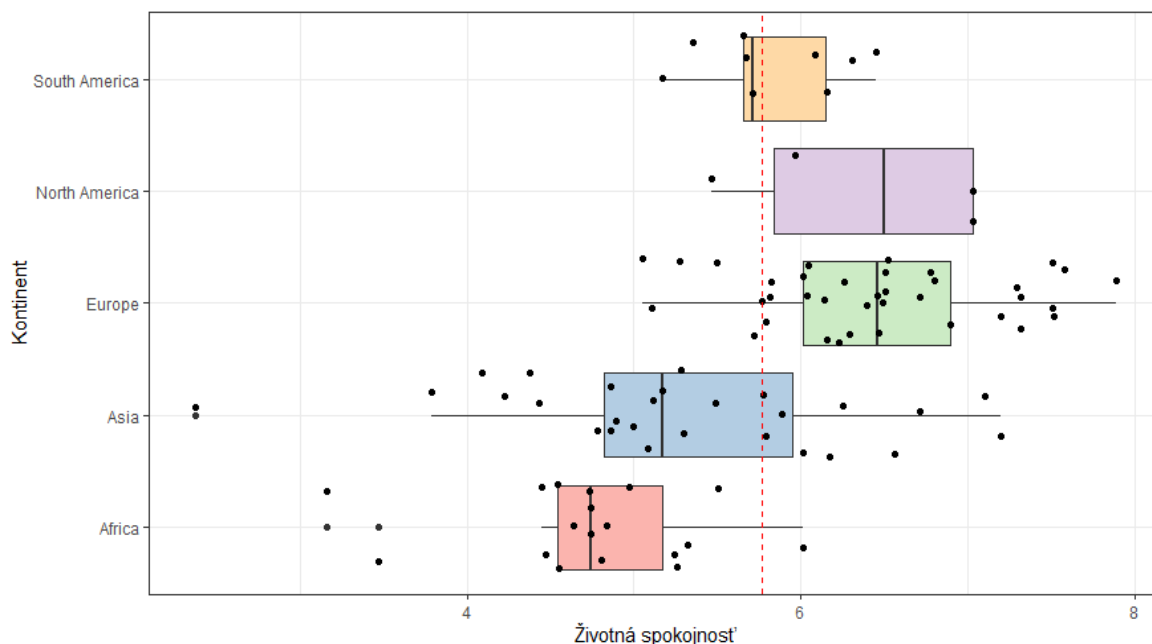
Stredná „krabicová“ časť diagramu je zhora ohraničená 3. kvartilom, zospodu 1. kvartilom a medzi nimi sa nachádza línia vymedzujúca medián. Box ploty môžu obsahovať aj línie vychádzajúce zo strednej časti diagramu kolmo hore a dole, vyjadrujúce variabilitu dát pod prvým a nad tretím kvartilom. Odláhlé hodnoty, tzv. outliery, potom môžu byť vykreslené ako jednotlivé body.

Box plot je rýchly spôsob grafického skúmania jednej alebo viacerých sád dát. Box ploty sa môžu zdať primitívnejšie ako histogramy alebo odhad hustoty jadra, ale majú niektoré výhody. Zaberajú menej miesta, a preto sú obzvlášť užitočné pre porovnávanie rozdelenia frekvencií medzi niekoľkými dátovými sadami.

```
gdp %>%
  filter(Continent %in% c("Africa", "Asia", "Europe",
"North America", "South America")) %>%
  ggplot(aes(Continent, Life.satisfaction,
             fill = Continent))+
  geom_boxplot()+
  geom_jitter()+
  scale_fill_brewer(palette = "Pastell")+
  coord_flip()+
  geom_hline(yintercept = mean(gdp$'Life.satisfaction'),
            linetype = 'dashed',
            colour = "red")+
  labs(x = "Kontinent",
       y = "Životná spokojnosť")+
  theme_bw()+
  theme(legend.position = "none")
```

Týmto kódom sme zostrojili nasledovný krabicový graf. Na jeho vytvorenie sme použili vrstvu `geom_boxplot()`, ktorá je súčasťou grafického balíčka `ggplot2`. Ďalej nasleduje geometrická vrstva `geom_jitter()`. Táto vrstva slúži na vykreslenie jednotlivých pozorovaní v podobe jednotlivých bodov, ktoré prekrývajú krabicové grafy. Pre túto vizualizáciu sme zvolili farebnú paletu s názvom `palette = "Pastell"`. Pomocou príkazu `coord_flip()` sme vymenili osy x a y. Funkciou `geom_hline()` bola do grafu pridaná horizontálna čiara (v našom prípade vertikálna vzhľadom na vymenené

osy), ktorej úroveň zodpovedá priemernej hodnote životnej spokojnosti. Typ a farbu danej čiary sme určili argumentmi `linetype = 'dashed'` a `color = "red"`.



Obrázok 11 Krabicový graf úrovne životnej spokojnosti podľa kontinentov (zdroj: ourworldindata.org, vlastné spracovanie)

Krabicové grafy zobrazené na obrázku 11 sumarizujú úroveň životnej spokojnosti (od 0 po 10) podľa jednotlivých kontinentov v roku 2020.

4.8 Husľový graf

Husľový graf zobrazuje distribúciu číselných údajov pre jednu alebo viac skupín pomocou kriviek hustoty. Šírka každej krivky zodpovedá približnej frekvencii údajových bodov v každej oblasti. Hustoty sú často sprevádzané prekrytým typom grafu, ako je napríklad krabicový graf, ktorý poskytuje ďalšie informácie.

Husľové grafy sa používajú, keď chcete sledovať rozdelenie číselných údajov, a sú obzvlášť užitočné, keď chcete porovnať rozdelenie medzi viacerými skupinami. Vrcholy, údolia a chvosty krivky hustoty každej skupiny možno porovnať, aby ste zistili, kde sú skupiny podobné alebo odlišné. Doplnkové prvky, ako sú kvartily krabicového grafu, sa často pridávajú do husľového grafu, aby poskytli ďalšie spôsoby porovnávania skupín. Husľové grafy môžu byť orientované buď vertikálnymi krivkami hustoty alebo horizontálnymi krivkami hustoty. Horizontálne orientované husľové grafy sú dobrou voľbou, keď potrebujete zobraziť napríklad dlhé názvy skupín.

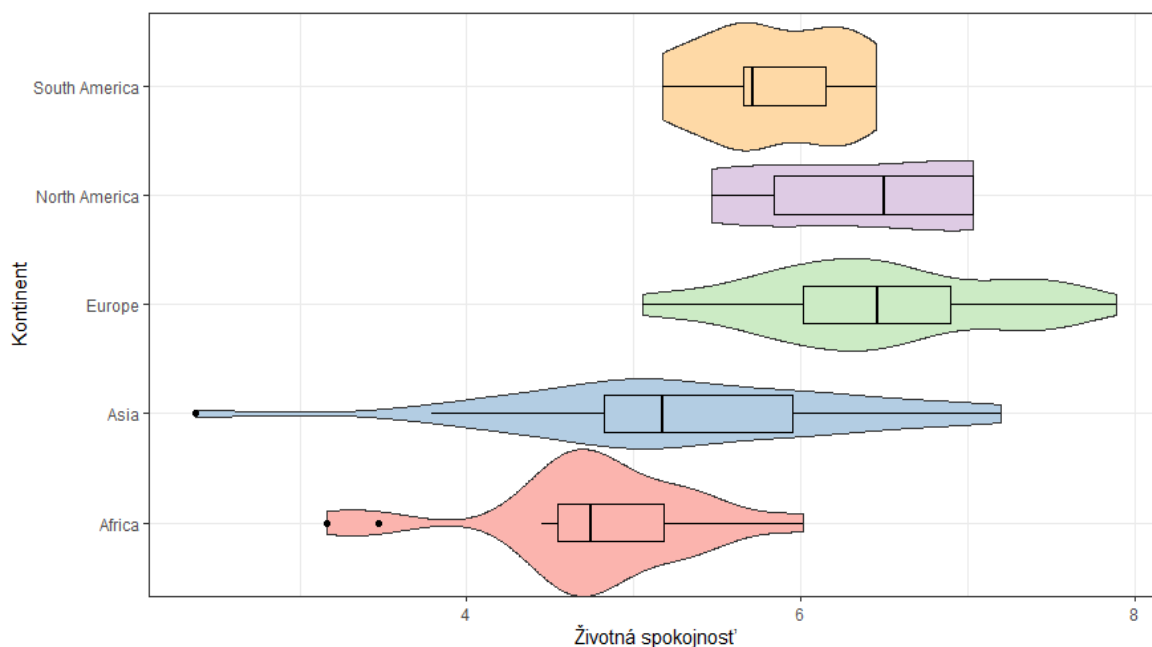
gdp %>%

```

filter(Continent %in% c("Africa", "Asia", "Europe",
"North America", "South America")) %>%
  ggplot(aes(Continent, Life.satisfaction,
             fill = Continent))+
  geom_violin(width = 1.35)+
  geom_boxplot(width = 0.35,
              color = "black")+
  scale_fill_brewer(palette = "Pastell1")+
  coord_flip()+
  labs(x = "Kontinent",
       y = "Životná spokojnosť")+
  theme_bw()+
  theme(legend.position = "none")

```

Daný kód husľového grafu je veľmi podobný predošlému kódu, ktorým bol vytvorený krabicový graf. Hlavný rozdiel je, že táto vizualizácia obsahuje aj vrstvu `geom_violin()`, ktorá slúži práve na zostrojenie husľového grafu. Parametrom `width` sme určili jednotlivé šírky konkrétnych vykreslených geometrických vrstiev.



Obrázok 12 Husľový graf (zdroj: ourworldindata.org, vlastné spracovanie)

Graf na obrázku 12 znázorňuje úroveň životnej spokojnosti (od 0 po 10) podľa jednotlivých kontinentov v roku 2020. Tvar grafu predstavuje odhad hustoty premennej.

Čím je v konkrétnom rozsahu viac dátových bodov, tým je daný graf v danom rozsahu viac rozšírený.

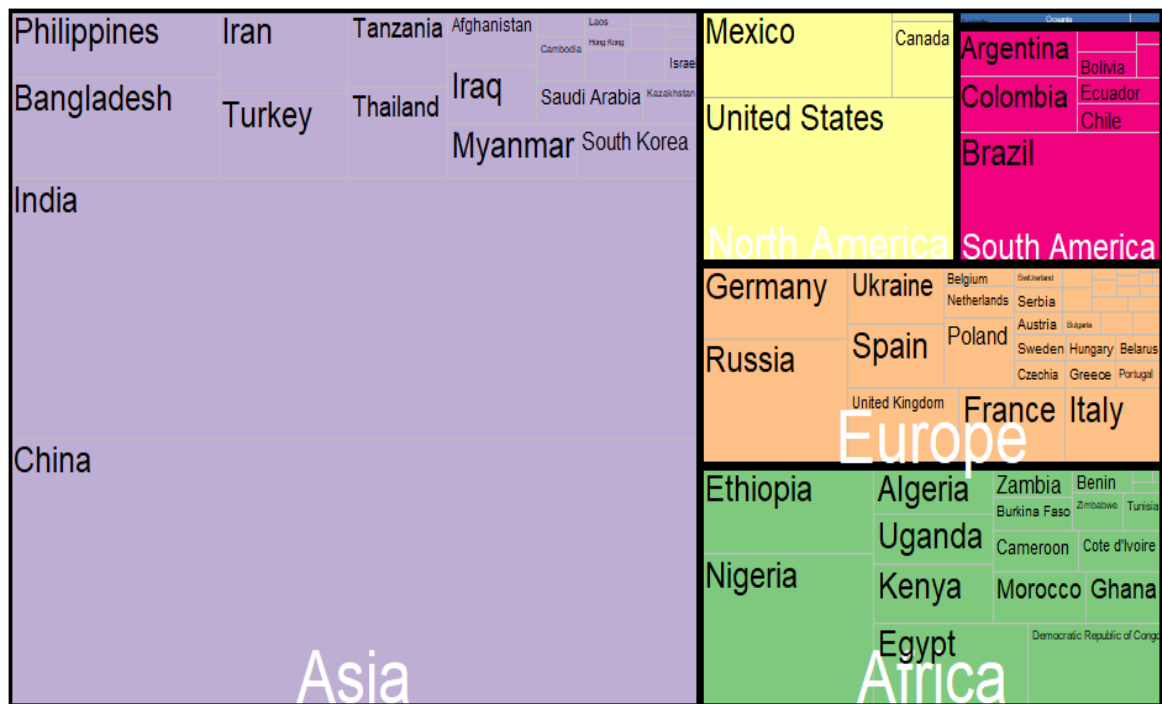
4.9 Stromový graf

Stromová mapa ponúka hierarchické zobrazenie. Vetvy stromu sú zobrazené ako obdĺžniky a všetky vedľajšie vetvy sú zobrazené ako menšie obdĺžniky. Stromová mapa zobrazuje kategórie podľa farby a vzdialenosti a dokáže prehľadne ukázať veľké množstvo údajov, v ktorých by sa v iných typoch grafu ťažko orientovalo. Stromové mapy sú vhodné na porovnanie proporcií v rámci hierarchie, tieto grafy ale nie sú úplne vhodné na zobrazenie hierarchických úrovní medzi najväčšími kategóriami a jednotlivými údajovými bodmi.

```
gdp %>%
  ggplot(aes( area = Population, fill = Continent,
              subgroup = Continent)) +
  geom_treemap() +
  geom_treemap_subgroup_border(size = 6,
                              color = "black") +
  geom_treemap_subgroup_text(color = "white") +
  geom_treemap_text(aes(label = Entity),
                   color = "black") +
  scale_fill_brewer(palette = "Accent") +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position = "none")
```

Na tvorbu stromovej mapy sa využíva geometrická vrstva `geom_treemap()`, ktorá sa nachádza v balíčku `treemapify`. Primárne rozdelenie je definované príkazom `fill = Continent`. Argument `area = Population` určuje ktorá premenná rozdeľuje stromovú mapu do menších obdĺžnikov. Ohraničenie obdĺžnikov spôsobuje funkcia `geom_treemap_subgroup_border()`, ktorá obsahuje argumenty `size = 6` a `color = "black"`. Ďalej máme funkciu `geom_treemap_subgroup_text(color = "white")`, ktorá špecifikuje farbu textu hlavnej kategórie. Následná funkcia `geom_treemap_text()`, definuje text a farbu textu v menších obdĺžnikoch. V tejto stromovej mape bola použitá farebná paleta

s názvom `palette = "Accent"`. Posledným príkazom `theme(legend.position = "none")` sme odstránili legendu, ktorá by bola v tomto prípade zbytočná.



Obrázok 13 Stromová mapa kontinentov a ich krajín podľa počtu obyvateľov (zdroj: ourworldindata.org, vlastné spracovanie)

Stromová mapa na obrázku 13 zobrazuje farebne rozlíšené rozdelenie kontinentov do jednotlivých podkategórií, ktoré sú tvorené konkrétnymi krajinami v daných kontinentoch. Veľkosť obdĺžnikov zodpovedá veľkosti populácie danej krajiny. Z grafu je možné vyčítať, že najľudnatejším kontinentom je jednoznačne Ázia. Tiež je možné porovnávať počty obyvateľov v jednotlivých krajinách. Obdĺžniky s približne rovnakou veľkosťou majú približne rovnaký počet obyvateľov.

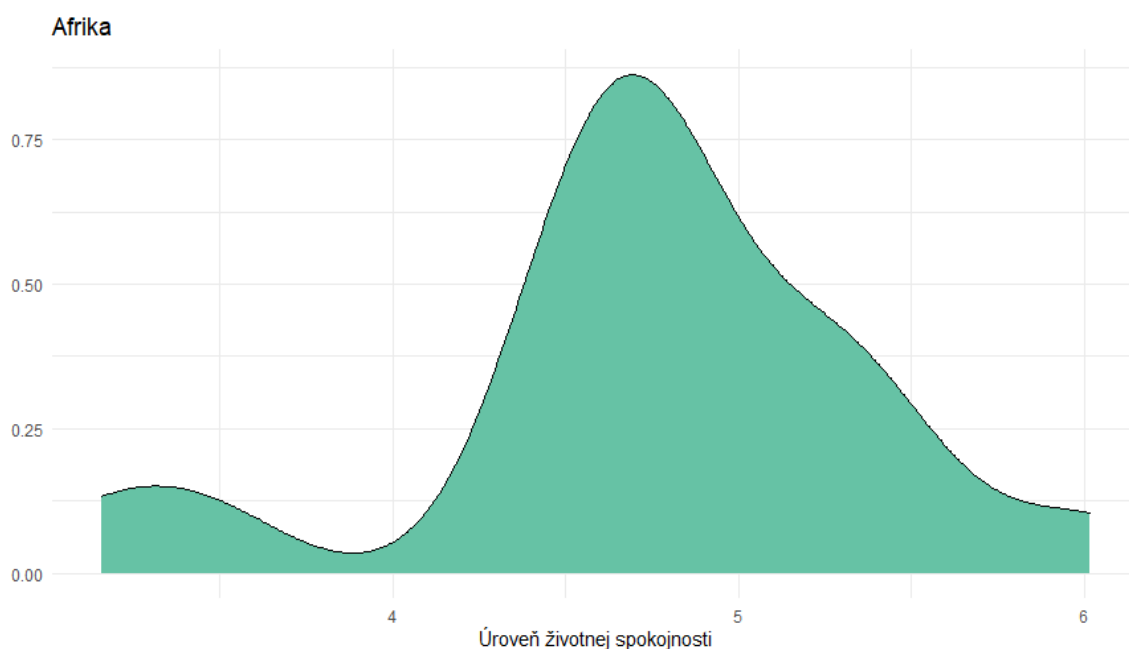
4.10 Graf hustoty

Graf hustoty je znázornením distribúcie číselnej premennej. Používa odhad hustoty jadra na zobrazenie funkcie hustoty pravdepodobnosti premennej. Ide v podstate o vyhladenú verziu histogramu a používa sa v rovnakom koncepte. Takýto graf zobrazuje vyhladené rozloženie bodov pozdĺž číselnej osi. Vrcholy grafu hustoty sú v miestach, kde je najvyššia koncentrácia bodov.

```
gdp %>%
  filter(Continent %in% c("Africa")) %>%
  ggplot(aes(x = Life.satisfaction,
             fill = Continent))+
```

```
scale_fill_brewer(palette = "Set2")+
geom_density()+
theme_minimal()+
labs(x = "Úroveň životnej spokojnosti",
      y = "",
      title = "Afrika")+
theme(legend.position = "none")
```

Tento kód slúži na vizualizovanie grafu hustoty za pomoci balička ggplot2. Na vytvorenie grafu je nutné použiť geometrickú vrstvu `geom_density()`.



Obrázok 14 Graf hustoty úrovne životnej spokojnosti v Afrike (zdroj: ourworldindata.org, vlastné spracovanie)

Na obrázku 14 je znázornený graf hustoty úrovne životnej spokojnosti na škále 0 - 10 v Afrike za rok 2020. Ako môžeme vidieť z grafu väčšina obyvateľov Afriky má úroveň životnej spokojnosti okolo hodnoty 5.

4.11 Ridgeline graf

Ridgeline je druh grafu, ktorý sa používa na vizualizáciu rozdelenia niekoľkých skupín. Každá kategória alebo skupina vytvára krivku hustoty, ktorá sa navzájom prekrýva, čím vytvára nádherný kúsok grafu.

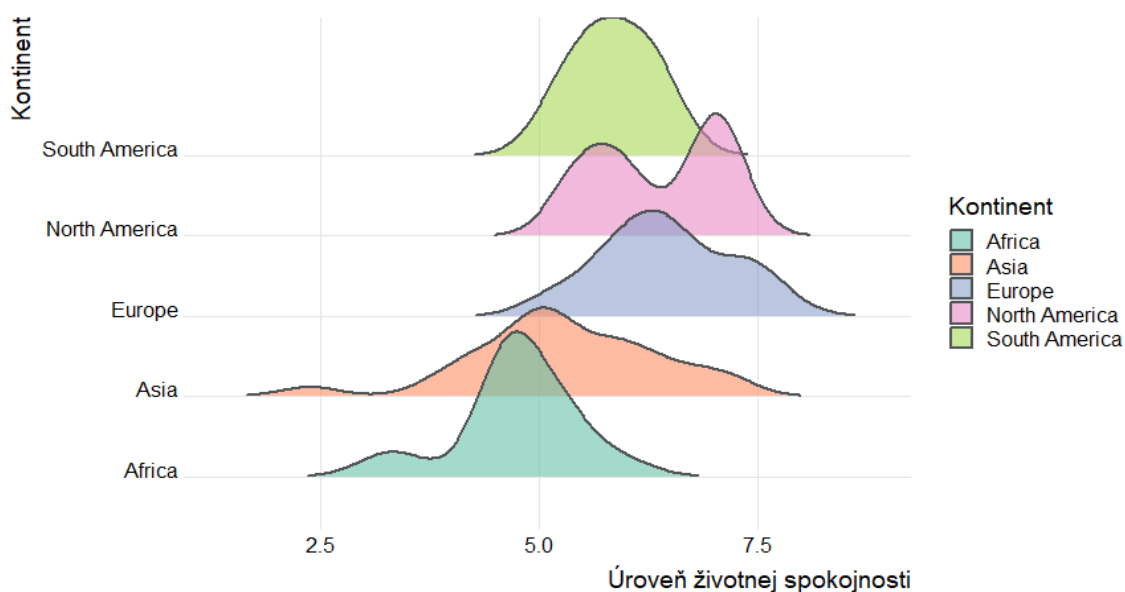
```
gdp %>%
```

```

filter(Continent %in% c("Africa", "Asia", "Europe",
"North America", "South America")) %>%
  ggplot(aes(x = Life.satisfaction, y = Continent,
             fill = Continent))+
  geom_density_ridges(rel_min_height = 0.005,
                    alpha = 0.6,
                    color = "#525557",
                    linetype = 1,
                    size = 0.8)+
  scale_fill_brewer(palette = "Set2")+
  labs(x = "Úroveň životnej spokojnosti",
       y = "Kontinent",
       fill = "Kontinent")+
  theme_ridges()

```

Pre zostrojenie tejto vizualizácie je potrebná inštalácia balíčka `ggridges`. V tomto balíčku sa nachádza funkcia `geom_density_ridges()`, ktorá umožňuje zhotovenie viacnásobných čiastočne sa prekrývajúcich hustôt pravdepodobností. Jej užitočnosť je využívaná na vizualizáciu zmien v distribúciách spojitej premennej v čase alebo priestore. Daná funkcia obsahuje argument `rel_min_height = 0.005`, ktorý zamedzuje vykresľovaniu nulových hodnôt a tiež argument `alpha = 0.6`, ktorý je potrebný kvôli prekrytiám v grafe. Tému tejto vizualizácie sme zvolili príkazom `theme_ridges()`.



Obrázok 15 Ridgeline graf (zdroj: ourworldindata.org, vlastné spracovanie)

Obrázok 15 zobrazuje ridgeline graf distribúcie úrovne životnej spokojnosti podľa jednotlivých kontinentov.

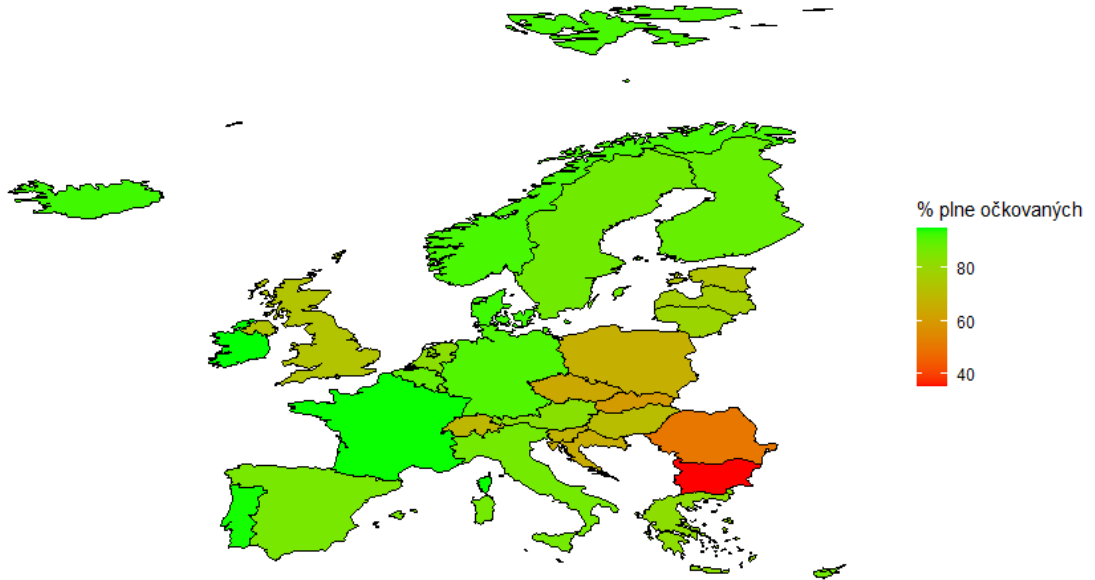
4.12 Choropletova mapa

Reprezentácia údajov prostredníctvom máp je rýchlo sa rozvíjajúca oblasť vizualizácie, ktorá má mnoho praktických aplikácií. Dnes existuje široká škála inovatívnych nástrojov na vizuálne znázornenie geografických údajov a systém R je jeden z nich. Choropletová mapa, je jedna z najpoužívanějších vyjadrovacích metód tematickej kartografie. Ide o jednoduchú mapu, v ktorej je graficky vyjadrená (farbou či rastrom) intenzita javu v sledovanom území.

```
euvac <- read.csv("D:/stiahnute subory/euvac.csv")
View(euvac)
mapdata <- map_data("world")
mapdata <- left_join(mapdata, euvac, by="region")
View(mapdata)
mapdata1 <- mapdata %>% filter(!is.na(mapdata$Vaccination))
View(mapdata1)
mapdata1 %>%
  ggplot(aes(x = long, y = lat, group=group)) +
  geom_polygon(aes(fill = Vaccination), color = "black")+
  scale_fill_gradient(name = "% plne očkovaných", low =
"red", high = "green", na.value = "grey50")+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        rect = element_blank())
```

Týmto kódom sme vytvorili konkrétnu choropletovu mapu za použitia grafického balíčka ggplot2. Na vykreslenie mapy sme použili dataset s názvom `map_data()`, ktorý je vstavaný v systéme R. Následne sme príkazom `mapdata <- left_join(mapdata, euvac, by="region")` k nemu pripojili druhý dataset s údajmi o percentuálnej zaočkovanosti populácie v jednotlivých krajinách Európy. Na

zostrojenie sme použili funkciu `geom_polygon()` s argumentom `fill = Vaccination`. Príkazom `scale_fill_gradient()` sme určili farebnú škálu danej mapy a názov legendy. Nakoniec sa v kóde nachádza vrstva `theme()`, kde sme pomocou príkazu `element_blank()` odstránili mriežkovanie, názvy osí x a y a pod.



Obrázok 16 Chorepletova mapa zobrazujúca % plne zaočkovanej populácie jednotlivých krajín Európy (zdroj: statista.com, vlastné spracovanie)

Zobrazená mapa na obrázku 16 demonštruje percentuálnu plnú zaočkovanosť populácie proti ochoreniu covid-19 v jednotlivých krajinách Európy.

Záver

Vizualizácia údajov a komunikácia s nimi je vo všeobecnosti na priesečníku vedy a umenia. Bezpochyby je v tom určitá veda, vizualizácia dát má svoje osvedčené postupy a pokyny, ktoré treba dodržiavať. Ale je tu aj umelecká zložka. To je jeden z dôvodov, prečo je oblasť dátovej vizualizácie taká zábavná a rôznorodá. Rôzni ľudia budú pristupovať k veciam rôznymi spôsobmi a prídu s odlišnými riešeniami rovnakej výzvy vizualizácie údajov. Neexistuje jediná „správna“ odpoveď. Skôr často existuje viacero potenciálnych ciest na efektívnu komunikáciu s údajmi.

Dnes žijeme vo svete, kde sa každú sekundu generujú tony údajov. Tieto údaje potrebujeme analyzovať, aby sme získali nejaký užitočný prehľad. Jednou z najsilnejších zbraní pre prehľad údajov je vizualizácia údajov. Pravdepodobne už každý počul vetu: „Obrázok povie viac ako tisíc slov dohromady“. Preto, aby sme rozprávali príbehy z dát, potrebujeme nástroje na výrobu adekvátnej a úžasnej grafiky.

Táto bakalárska práca bola zameraná na vizualizáciu údajov v systéme R. Prvá kapitola charakterizovala pojem vizualizácia údajov, jej počiatky a význam v súčasnom svete veľkých dát. Ďalej sme priblížili programovací systém R. Toto prostredie je právom považované za veľmi schopný nástroj v oblasti matematicko-štatistických operácií. Jeho veľkou výhodou je, že na rozdiel od ostatných drahých komerčných softvérov je R open source projekt bez nutnosti poplatku za licenciu. Ďalším obrovským plusom je, že oproti ostatným obmedzeným softvérom, nie je pri využívaní algoritmov, respektíve metód, ktoré nie sú implementované v jeho prostredí, potrebná kombinácia viacerých programov. Používateľ má možnosť veľmi jednoduchým spôsobom doinštalovať veľké množstvo rôznych balíkov a funkcií, ktoré sú zamerané na tvorbu grafov, a to ako bežných, tak aj interaktívnych. Niektoré z nich boli špecifikované v tejto kapitole. V ďalšej časti sme sa oboznámili s prostredím RStudio, ktoré sme aktívne používali pri písaní tejto bakalárskej práce. Podľa môjho názoru je toto prostredie najlepším softvérom na prácu s jazykom R.

Hlavný nástroj, ktorý bol využívaný pri tvorbe takmer všetkých vizualizácií v tejto práci je balík `ggplot2`. Tento balíček patrí k najpoužívanejším a najlepšie udržiavaným v rámci systému R. Použitím tohto nástroja sa stáva vytváranie vizualizácií vysoko efektívnym a intuitívnym. Ďalšia grafická knižnica využitá v tejto práci je balík s názvom `plotly`, vďaka ktorému je možné vytvárať interaktívne grafy publikačnej kvality. Veľkým plusom je tiež kompatibilita medzi týmito dvoma balíkmi.

Nasledujúca časť bola venovaná tvorbe grafov. Hlavným faktorom pri tvorbe grafov je typ zobrazovaných premenných. Vďaka tomu vieme určiť aký druh vizualizácie bude najvhodnejší. V systéme R je možné vytvárať veľké množstvo reprezentatívnych vizualizácií rôznych typov a kombinácií. Táto kapitola obsahuje stručný popis jednotlivých typov grafov a tiež vysvetlený kód, ktorým bol konkrétny typ grafu vytvorený. Táto časť môže slúžiť ako návod pri zostrojovaní podobných vizualizácií v jazyku R. Všetky grafy v tejto práci boli vytvorené na reálnych dátach z rôznych oblastí, ktoré museli byť vopred upravené a prevedené na požadovaný formát. Dôležitá súčasť pri tvorbe grafov je tiež zvolenie správnych farebných odtieňov, vhodný výber pozadia, umiestnenie legend a prispôsobenie veľkostí zobrazovaných dát.

Správna vizualizácia údajov je právom považovaná za umenie. Musí byť pútavá, presná, efektívna a pochopiteľná ľuďom, pre ktorých je určená. Dobré vizualizácie vyžadujú neustálu prax a tiež predstavivosť.

Zoznam použitej literatúry

1. BACIGÁL, Tomáš. Úvod do analýzy údajov pomocou R [elektronický zdroj]. 17.2.2022 Dostupné na: <https://bookdown.org/content/8ab7947b-f480-4e49-a28c-32c1970beb76/>
2. BOOKER, Christopher. Seven Basic Plots, Anglicko: Bloomsbury, 2019. 736 s. ISBN 978-14-7297-618-5
3. CAMPBELL, Matthew. Learn RStudio IDE: Quick, Effective, and Productive Data Science, 1. vyd. USA: Apress, 2019. 162 s. ISBN 978-14-8424-510-1
4. DETKO, Marek. Interaktivní vizualizace dat a metadat [elektronický zdroj]. 16.5.2017 Dostupné na: <https://www.cognito.cz/design/interaktivni-vizualizace-dat-a-metadat>
5. FARNWORTH, Richard. A Short History of Data Visualisation [elektronický zdroj]. 3.7.2020 Dostupné na: <https://towardsdatascience.com/a-short-history-of-data-visualisation-de2f81ed0b23>
6. FIELD, Andy a kol. Discovering Statistics Using R, 1. vyd. USA: SAGE Publications Ltd, 2012. 992 s. ISBN 978-14-4620-046-9
7. CHANG, Winston. R Graphics Cookbook, 2. vyd. USA: O'Reilly Media, 2018. 444 s. ISBN 978-14-9197-860-3
8. KVASNIČKA, Michal – MIKULA, Štěpán. Analýza a vizualizace dat v jazyce R [elektronický zdroj], 2021. 341 s. Dostupné na: https://is.muni.cz/el/econ/podzim2017/BPE_AVED/um/kniha/aved.pdf
9. KUMAR, Mohit. R Overview and history [elektronický zdroj]. 12.8.2020 Dostupné na: <https://medium.com/@ArtisOne/r-overview-and-history-75ecb036d0df>
10. MURRELL, Paul. R Graphics, 3. vyd. Anglicko: Chapman and Hall/CRC, 2018. 441 s. ISBN 978-14-9878-905-9
11. WICKHAM, Hadley a kol. Ggplot2: Elegant Graphics for Data Analysis, 3. vyd. Nemecko: Springer, 2010. 213 s. ISBN 978-03-8798-140-6
12. WILKE, Claus O. Fundamentals of Data Visualization - A Primer on Making Informative and Compelling Figures, 1. vyd. USA: O'Reilly Media, 2019. 390 s. ISBN 978-14-9203-108-6

13. PENG, Roger. R Programming for Data Science, 5. vyd. USA: lulu.com, 2016. 194 s. ISBN 978-13-6505-682-6
14. RAHLF, Thomas. Data visualisation with R 2. vyd. Nemecko: Springer, 2019. 471 s. ISBN 978-30-3028-443-5
15. VERZANI, John. Getting Started with RStudio: An Integrated Development Environment for R, 1. vyd. USA: O'Reilly Media, 2011. 98 s. ISBN 978-14-4930-903-9
16. McLEOD, Saul. What does a box plot tell you? [elektronický zdroj]. 2019
Dostupné na: <https://www.simplypsychology.org/boxplots.html>