

**EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY**

Evidenčné číslo: 103006/B/2024/36122176491842308

**POKROČILÉ METÓDY VIZUALIZÁCIE
A PREZENTÁCIE VÝSTUPOV V JAZYKU R NA
AKTUÁRSKYCH DÁTACH**

Diplomová práca

**EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY**

**POKROČILÉ METÓDY VIZUALIZÁCIE
A PREZENTÁCIE VÝSTUPOV V JAZYKU R NA
AKTUÁRSKYCH DÁTACH**

Diplomová práca

Študijný program: Aktuárstvo
Študijný odbor: Ekonómia a manažment
Školiace pracovisko: Katedra matematiky a aktuárstva
Vedúci záverečnej práce: doc. Ing. Michal Páleš, PhD.

Bratislava 2024

Bc. Simona Buchancová

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracovala samostatne, a že som uviedla všetku použitú literatúru.

V Bratislave dňa

.....

Simona Buchancová

Pod'akovanie

Ďakujem svojmu školiteľ'ovi doc. Ing. Michalovi Pálešovi, PhD., za cenné rady a odborné odporúčania pri písaní mojej záverečnej práce. Rada by som vyjadrila poďakovanie aj spoločnosti Zurich Insurance Company Ltd za poskytnutú podporu a príležitosti, ktoré mi boli počas práce na tejto diplomovej práci dostupné. Špeciálna vďaka patrí Branislavovi Kalinovi, ktorý svojimi radami značne prispel k úspešnému dokončeniu tejto práce.



ABSTRAKT

BUCHANCOVÁ, Simona: *Pokročilé metódy vizualizácie a prezentácie výstupov v jazyku R na aktuárskych dátach*. – Ekonomická univerzita. Fakulta hospodárskej informatiky: Katedra matematiky a aktuárstva – Vedúci záverečnej práce: doc. Ing. Michal Páleš, PhD. – Bratislava, FHI EU, 2024, 146 strán

Práca sa zaoberá pokročilými metódami vizualizácie a prezentácie dát, konkrétne so zameraním na využitie jazyka R pre aktuársku analýzu dát. Ponorí sa do vývoja vizualizácie údajov, najnovších pokrokov v tejto oblasti a významu interaktívnych dátových nástrojov v akademickom výskume. Cieľom výskumu je zlepšiť pochopenie a prezentáciu aktuárskych údajov prostredníctvom inovatívnych vizualizačných techník. Na dosiahnutie tohto cieľa práca skúma balíky R, ako sú *gganimate*, *plotly*, R Markdown a Shiny. Tieto nástroje sa používajú na vytváranie interaktívnej a animovanej grafiky, vytváranie vysokokvalitných správ a vývoj interaktívnych webových aplikácií. Začlenením princípov vedy o údajoch a analýzy údajov sa práca zameriava na poskytnutie prehľadu o prekonaní zložitosti údajov a chaosu. Prostredníctvom komplexnej analýzy vizualizačných metodológií a ich aplikácie v poistno-matematických kontextoch táto diplomová práca prispieva k rozširovaniu poznatkov v oblasti vizualizácie dát a jej praktických implementácií pre aktuársku vedu.

Kľúčové slová: vizualizácia, jazyk R, aktuárske údaje, *gganimate*, *plotly*, R Markdown, Shiny

ABSTRACT

BUCHANCOVÁ, Simona: *Advanced methods of visualization and presentation of outputs in R language on actuarial data.* – University of Economics in Bratislava. Faculty of Economic Informatics: Department of Mathematics and Actuarial Science – Thesis supervisor: doc. Ing. Michal Páleš, PhD. – Bratislava, FHI EU, 2024, 146 pages

The work deals with advanced methods of visualization and presentation of data, specifically focusing on the use of the R language for actuarial data analysis. The study dives into the evolution of data visualization, the latest advances in this field, and the significance of interactive data tools in academic research. The goal of the research is to improve understanding and presentation of actuarial data through innovative visualization techniques. To achieve this objective, the study examines R packages such as *gganimate*, *plotly*, R Markdown, and Shiny. These tools are used to create interactive and animated graphics, produce high-quality reports, and develop interactive web applications. By incorporating principles of data science and data analysis, the thesis aims to provide an overview of overcoming data complexity and chaos. Through a comprehensive analysis of visualization methodologies and their application in insurance mathematical contexts, this thesis contributes to the expansion of knowledge in the field of data visualization and its practical implementations for actuarial science.

Keywords: visualization, R language, actuarial data, *gganimate*, *plotly*, R Markdown, Shiny

Obsah

ÚVOD	9
1 SÚČASNÝ STAV	10
1.1 NÁHLED DO ZAČIATKOV VIZUALIZÁCIE ÚDAJOV	10
1.1.1 Významné milníky v histórii vizualizácie údajov	11
1.1.2 Vývoj inovácií vo vizualizácii údajov	11
1.2 VIZUALIZÁCIA A SÚČASNOSŤ	12
1.2.1 Nedávne pokroky vo výskume vizualizácie údajov	15
1.3 VIZUALIZÁCIA V JAZYKU R	16
1.4 INTERAKTÍVNE DÁTOVÉ NÁSTROJE A AKADEMICKÝ VÝSKUM	18
2 CIEĽ PRÁCE	22
3 METODIKA PRÁCE	23
3.1 DATA SCIENCE	23
3.2 DATA SCIENCE PROCES	24
3.3 DÁTOVÁ ANALYTIKA	26
3.3.1 Nástroje na analýzu údajov	26
3.4 AKTUÁRSTVO	27
3.5 VIZUALIZÁCIA DÁT	28
3.5.1 Grafy	29
3.5.2 Čo je animácia?	30
3.5.3 Cieľ a účel vizualizácie	30
3.6 PRÍSTUPY AKÝMI MOŽNO PREKONAŤ CHAOS V ÚDAJOCH	31
3.6.1 Storytelling a jeho základné techniky	31
3.6.2 Informačná grafika	36
3.7 JAZYK R AKO NÁSTROJ NA VIZUALIZÁCIU ÚDAJOV	39
3.8 R GG PLOT2	42
3.8.1 Syntax ggplot2 a prispôsobenie grafu	43
3.8.2 Proces tvorby statického grafu pomocou ggplot2	44
3.8.3 Zobrazenie grafov ggplot2 a ich výstupný formát	46
3.9 R PLOTLY	47
3.9.1 Syntax R plotly a prispôsobenie grafu	49
3.9.2 Proces tvorby interaktívneho grafu pomocou R plotly	50
3.9.3 Zobrazenie a výstupný formát grafu R plotly	54
3.10 R GG ANIMATE	55
3.10.1 Syntax R gganimate a prispôsobenie animácie	56
3.10.2 Inštalácia balíka R gganimate	60
3.10.3 Vykresľovanie a výstup objektu R gganimate	61
3.11 R MARKDOWN	63
3.11.1 R Markdown vo formáte PowerPoint prezentácie	65
3.11.2 R Markdown dokument a jeho štruktúra	65
3.11.3 Code Chunks	67
3.11.4 Štruktúra prezentácie	67
3.11.5 Pandoc	68
3.12 R SHINY	69
3.12.1 Štruktúra UI	70
3.12.2 Vstupy UI	72
3.12.3 Serverová logika R Shiny	73

4	VÝSLEDKY PRÁCE A DISKUSIA	75
4.1	PRAKTICKÁ UKÁŽKA R <i>PLOTLY</i>	76
4.1.1	Vizualizácia údajov úmrtnosti na srdcové choroby.....	76
4.1.2	Vizualizácia údajov úmrtnosti pri dopravných nehodách.....	80
4.1.3	Vizualizácia nasimulovaných dát.....	82
4.1.4	Interaktívna prezentácia.....	87
4.2	PRAKTICKÁ UKÁŽKA R <i>GGANIMATE</i>	89
4.2.1	Vizualizácia údajov úmrtnosti pri dopravných nehodách.....	89
4.2.2	Vizualizácia údajov úmrtnosti na srdcové choroby.....	90
4.2.3	Vizualizácia údajov obetí na cestách.....	92
4.2.4	Vizualizácia nasimulovaných dát.....	93
4.3	PRAKTICKÁ UKÁŽKA R <i>MARKDOWN</i>	100
4.3.1	Tvorba grafu: Miera úmrtnosti jednotlivých krajín.....	102
4.3.2	Tvorba grafu: Výskyt rakoviny v jednotlivých krajinách.....	103
4.3.3	Tvorba grafu: Analýza úmrtnosti krajín v priebehu času	104
4.4	PRAKTICKÁ UKÁŽKA R <i>SHINY</i>	108
	ZÁVER	117
	ZOZNAM POUŽITEJ LITERATÚRY	119
	PRÍLOHY	123

ÚVOD

V súčasnej dobe, v ktorej sa množstvo dostupných dát neustále zväčšuje a ich analýza a interpretácia sú kľúčové pre rozhodovanie v rôznych odvetviach sa vizualizácia stáva nepostrádateľným nástrojom pre efektívne spracovanie informácií. V tomto kontexte sa vizualizácia javí ako kľúčový nástroj, ktorý jednotlivcom umožňuje získať cenné poznatky z množstva informácií, ktoré majú k dispozícii.

Predložená práca je venovaná skúmaniu pokročilých techník vizualizácie a prezentácie údajov, konkrétne v oblasti aktuárskych údajov, s využitím všestranného programovacieho jazyka R. Zameriava sa teda na analýzu techník vizualizácie dát s cieľom podporiť efektívne rozhodovanie a porozumenie informáciám.

Hlavným zámerom tejto práce je hlbšie sa ponoriť do moderných prístupov vizualizácie a jej aplikácii v rámci aktuárstva. Zameraním sa na využitie jazyka R, populárneho medzi odborníkmi v oblasti štatistiky a analýzy údajov, pre efektívne vizualizovanie zložitých aktuárskych informácií, sa naša práca nezameriava len na samotné techniky vizualizácie, ale aj na spôsoby, ako tieto vizualizácie efektívne prezentovať a interpretovať pre rôzne cieľové skupiny.

V nasledujúcich častiach sa zameriame na súčasný stav vizualizácie, preskúmame historické mílniky, najnovší vývoj vo výskume a význam vizualizácie v jazyku R. Práca zahŕňa procesy spracovania údajov, analýzu údajov a využitie pokročilých balíkov v R, ako napríklad aplikácie `ggplot2`, `plotly`, `gganimate`, `R Markdown` a `Shiny`.

Impulzom tejto práce je preklenúť priepasť medzi niekedy zložitými poistno-matickými údajmi a jasnými a dostupnými vizualizáciami, ktoré pomáhajú aktuárom a iným zainteresovaným stranám pri prijímaní relevantných rozhodnutí. Využitím sily pokročilých metód vizualizácie v jazyku R je cieľom tejto práce prispieť k rozvoju poznatkov založených na údajoch v aktuárstve.

Celkovo práca poskytuje pohľad na dôležitosť aplikácie techník vizualizácie dát na efektívnu komunikáciu a analýzu komplexných aktuárskych dát. Ciele, metodológia a výsledky tohto výskumu budú rozpracované v nasledujúcich častiach, ktoré objasnia potenciál vizualizácie údajov v aktuárstve.

1 Súčasný stav

1.1 Náhľad do začiatkov vizualizácie údajov

Grafická prezentácia informácií bola vždy súčasťou civilizovanej kultúry. Pred vznikom písomných slov existovali piktogramy, jaskynné maľby a hieroglyfy. Archeologické objavy naznačujú, že symbolické obrazy používané starovekými Egypťanmi ako forma písomnej komunikácie predstavujú najstaršie formy písma. Staroveké kultúry v Číne, Mezopotámii a Amerike tiež využívali obrazy na komunikáciu pred vznikom pôvodných abeced a vývojom moderných systémov písaného jazyka.

Ako sa ľudské poznávanie rozvíjalo počas renesancie a obdobia osvietenstva a rozumu, tak sa postupne vyvíjalo aj využívanie máp, grafov a diagramov ako metódy na zaznamenávanie dôležitých vedeckých, ekonomických a sociálnych údajov. Neskôr noviny čoraz viac integrovali ilustrácie, grafy a mapy do spravodajstva o kľúčových udalostiach.

Tiež vynález tlačiarenskeho stroja v roku 1886 viedol k väčšej a detailnejšej informačnej grafike a ilustráciám v novinách. Počas obidvoch svetových vojen americké noviny často publikovali mapy, a v 60. a 70. rokoch sa v drvivej väčšine amerických novín začali denne objavovať grafy a diagramy.

Avšak počítač Macintosh, ktorý sa objavil na začiatku 80. rokov, vyvolal skutočnú explóziu informačnej grafiky. Mac spolu so sofistikovaným grafickým softvérom – niektoré sa stále používajú, iné sú teraz spomienkou – umožnili grafickým dizajnérom jednoducho vytvárať podrobné mapy, grafy a diagramy v stanovenom termíne.

Lídrom hnutia vizuálneho rozprávania bol USA Today¹. Jeho redakčná misia bola jednoduchá: uspokojiť čitateľa s obmedzeným časom ponúkaním stručne upravených balíkov príbehov v zábavnom a ľahko čitateľnom formáte. To zahŕňalo kratšie príbehy, kreatívne používanie farieb a časté využívanie máp, grafov, prieskumov verejnej mienky a iných farebných grafík namiesto tradičných, dlhších textov.

V USA Today bola grafika pravidelne vytváraná ako samostatné príbehy a tiež ako sprievodca textových článkov. Tento prístup čoskoro nasledovalo mnoho ďalších novín a časopisov, pričom informačná grafika sa stala štandardnou metódou pre vizuálne vyjadrenie informácií. [2], [15]

¹ USA Today je jednou z popredných spravodajských organizácií v Spojených štátoch, ktorá je známa svojim záväzkom poskytovať komplexné a pútavé spravodajstvo.

1.1.1 Významné milníky v histórii vizualizácie údajov

Komerčný a politický atlas Williama Playfair, publikovaný v roku 1801, je jedným z prvých príkladov systematickej vizualizácie údajov. Playfair predstavil niekoľko prelomových grafických techník, vrátane čiarových grafov, stĺpcových grafov a koláčových grafov, ktoré reprezentujú ekonomické a štatistické údaje. Jeho vizualizácie poskytovali jasné a stručné zobrazenie zložitých informácií, čím ich sprístupňovali širšiemu publiku a uľahčovali pohľad na ekonomické trendy, obchodné vzorce a demografické štatistiky. Inovatívne využitie grafických reprezentácií Playfair položilo základy modernej štatistickej grafiky a demonštrovalo silu vizuálneho rozprávania príbehov pri sprostredkovaní kvantitatívnych informácií.

Florence Nightingale, známa ako zakladateľka moderného ošetrovateľstva, využila vizualizáciu údajov na obhajobu reforiem zdravotníctva počas krymskej vojny. V roku 1858 Nightingale vytvorila „ružový diagram“, aby ilustrovala sezónne variácie úmrtnosti medzi vojakmi vo vojenských nemocniciach. Diagram pripomínajúci ružu s okvetnými lístkami rôznej šírky vizualizoval významný vplyv chorôb, ktorým sa dá predchádzať, a zlých hygienických podmienok na úmrtia vojakov. Vizualizácia Nightingale účinne komunikovala naliehavosť implementácie sanitačných opatrení a podnietila reformy v praxi vojenského zdravotníctva, čo v konečnom dôsledku zachránilo nespočetné množstvo životov.

Charles Minard, francúzsky stavebný inžinier, vytvoril pozoruhodnú vizualizáciu zobrazujúcu Napoleonovu nešťastnú inváziu do Ruska v roku 1812. Minardova mapa toku, publikovaná v roku 1869, skombinovala geografické a štatistické údaje, aby rozprávala príbeh Napoleonovej katastrofálnej kampane. Vizualizácia obsahuje hrubú čiaru predstavujúcu veľkosť Napoleonovej armády, keď postupuje do Ruska a ustupuje späť, pričom vetvy označujú smer a teplotu zaznamenanú počas cesty. Použitím rôznych širok čiar Minard efektívne zobrazil dramatické straty na životoch, ktoré utrpeli Napoleonove jednotky v dôsledku bitiek, chorôb a drsných poveternostných podmienok. Minardova toková mapa zostáva oslavovaná pre svoju schopnosť sprostredkovať viacero vrstiev informácií v jedinej, vizuálne podmanivej reprezentácii, čím stanovuje štandard pre vizualizáciu dát založenú na rozprávaní. [30]

1.1.2 Vývoj inovácií vo vizualizácii údajov

Nástroje ranej vizualizácie (60. – 70. roky 20. storočia): Vytvorenie prvého počítačom generovaného grafu v roku 1961 znamenalo významný míľnik. V tomto období sa v 70-

tych rokoch minulého storočia vyvinul softvér na interaktívnu vizualizáciu, ktorý po prvýkrát umožnil výskumníkom manipulovať a analyzovať dáta vizuálne.

Vzostup Business Intelligence (1980-1990): Počas 80. a 90. rokov 20. storočia vznik nástrojov Business Intelligence (BI) transformoval vizualizáciu údajov. Tieto nástroje uľahčili vytváranie a šírenie grafických reprezentácií obchodných údajov, čo pomáha pri rozhodovacích procesoch.

Internetová éra a webová vizualizácia (1990-2000): Príchod World Wide Web umožnil vývoj webových nástrojov na vizualizáciu údajov. V tejto ére sa zaviedli dynamické a interaktívne vizualizácie, ku ktorým bolo možné pristupovať prostredníctvom webových prehliadačov, čím sa výrazne zvýšila dostupnosť údajov.

Big Data a pokročilá vizuálna analýza (2010-súčasnosť): Explózia veľkých dát viedla k pokroku vo vizualizačných technológiách na spracovanie zložitých, rozsiahlych súborov údajov. Nástroje ako Tableau, Qlik a Power BI v kombinácii s programovacími jazykmi s otvoreným zdrojovým kódom, ako sú Python a R, sa stali ústrednými pri získavaní prehľadov z veľkých údajov prostredníctvom pokročilej vizuálnej analýzy.

Vek AI a strojové učenie (2010-súčasnosť): Modely AI a strojového učenia sa teraz používajú na automatizáciu vytvárania vizualizácií údajov a na odhaľovanie vzorov a poznatkov, ktoré by ľudia len ťažko našli bez pomoci. Integrácia AI s nástrojmi na vizualizáciu údajov umožňuje prediktívnu analýzu a sofistikovanejšiu interpretáciu súborov údajov. [24]

1.2 Vizualizácia a súčasnosť

Dnes sme svedkami zvýšeného používania vizualizácie údajov v rozličných oblastiach. V žurnalizme, vzdelávaní a pri šírení informácií verejnosti, ako aj v profesionálnom prostredí, sú bežné rôzne typy grafiky, máp a vizualizácií, slúžiace na vysvetlenie, presvedčanie a rozprávanie príbehov. V najlepších prípadoch môžu vizuálne prezentácie štatistických a iných často kvantitatívnych dát ponúknuť komplexné informácie a trendy rýchlo a účinne.

Na druhej strane však môžu byť mätúce alebo dokonca manipulatívne. V ére, keď sa vytvára, zdieľa a sprístupňuje čoraz viac dát online a digitálne nástroje uľahčujú ich vizualizáciu, je kľúčové sa venovať podmienkam, za ktorých sú tieto vizuálne informácie vytvárané, distribuované a analyzované, pretože prispievajú k procesom porozumenia, učenia a angažovanosti.

Grafické znázornenie dát nie je nič nové. Naše súčasné spoločnosti sa rozvíjajú v prostredí, kde vizualizácia dát nadobúda sociálny a kultúrny význam, ako nikdy predtým, čo je čiastočne výsledkom toho, že dáta sú čoraz viac cenenejšie a dôveryhodnejšie, pretože zohrávajú kľúčovú úlohu v rozhodovaní a chápaní sveta. Dá sa povedať, že nikdy predtým nebol prístup k dátam a ich generovanie, zhromažďovanie a uchovávanie také jednoduché a rozšírené.

Hlavným stimulom pre rozširovanie vizualizácií údajov je rozvoj príslušných technológií. Inovácie v nástrojoch a technikách pre zbierku, filtrovanie, analýzu a vizualizáciu údajov robia tieto procesy jednoduchšími a cenovo dostupnejšími. Svedčíme tiež vzniku nových priestorov pre šírenie a interakciu s údajmi, pretože dátové techniky nachádzajú čoraz väčšie uplatnenie v rôznych sektoroch.

Namiesto statických obrázkov alebo grafov sú čoraz bežnejšie interaktívne vizualizácie. Používatelia môžu manipulovať s parametrami, približovať alebo oddiaľovať, filtrovať údaje a dynamicky skúmať rôzne perspektívy. Táto interaktivita podporuje zapojenie a hlbšie pochopenie zložitých súborov údajov. [38]

Technológie virtuálnej a rozšírenej reality („Virtual and Augmented Reality“) prinášajú revolúciu do vizualizácie tým, že ponúkajú pohlcujúce zážitky. Vo virtuálnej realite môžu používatelia skúmať trojrozmerné dátové prostredia, čo im umožňuje intuitívnejšie vnímať priestorové vzťahy. Rozšírená realita prekrýva digitálne informácie do reálneho sveta, zvyšuje situačné povedomie a pomáha pri rozhodovaní v oblastiach ako architektúra, medicína a vzdelávanie.

Vizualizácia sa čoraz viac používa ako nástroj na rozprávanie príbehov, kde sú údaje prezentované v naratívnom formáte, aby efektívne sprostredkovali poznatky. Techniky, ako sú animované grafy, infografiky a interaktívne časové osi, pomáhajú zaujať publikum a pútavým spôsobom komunikovať zložité myšlienky.

Algoritmy strojového učenia sa integrujú do vizualizačných nástrojov na automatizáciu analýzy údajov a automatické generovanie zmysluplných vizualizácií. Tieto systémy založené na AI dokážu identifikovať vzory, anomálie a korelácie v rámci veľkých súborov údajov, čo používateľom umožňuje efektívne získavať cenné poznatky.

S exponenciálnym rastom údajov sa vizualizačné nástroje prispôbujú tak, aby efektívne zvládali obrovské množiny údajov. Techniky, ako sú paralelné súradnice, stromové mapy a tepelné mapy, umožňujú používateľom skúmať veľké objemy informácií a získavať použiteľné poznatky bez toho, aby ich zložitosť zahltila.

V komunite vizualizácie údajov bol nedávno zaznamenaný rozpor ohľadom hodnoty a vplyvu nadmerného používania anotácií a dekoratívnych prvkov v grafoch, známe aj ako "grafický odpad". Tradičný prístup, ktorý podporujú odborníci na vizualizáciu ako Edward Tufte a Stephen Few, zdôrazňuje, že grafy by nemali obsahovať zbytočné prvky a mali by prezentovať dáta čo najjasnejšie, bez rušivých faktorov.

Tento názor je podporený aj výskumom v psychologických laboratóriách, ktorý naznačuje, že jasné vizualizácie sú ľahšie spracovateľné. Na druhej strane spektra, niektoré štúdie navrhujú, že pridanie určitých prvkov nad rámec striktnej prezentácie údajov môže zlepšiť zapamätanie informácií divákmi a môže stimulovať vyššiu kognitívnu snahu pre lepšie pochopenie grafu, čo by malo viesť k lepšiemu uchopeniu a porozumeniu prezentovaných údajov.

Výskumy ukazujú, že zbytočné prvky nie sú jediným faktorom, ktorý ovplyvňuje spôsob, akým ľudia vnímajú, interpretujú a pamätajú si vizualizácie. Ďalšie prvky, ako typ grafu, použitie farieb a celková estetika, tiež zásadne ovplyvňujú kognitívne zaťaženie pri spracúvaní vizualizácie a schopnosť informácie si zapamätať. Aby sme odstránili zmätky spôsobené týmito faktormi, sústredíme sa na zodpovedanie základnej otázky: „Čo robí vizualizáciu zapamätateľnou a ľahko pochopiteľnou?“

Predpokladá sa, že vizualizácia, ktorá je ľahko zapamätateľná, nemusí byť nevyhnutne zabudnuteľná. Avšak, pochopenie toho, čo robí vizualizáciu zapamätateľnou, môže byť kľúčom k zodpovedaniu komplexnejších otázok, ako napríklad „Čo robí vizualizáciu príťažlivou?“ alebo „Čo činí vizualizáciu efektívnou?“ Nedávne štúdie naznačujú, že zapamätateľnosť scén z prirodzeného prostredia je medzi ľuďmi konzistentná, čo naznačuje, že niektoré obrázky sú v zásade zapamätateľnejšie než iné, nezávisle na individuálnom kontexte alebo predispozíciách.

Výskum v oblasti kognitívnej psychológie naznačil, že naše pojmové znalosti hrajú kľúčovú úlohu pri organizácii, ukladaní a získavaní informácií z pamäti. Príkladom môže byť, že detaily príbehu alebo obrázka, ktoré sú v súlade s existujúcimi schémami alebo koncepcnými rámcami v pamäti človeka, sú si pamätané s väčšou pravdepodobnosťou než tie nekonzistentné.

Nedávne štúdie o vizuálnej pamäti odhalili, že existujúce kategorické znalosti môžu zvýšiť zapamätateľnosť špecifických detailov. To znamená, že keď si zapamätávame významné objekty z obrázka, nadobúdanie ďalších vizuálnych detailov k týmto objektom je efektívnejšie. Chápanie mechanizmov zapamätateľnosti vizualizácií nám teda môže ponúknuť

lepšie porozumenie a využitie našich kognitívnych schopností pre lepšie uchovávanie informácií.

V poslednej dobe bolo uskutočnených niekoľko štúdií s cieľom posúdiť vplyv dekoratívnych prvkov (ozdôb) na zapamätateľnosť a porozumenie vizualizáciám. Profesor Bateman a jeho kolegovia z „University of New Brunswick“ vykonali štúdiu, v ktorej testovali porozumenie a zapamätateľnosť grafov v ich prikrášených a jednoduchých verziách. Zistili, že prikrášené grafy mali lepšie výsledky v zapamätateľnosti v porovnaní s jednoduchými verziami, a ďalej konštatovali, že prikrášené verzie nezaostávali za jednoduchými verziami v pochopení. Toto zistenie získalo podporu aj z neurobiologického pohľadu, kde sa predpokladalo, že pridanie "vizuálnych ťažkostí" môže divákovi napomôcť k lepšiemu porozumeniu. Na druhej strane, iné štúdie naznačili, že stylistické voľby a vizuálne metafory nemusia mať taký výrazný dopad na vnímanie a porozumenie vizualizácii.

Keď sa vizualizácia stáva výkonnejšou a všadeprítomnou, do popredia sa dostali aj etické úvahy týkajúce sa ochrany osobných údajov, zaujatosti a transparentnosti. Je nutné zabezpečiť, aby boli vizualizácie presné, spravodlivé a dostupné všetkým používateľom bez ohľadu na ich zázemie alebo odbornosť.

Vyvíjajú sa snahy o to, aby boli vizualizačné nástroje prístupnejšie pre používateľov so zdravotným postihnutím, čím sa zabezpečí, že každý môže mať úžitok z prehľadov odvodených z vizualizácií. To zahŕňa funkcie, ako sú čítačky obrazovky, navigácia pomocou klávesnice a palety vhodné pre farboslepých. [2], [38], [40]

1.2.1 Nedávne pokroky vo výskume vizualizácie údajov

S rastúcim využívaním algoritmov umelej inteligencie (AI) a strojového učenia (ML) v rôznych aplikáciách rastie potreba metód na vysvetlenie rozhodnutí, ktoré tieto systémy robia. Výskum v oblasti vizualizácie „Vysvetliteľná umelá inteligencia“ (XAI) sa zameriava na vývoj techník, aby boli modely AI/ML lepšie interpretovateľné a transparentné pre používateľov, čo pomáha pri porozumení a budovaní dôvery. Dôležitá práca zahŕňa pokroky v technikách, ako je vizualizácia dôležitosti funkcií, mapy význačnosti a vysvetlenia rozhodovacieho stromu.

Neistota je sa vyskytuje v mnohých úlohách analýzy údajov, najmä v oblastiach, ako je predpoveď počasia, financie a lekárska diagnostika. Nedávny výskum v oblasti vizualizácie neistoty sa zameriava na vývoj účinných metód na reprezentáciu a komunikáciu neistoty vo vizualizáciách, čo pomáha používateľom robiť informovanejšie rozhodnutia. Techniky,

ako sú pravdepodobnostné grafické modely, kvantifikácia neistoty a vizuálne kódovanie neistoty, boli a sú v súčasnosti rozsiahle študované.

Keďže objem, rýchlosť a rozmanitosť údajov neustále rastie, sú potrebné pokročilé techniky vizuálnej analýzy na extrahovanie zmysluplných poznatkov z veľkých a zložitých súborov údajov. Súčasný výskum v tejto oblasti sa zameriava na vývoj škálovateľných metód vizualizácie, interaktívnych prieskumných nástrojov a algoritmov strojového učenia prispôbených pre analýzu veľkých dát. To zahŕňa výskum vizualizácie vysokorozmerných údajov, vizualizáciu streamovaných údajov a distribuované výpočty pre vizuálnu analýzu.

Imerzívne technológie, ako je virtuálna realita (VR), rozšírená realita (AR) a zmiešaná realita (MR), ponúkajú nové príležitosti na prieskum a vizualizáciu údajov. Nedávny výskum skúmal použitie imerzívnych vizualizačných prostredí pre úlohy, ako je vedecká vizualizácia, architektonický dizajn, lekárske zobrazovanie a tréningové simulácie. Štúdie skúmali účinnosť imerzívnych techník pri zvyšovaní zapojenia používateľov, priestorového chápania a rozhodovania v porovnaní s tradičnými 2D vizualizáciami.

Učenie sa reprezentácie, ktoré zahŕňa učenie sa informatívnej a kompaktnej reprezentácie údajov, je kľúčovou oblasťou výskumu strojového učenia. V kontexte vizualizácie sa nedávne štúdie zamerali na učenie vizuálnych vložení alebo reprezentácií priamo z údajov, aby sa uľahčili úlohy, ako je zhľukovanie, klasifikácia a vizualizácia. Techniky ako t-SNE (t-distribuované stochastické vkladanie susedov) a UMAP (jednotná aproximácia a projekcia) získali popularitu pri vizualizácii vysokorozmerných údajov v priestoroch s nižšími dimenziami pri zachovaní dôležitej štruktúry. [24]

1.3 Vizualizácia v jazyku R

Históriu vizualizácie dát v R možno vysledovať až k ranému vývoju základného grafického systému v R, ktorý poskytol solídny základ pre vytváranie širokej škály statických grafov. Tento systém umožňoval používateľom vytvárať základné grafy, ako sú bodové grafy, stĺpcové grafy, histogramy a čiarové grafy. Aj keď sú tieto základné grafiky funkčné, mali obmedzenia z hľadiska prispôsobenia a estetiky.

Hadley Wickham, prominentná postava v komunite R, ktorej prelomová práca spôsobila revolúciu v spôsobe, akým sa pristupovalo k vizualizácii údajov, predstavil `ggplot2`, výkonný a intuitívny balík na vykresľovanie, ktorý využíva vrstvenú gramatiku grafickej paradigmy. Tento prístup nanovo definoval proces vytvárania vizualizácií poskytnutím štru-

ktúrovaného a koherentného rámca na budovanie zložitých grafov z jednoduchých komponentov.

Prijatie `ggplot2` znamenalo významný obrat v histórii vizualizácie dát v R. Výskumníci, dátoví vedci a analytici mali teraz k dispozícii robustný nástroj na jednoduché vytváranie grafiky v kvalite publikácie. Všestrannosť `ggplot2` umožnila vytvárať rozmanitú škálu grafov, od elegantných čiarových grafov a informatívnych rozptylových grafov až po zložité tepelné mapy a interaktívne vizualizácie. [40]

Ako sa oblasť vizualizácie dát neustále vyvíjala, dopyt po interaktívnych a dynamických grafoch rástol. V reakcii na túto potrebu sa objavili balíčky ako `plotly` a `highcharter`, ktoré používateľom R ponúkajú možnosť vytvárať interaktívne grafy, ktoré možno vložiť do webových aplikácií a prezentácií. Tieto knižnice otvorili nové možnosti interakcie s údajmi a ich interaktívneho skúmania, čo používateľom umožňuje získať hlbší prehľad a efektívnejšie komunikovať zistenia.

Okrem toho vzostup Shiny, rámca webových aplikácií pre R, zmenil prostredie vizualizácie údajov tým, že používateľom umožnil vytvárať interaktívne webové aplikácie s minimálnym úsilím. So Shiny môžu používatelia R vyvíjať dashboards založené na údajoch, interaktívne zostavy a prispôsobené vizualizácie, ktoré možno zdieľať a pristupovať k nim online. Integrácia Shiny s interaktívnymi vizualizačnými knižnicami, ako je `plotly` a `leaflet`, rozšírila rozsah možností na vytváranie interaktívnych dátových zážitkov.

Spolu s pokrokmi vo vizualizačných nástrojoch sa ekosystém `tidyverse` ukázal ako zásadný prevrat v oblasti manipulácie s údajmi a vizualizácie v R. Balíčky ako `dplyr` a `tidyr` zjednodušili proces zápasenia s údajmi a zjednodušili ich čistenie, pretváranie a predspracovanie. dáta pre vizualizáciu. Integráciou týchto nástrojov na manipuláciu s údajmi s výkonnými vizualizačnými knižnicami, ako sú `ggplot2` a `plotly`, môžu používatelia R bezproblémovo prejsť od prípravy údajov k prieskumu a prezentácii, a to všetko v rámci jednotného a koherentného pracovného postupu. [43]

S `gganimate` môžu používatelia R vdýchnuť život svojim vizualizáciám animovaním zmien v priebehu času. Integrácia `gganimate` s vrstvenou gramatikou grafiky `ggplot2` zvyšuje všestrannosť a flexibilitu vytvárania animovaných vizualizácií. Používatelia môžu využiť rozsiahle možnosti `ggplot2` na prispôsobenie estetiky grafu, kombinovanie rôznych typov grafov a začlenenie interaktívnych prvkov, to všetko v rámci `gganimate`. Táto integrácia umožňuje používateľom vytvárať prepracované a vizuálne pôsobivé animácie, ktoré efektívne komunikujú komplexné prehľady údajov.

Pridanie `gganimate` do sady nástrojov na vizualizáciu údajov v R ďalej rozširuje obzory toho, čo je možné z hľadiska rozprávania príbehov a skúmania údajov. Od animovaných čiarových grafov a stĺpcových grafov až po dynamické bodové grafy a geopriestorové vizualizácie, `gganimate` umožňuje používateľom vytvárať pohlcujúce a interaktívne vizualizácie údajov, ktoré upútajú publikum a uľahčia hlbšie pochopenie základných vzorov údajov.

Dnešný stav vizualizácie údajov v jazyku R je charakterizovaný živým a rôznorodým ekosystémom balíkov, nástrojov a rámcov, ktoré uspokojujú široké spektrum potrieb vizualizácie údajov. Od statických grafov pre vysvetľujúcu analýzu až po interaktívne dashboards pre prieskumnú vizualizáciu údajov, R ponúka komplexnú sadu funkcií na podporu vytvárania pôsobivých a prenikavých vizualizácií údajov. [50]

1.4 Interaktívne dátové nástroje a akademický výskum

Táto podkapitola vyzdvihuje podstatu dynamických vizualizačných technológií, odhaľuje možnosti, aplikácie a akademický prínos popredných nástrojov. Snažili sme sa objasniť, ako tieto technológie nielen zvýšili jasnosť a vplyv výskumných zistení, ale tiež podporili interaktívne vzdelávacie prostredie, uľahčili komplexné analýzy a podporili reprodukovateľnosť vedeckých výskumov.

Skúmanie balíka `gganimate` naprieč rôznymi výskumnými publikáciami ukazuje jeho mnohostranné aplikácie, pričom zdôrazňuje nielen vylepšenie vizuálnej reprezentácie údajov, ale aj jeho príspevok k vzdelávacím a interdisciplinárnym výskumným metodológiám. Cesta `gganimate` v akademickom a praktickom prostredí sa začala v roku 2020 tvorivým počínom Mickaela Canouila, „`ggpacman`“, čo znamená prvý zaznamenaný prieskum potenciálu `gganimate` na vzdelávacie a interaktívne účely. Tento prieskum pripravil odraz pre ďalší vývoj a aplikácie v rôznych disciplínach a kontextoch.

Inovatívne využitie `gganimate` na znovu vytvorenie ikonickej hry Pac-Man od Mickaela Canouila prinieslo zmes zábavy a vzdelávania, čo podčiarkuje schopnosť balíka urobiť učenie programovania pútavým prostredníctvom interaktívnej vizualizácie.

Z lingvistického hľadiska výskum Joshua Wilsona Blacka o zmenách vokálov využíva `gganimate` na oživenie lingvistických údajov, pričom zdôrazňuje časové posuny v jazykových zvukoch spôsobom, ktorý statické grafy nedokážu. To zdôrazňuje úlohu `gganimate` vďaka čomu sú abstraktné lingvistické koncepty hmatateľné a vizuálne zrozumiteľné.

Príspevky Nicholasa Sprosona ďalej ilustrujú užitočnosť `gganimate` pri skúmaní viacrozmerných údajov prostredníctvom manuálnych prehliadok a dynamických projekcií. Tým, že umožňuje výskumníkom animovať zmeny orientácie viacrozmerných údajov, `gganimate` ponúka pohľad na komplexné dátové štruktúry, čo uľahčuje intuitívnejšie skúmanie vysokorozmerných súborov údajov.

Súhrnne tieto publikácie nielen podčiarkujú všestrannosť a schopnosť balíka `gganimate` pri transformácii statických vizualizácií na pútavé informatívne animácie, ale tiež zdôrazňujú jeho významný vplyv v rôznych oblastiach. Od vylepšenia vzdelávacích nástrojov, prehĺbenia lingvistického výskumu a pokroku v štatistickej vizualizácii až po prispievanie k environmentálnej genetike a sprístupnenie komplexných údajov sa `gganimate` stáva kľúčovým nástrojom vo vizualizácii a komunikácii dynamických súborov údajov.

V akademickej a technickej sfére sa v roku 2015 prvý krát objavila dokumentácia venujúca sa balíku `plotly`. Distribúcia týchto dokumentov, prevažne mimo tradičných akademických článkov, hovorí o všestrannosti a prístupnosti `plotly`. Softvérová dokumentácia, praktické príručky a zdieľané súbory údajov dopĺňajú akademické články, rozširujú vedomostnú základňu a použiteľnosť grafov v rôznych disciplínach a pre používateľov na rôznych úrovniach zručností.

Medzi pozoruhodné akademické príspevky patrí práca „The design of the structure of the software system for processing text document corpus“ od Barakhnina, ktorá poukazuje na užitočnosť `plotly` v textovej analýze. To ilustruje adaptabilitu balíka nad rámec jeho pôvodného rozsahu, ktorý vyhovuje potrebám návrhu softvérového systému na spracovanie textových údajov.

Ďalším príkladom aplikácie `plotly` je „Interfacing Seurat with the R tidy Universe“ od Stefana Mangiola a kol., ktorá integruje `plotly` do pracovného toku jednobunkovej genómovej analýzy údajov, čím demonštruje svoju schopnosť zlepšiť interaktívne skúmanie údajov vo vysoko špecializovanej doméne genomika.

V oblasti vzdelávania „Helping Introductory Statistics Students Find Their Way Using Maps“ od Daniela Adriana predstavuje zápletku ako vzdelávací nástroj, ktorý pomáha pri výučbe štatistických pojmov prostredníctvom interaktívnych máp. Tento prístup podčiarkuje úlohu `plotly` pri vytváraní komplexných údajov, ktoré sú pre študentov prístupnejšie a pútavejšie.

Od svojho založenia sa `plotly` vyvinul do mnohostranného balíka využívaného v širokej škále oblastí výskumu, od dizajnu softvéru a genomiky až po vzdelávanie, štatistic-

ké analýzy a zdravotníckej vedy. Jeho rozsiahla dokumentácia demonštruje snahu globálnej výskumnej a vývojovej komunity skúmať a rozširovať možnosti `plotly`, čo z neho robí základný kameň pre inovatívnu, interaktívnu vizualizáciu dát.

Po rozšírení užitočnosti R Markdown slúžia publikácie, ako napríklad „R Markdown“ od Benjaminu Baamera a Dany Udwinovej, prvýkrát publikované vo februári 2015, ako základné referencie objasňujúce úlohu R Markdown pri reprodukovateľnom výskume a vytváraní interaktívnych správ. Táto práca zmenila akademické a výskumné postupy a umožnila transparentnejšie a reprodukovateľnejšie pracovné postupy.

Vplyv R Markdown ďalej ilustruje článok Tomáša Dvořáka a kol. „Efektívny empirizmus: Zefektívnenie vyučovania, výskumu a učenia v empirických kurzoch“, publikovaný v júni 2019. Táto práca podčiarkuje kľúčovú úlohu R Markdown pri zvyšovaní efektívnosti vyučovania, výskumných a vzdelávacích oblastí, čím demonštruje svoju hodnotu pri zefektívňovaní empirických kurzov a podpore súdržnejších a interaktívnejších vzdelávacích skúseností.

Doplňkom týchto akademických príspevkov je časopis „Reproducible Research with R and RStudio, 2nd Edition“. Táto publikácia zdôrazňuje kľúčový význam nástrojov ako R Markdown pri presadzovaní princípov reprodukovateľnosti a transparentnosti vo výskumných snahách.

Zbierka dokumentov súvisiacich s R Markdown, vrátane významných článkov a špecializovaných publikácií, podčiarkuje mnohostranné aplikácie balíka a jeho transformačnú úlohu pri zvyšovaní efektívnosti, transparentnosti a interaktivity v rôznych akademických a výskumných disciplínach.

V rámci rozsiahlej oblasti akademickej literatúry a technickej dokumentácie okolo Shiny, výkonného rámca na vytváranie interaktívnych webových aplikácií v R, existuje mnoho zdokumentovaných diel zdôrazňujúcich rôznorodé aplikácie a funkcie Shiny. Tieto publikácie podčiarkujú rozsiahly vplyv a neustály vývoj Shiny od roku 2012, čím sa upevňuje jeho postavenie ako základného nástroja pre interaktívnu vizualizáciu údajov a vývoj webových aplikácií.

Niekoľko príkladných prác ilustruje všestrannosť a užitočnosť Shiny v rôznych doménach. Publikácia Eliasa Laurina Meyera na tému „An interactive R-Shiny app for quickly visualizing a tidy, long dataset with multiple dimensions with an application in clinical trial simulations for platform trials“ zdôrazňuje zjednodušené vizualizačné možnosti Shiny, konkrétne v kontexte klinických skúšobných simulácií. Táto práca ukazuje, ako môže Shiny umožniť analytikom údajov ľahko a efektívne skúmať a prezentovať zložité súbory údajov.

Štúdia Ziren Jiang a kol. o „A comprehensive review and shiny application on the matching-adjusted indirect comparison“ zdôrazňuje, ako môže Shiny uľahčiť podrobné porovnania a umožniť informované rozhodovanie prostredníctvom interaktívnych vizuálnych zobrazení.

Okrem toho výskum Immanuela Jamesa Williamsa a Kelley Kim Williamsovej na tému „Using an R shiny to enhance the learning experience of confidence intervals“ ilustruje úlohu Shiny vo vzdelávacích kontextoch vytvorením interaktívnych nástrojov, ktoré zlepšujú koncepčné porozumenie a zapojenie študentov. To dokazuje prispôsobivosť Shiny pri transformácii tradičných vyučovacích metód na dynamické a interaktívne vzdelávacie skúsenosti.

V oblasti štatistickej analýzy práca Clarke J. M. van Steenderena a kol. „Sample size assessments for thermal physiology studies: An R package and R Shiny application“ zdôrazňuje, ako môže Shiny zjednodušiť zložité výpočty a hodnotenia, vďaka čomu sú štatistické analýzy prístupnejšie a pre používateľov.

Celkovo vzaté, súbor článkov a dokumentov obklopujúcich Shiny vyzdvihuje dynamické schopnosti tohto rámca a ďalekosiahly vplyv v oblasti výskumu, vzdelávania a vizualizácie dát. Prostredníctvom interaktívnych webových aplikácií a užívateľsky prívetivých rozhraní, Shiny umožňuje používateľom vytvárať presvedčivé a informatívne vizualizácie, čo uľahčuje lepšie pochopenie zložitých dátových štruktúr a štatistických konceptov.

2 Cieľ práce

Hlavným cieľom tejto práce je zlepšiť metodiky na vizualizáciu a prezentáciu dát pomocou programovacieho jazyka R so zameraním na vytváranie efektívnejších, intuitívnejších a pútavejších dátových vizuálov pre dátových profesionálov, poistných matematikov a analytikov. To zahŕňa podrobné skúmanie a aplikáciu balíkov R, ako sú `gganimate`, `plotly`, R Markdown a Shiny, ktoré pomáhajú pri pridávaní časovej dynamiky, interaktivity a naratívneho rozmeru do vizualizácií údajov. Analýzou funkcií a potenciálu týchto balíkov má výskum v úmysle odhaliť, ako môžu dynamické a interaktívne vizualizácie premeniť zložité súbory údajov na pútavé príbehy a interaktívne zážitky pre publikum.

Konkrétne sa práca snaží aplikovať funkcionality `gganimate` na animáciu vizualizácií údajov, čím sa do analýzy pridá časový rozmer a umožní publiku sledovať trendy a vzorce v čase. Okrem toho, využitie `plotly` má za cieľ vytvárať interaktívne grafy a tabuľky, ktoré umožňujú používateľom interagovať s údajmi, priblížiť konkrétne detaily a získať komplexnejšie pochopenie základných informácií.

Ďalším kľúčovým aspektom je zlepšiť prezentovanie údajov pomocou R Markdown. Tento nástroj je zameraný na vytváranie vysokokvalitných, reprodukovateľných zostáv, ktoré hladko integrujú kód, text a vizuály, čím uľahčujú zdieľanie údajov. Využitím Shiny na vývoj interaktívnych webových aplikácií a dashboardov chceme docieľiť jednoduchšiu prácu s údajmi v reálnom čase, prispôsobenie vizualizácie na základe preferencií a uľahčenie hlbšie skúmaných údajov.

Okrem toho sa táto práca snaží uľahčiť „rozprávanie príbehov“ (storytelling) pomocou údajov skúmaním princípov vizualizácie údajov a techník rozprávania príbehov. Zameriava sa na to, ako môžu pokročilé koncepty, ako je interaktivita, animácia a dizajn založený na údajoch, zlepšiť kvalitu prezentácií.

V konečnom dôsledku je cieľom vytvoriť vizualizácie, ktoré upútajú a zapoja publikum a podporia hlbšie pochopenie a uchovávanie komplexných údajov. Táto práca sa snaží preklenúť priepasť medzi komplexnými súbormi údajov a ich porozumením širším publikom, čím umožňuje sprostredkovať zložité dátové príbehy „stráviteľnejším“ a interaktívnejším spôsobom a teda prehľady založené na údajoch sa stanú prístupnejšími a použiteľnejšími.

3 Metodika práce

3.1 Data Science

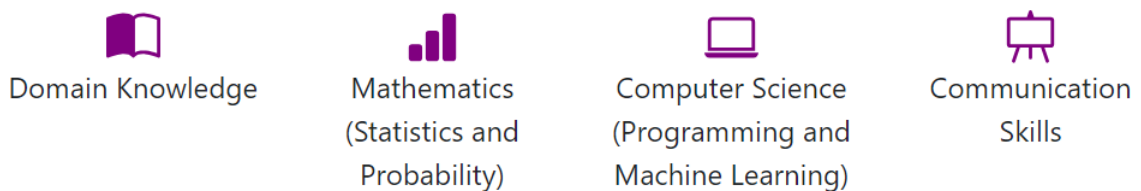
Dátová veda je oblasť, ktorá sa zameriava na analýzu a získavanie cenných poznatkov z údajov na riešenie rôznych problémov.

Dátoví vedci to robia pomocou kombinácie počítačovej vedy, matematiky a znalostí z oblasti podnikania, ktorú analyzujú. Potrebujú tiež dobré komunikačné schopnosti.

Pomocou údajov môžu dátoví vedci:

1. Povedať, čo sa stalo v minulosti (*Descriptive Analytics*).
2. Vysvetliť, prečo sa to stalo (*Diagnostic Analytics*).
3. Predvídať budúce výsledky (*Predictive Analytics*).
4. Odporučiť najlepší postup na základe predpokladaných výsledkov (*Prescriptive Analytics*). [12]

Základné zručnosti, ktoré vedci potrebujú:



Obrázok 1: Základné zručnosti Data Science, zdroj: [12]

Rozumieť doméne znamená mať hlboké znalosti o konkrétnom predmete, oblasti alebo odvetví, na ktorom pracuje dátový vedec. Tieto znalosti sa získavajú prostredníctvom vzdelávania, skúseností, zvedavosti a schopnosti klásť relevantné otázky. Znalosť domény je kľúčová pre dátových vedcov, aby dokázali porozumieť problémom, ktoré sa snažia riešiť. Je nevyhnutná pre identifikáciu správnych otázok a predchádzanie odpovedi na nesprávne otázky. Okrem toho pomáha dátovým vedcom lepšie chápať výsledky ich analýz. [21]

Štatistika predstavuje matematickú disciplínu s dôrazom na efektívne metódy zhromažďovania, analýzy a prezentácie dát. Je neodmysliteľnou súčasťou vedeckého procesu v oblasti práce s údajmi.

Pravdepodobnosť nám poskytuje informáciu o tom, aká je šanca, že určitá udalosť nastane. To je dôležité pri vytváraní predpovedí. Vyššia pravdepodobnosť by ľuďom poskytla väčšiu dôveru vo výsledky.

Programovanie je akt zadávania pokynov či inštrukcií počítaču s cieľom vykonania určitej úlohy. Obrovské množstvo údajov je dostupné prostredníctvom internetu, satelitných snímok, prieskumov, zákazníckych údajov, ktoré spoločnosti zhromažďujú a podobne. Na analýzu týchto údajov je vo veľkom rozsahu potrebná informatika, najmä programovanie.

Machine learning modely sú počítačové programy, ktoré umožňujú využitie údajov na "trénovanie" počítačov s cieľom identifikovať vzory a znalosti. Po úspešnom tréovaní môžeme tieto modely využiť na predpovede. Táto významná inovácia vznikla v dôsledku kombinácie počítačovej vedy, štatistiky a pravdepodobnosti. Dátoví vedci využívajú svoje odborné znalosti na prípravu a poskytovanie kvalitných údajov pre modely machine learning, obdobne ako učitelia, ktorí svojim študentom poskytujú najlepšie informácie s cieľom im pomôcť dosiahnuť úspech.

Schopnosť komunikácie je kľúčová, pretože dátoví vedci sú často v pozícii, kde musia zdieľať svoje poznatky s ľuďmi, či už s technickým alebo netechnickým pozadím. V prípade, že odborníci v oblasti údajov nie sú schopní efektívne komunikovať svoje zistenia, môže sa stať, že tieto poznatky nebudú využiteľné pri rozhodovacom procese.

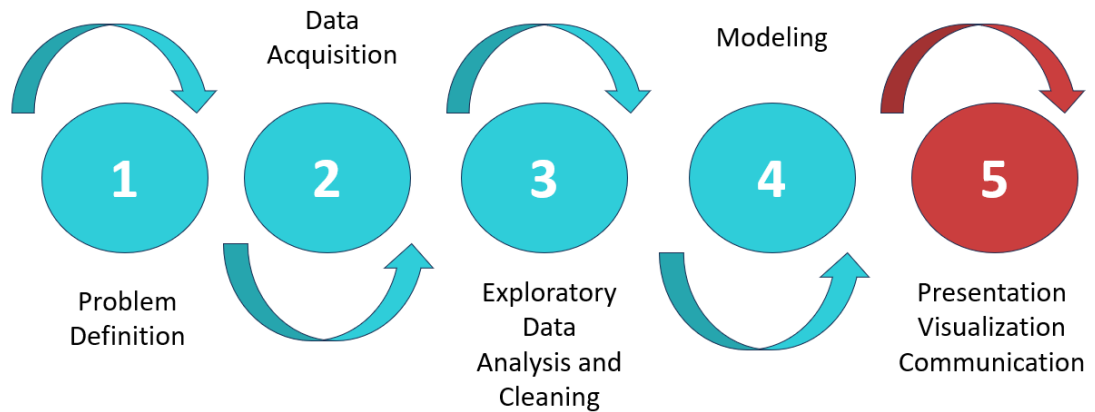
Dátoví vedci extrahujú hodnotné informácie z dát s cieľom riešiť rozličné problémy. Tento proces zahŕňa kombináciu počítačovej vedy, štatistiky a expertíznej znalosti v danej oblasti analýzy. [10], [11], [12]

3.2 Data Science proces

Štandardné kroky v procese data science definujú postup, ktorým sa dátoví vedci riadia pri riešení problémov. Tento postup je tiež označovaný ako životný cyklus data science, pretože každý krok poskytuje informácie a usmerňuje nasledujúci. Proces data science obvykle prebieha iteratívne, čo znamená, že vedci údajov môžu medzi niektorými krokmi podľa potreby prechádzať tam a späť.

1. Definícia problému

Toto je kľúčový prvý krok, plánovacia fáza procesu. Dátoví vedci spolupracujú s klientmi na pochopení obchodného problému, identifikácii relevantných zdrojov údajov a stanovení cieľov projektu. V tejto etape je dôležitá expertíza v danej oblasti, ale ak je dátový vedec nový v tomto odvetví, tiež sú potrebné silné komunikačné schopnosti a zvedavosť.



Obrázok 2: Data Science životný cyklus,
zdroj: vlastné spracovanie

2. Získavanie údajov

Ďalším krokom je získanie potrebných údajov na riešenie daného problému. Extrahovanie údajov môže byť jednoduché alebo zložité v závislosti od ich zdroja. V prípade, že požadované údaje pochádzajú z viacerých zdrojov, môže byť potrebné vykonať agregáciu alebo spojenie rôznych súborov údajov. Príklady zdrojov údajov zahŕňajú MS Excel, textové dokumenty, databázy (SQL alebo NoSQL), webové stránky, obrázky a podobne.

3. Prieskumná analýza údajov a čistenie údajov

Vo fáze prieskumnej analýzy dátoví vedci získajú pochopenie údajov pomocou štatistík (sumarizácie údajov) a vizualizácií. Je dôležité mať zvedavý prístup a klásť relevantné otázky, aby sa údaje dostatočne preskúmali. V prípade, že sa vyskytne niečo zaujímavé alebo znepokojujúce, je dôležité položiť si otázku, prečo alebo ako k tomu došlo, a identifikovať možnosti ďalšieho hlbšieho preskúmania.

V priebehu prieskumnej analýzy sa identifikujú problémy týkajúce sa kvality údajov, a vytvára sa plán na ich riešenie. V etape čistenia údajov sa následne realizuje tento plán s cieľom vylepšiť kvalitu údajov. Úspech modelu machine learning výrazne závisí od kvality údajov. Používanie nečistých údajov môže viesť k nespoľahlivým informáciám. Keďže väčšina údajov v reálnom svete je nečistých, čistenie údajov je kľúčové pre dosiahnutie najlepších výsledkov. [4], [12]

4. Modelovanie

V tejto časti dátoví vedci trénujú modely strojového učenia a vyhodnocujú ich výkon. Modelovanie zahŕňa:

- **Inžinierstvo funkcií:** Vytvoriť nové premenné z existujúcich premenných, ktoré by mohli vylepšiť model.
- **Rozdelenie údajov:** Rozdeliť údaje na tréningové a testovacie súbory údajov.
- **Tréning modelu:** Použiť vhodné modely na tréningové dáta a vyladiť modely, aby sa dosiahli lepšie výsledky.
- **Hodnotenie modelu:** Posúdiť výkonnosť modelu na testovacej sade a interpretovať výsledky modelu.

5. Prezentácia, vizualizácia a komunikácia

Pre dosiahnutie úspechu musia dátoví vedci prezentovať výsledky tak, aby boli zrozumiteľné ich zákazníkom, kolegom a nadriadeným. Je nevyhnutné, aby dokázali prezentovať hodnotu svojej analýzy. Keď dátový vedec píše pokročilé algoritmy prediktívnej analýzy alebo strojového učenia, je dôležité vizualizovať výstupy na monitorovanie výsledkov a zabezpečiť, aby modely fungovali podľa plánu. Je to preto, že vizualizácie zložitých algoritmov sa vo všeobecnosti interpretujú ľahšie ako numerické výstupy. [4]

3.3 Dátová analytika

Väčšina spoločností neustále zbiera obrovské množstvo údajov, no v ich surovej podobe tieto údaje nemajú skutočný význam. Pre ich nadobudnutie je nevyhnutná analýza údajov. Analýza údajov je proces analýzy nespracovaných údajov s cieľom získať zmysluplné a použiteľné poznatky, ktoré následne slúžia na informovanie a riadenie inteligentných obchodných rozhodnutí.

Dátový analytik získava surové údaje, usporadúva ich a následne ich analyzuje s cieľom premeniť ich z nezrozumiteľných čísel na súvislé a zrozumiteľné informácie. Po interpretácii údajov dátový analytik prezentuje svoje zistenia formou návrhov alebo odporúčaní, ktoré spoločnosť môže nasledovať v ďalších krokoch. [3]

3.3.1 Nástroje na analýzu údajov

V tejto podkapitole sa budeme zaoberať 6 nástrojmi na analýzu údajov, zahŕňajúc niekoľko programovacích jazykov, ktoré môžu poskytnúť vylepšené možnosti pre vykonávanie analýzy.

1. R: R je programovací jazyk s otvoreným zdrojovým kódom, ktorý sa používa hlavne na numerickú a štatistickú analýzu. Poskytuje celý rad knižníc na analýzu a vizualizáciu údajov.

2. Python: Python je objektovo orientovaný open-source programovací jazyk. Podporuje celý rad knižníc na manipuláciu s údajmi, vizualizáciu údajov a modelovanie údajov.

3. Tableau: Ide o zjednodušený nástroj na vizualizáciu a analýzu údajov. Umožňuje vytváranie rôznych interaktívnych vizualizácií, tvorbu zostáv a informačných panelov na prezentáciu prehľadov a trendov.

4. Power BI: Power BI predstavuje nástroj podnikovej inteligencie, ktorý disponuje jednoduchou funkciou presúvania myšou. Tento nástroj má schopnosť integrovať viaceré zdroje údajov a ponúka vizuálne atraktívne možnosti pre zobrazenie údajov. Power BI disponuje funkciami, ktoré umožňujú pokladať otázky o údajoch a získavať okamžité štatistiky.

5. QlikView: QlikView poskytuje interaktívnu analýzu s využitím technológie in-memory storage na spracovanie obrovských objemov údajov a využitie objavov údajov pri podpore rozhodovania. Tento nástroj umožňuje preskúmanie sociálnych údajov a interaktívne riadené analýzy. Schopnosť rýchlo a presne manipulovať s masívnymi súbormi údajov je jeho silnou stránkou.

6. Apache Spark: Apache Spark predstavuje nástroj s otvoreným zdrojom na analýzu údajov, ktorý realizuje spracovanie údajov v reálnom čase a vykonáva pokročilé analýzy prostredníctvom SQL dotazov a algoritmov strojového učenia. [22], [28]

3.4 Aktuárstvo

V tejto práci budeme pracovať s aktuárskymi dátami a zároveň prezentovať výstupy pomocou vizualizácie. V tejto časti si preto bližšie vysvetlíme, čo znamená pojem aktuárstvo a analyzujeme dôvody, prečo je vizualizácia v oblasti aktuárstva nevyhnutná.

Aktuárstvo je disciplína, ktorá sa zaoberá hodnotením rizík v oblasti poisťovníctva a financií pomocou rôznych matematických a štatistických metód. Profesionálmi, ktorí vykonávajú tieto úlohy zisťovania, analýzy a poskytovania riešení budúcich neistôt s finančnými rizikami, sú aktuári.

Aktuári pracujú v rôznych odvetviach vrátane poisťovní, penzijných fondov, vládnych agentúr a finančných inštitúcií. Zohrávajú kľúčovú úlohu pri navrhovaní poisťných zmlúv, určovaní sadzieb poisťného a odhadovaní budúcich finančných záväzkov pre dôchodkové plány. Poistní matematici tiež prispievajú k rozvoju investičných stratégií a hodnotia finančný dopad rôznych scenárov na stabilitu a solventnosť organizácie. [20]

V kontexte so sylabami aktuárskeho vzdelávania SSA, resp. AAE, konkrétne časť 9.1 Efektívna komunikácia, sme identifikovali, že vizualizácia hrá kľúčovú úlohu pri prezen-

tovaní výsledkov aktuárskej práce publiku ako sú napríklad spolupracovníci, manažéri alebo klienti.

Vizualizácia histórie údajov a trendov umožňuje aktuárom identifikovať vzory a trendy efektívnejšie ako prostredníctvom samotných číselných údajov. To znamená, že vizualizácia môže byť nástrojom na analýzu scenárov a testovanie citlivosti, čo aktuárom umožňuje preskúmať vplyv rôznych predpokladov, zmien parametrov alebo rôznych scenárov na finančné výsledky.

Vizualizácia v aktuárstve teda nie je len prostriedkom na prezentovanie výsledkov, ale aj nástrojom na analyzovanie a porozumenie údajom, čo z nej robí kľúčový prvok v procese rozhodovania. Toto všetko spolu s požiadavkami na efektívnu komunikáciu poskytuje rámec pre použitie techník vizualizácie dát v našej práci.

3.5 Vizualizácia dát

Vizualizácia predstavuje proces skúmania, transformovania a prezerania údajov v podobe obrázkov s cieľom dosiahnuť lepšie pochopenie a prehľad o týchto údajoch. Štúdium ľudského vnímania, počítačová grafika, zobrazovanie, numerická analýza, štatistické metódy a analýza údajov prispeli k významnému rozvoju vizualizácie. Obrázky majú jedinečnú schopnosť sprostredkovať informácie a myšlienky. Neformálne je vizualizácia transformáciou údajov alebo informácií na obrázky. To zapája primárny ľudský zmyslový aparát, zrak, ako aj výpočtovú silu ľudskej mysle. Výsledkom je efektívne médium na komunikáciu komplexných a rozsiahlych informácií.

Keďže množstvo údajov prevyšuje schopnosť človeka ich asimilovať a pochopiť, nemôžeme vizualizáciu obísť. Poistní matematici teda musia vypracovať konvencie na reprezentáciu ich údajov a výsledkov. [34]

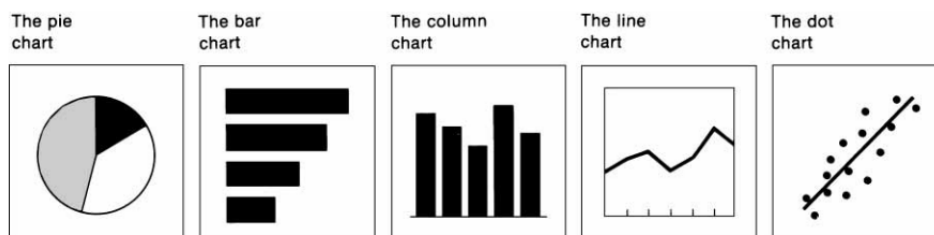
Existuje množstvo vizualizačných prístupov: mapovanie skalárov na farby, kontúrovanie (vykresľovanie isopovrchov), glyfy (šípky rôznej farby, dĺžky, smeru), deformácia (zobrazenie rôznych fáz v pohybe), grafy posunu, časové animácie, prúdnice (stopy častíc) a tenzor algoritmy.

Pri viacrozmernej vizualizácii sa vyskytujú nevyhnutné problémy, ktorými sú projekcia (kde sú k dispozícii len dve alebo tri dimenzie) a porozumenie (pretože nám ako ľuďom nie je ľahké pochopiť viac ako trojrozmerné alebo trojrozmerné plus časové animácie). Je však možné realizovať projekcie ak sú tri najdôležitejšie rozmery identifikované tak, že

zostávajúce rozmery možno ignorovať. Výber týchto rozmerov je opäť úlohou poistnej matematiky. [42]

3.5.1 Grafy

Bez ohľadu na to, koľko business grafov vidíme v rôznych druhoch komunikácie – vrátane tabuliek, grafov organizácie, vývojových diagramov, matíc, a máp, pokiaľ ide o kvantitatívne grafy, existuje iba päť základných foriem grafov na výber. Zjednodušene povedané, tieto to sú: koláčový graf, bar graf, stĺpcový graf, čiarový graf, bodový graf.



Obrázok 3: Vizualizácia základných grafov, zdroj [48]

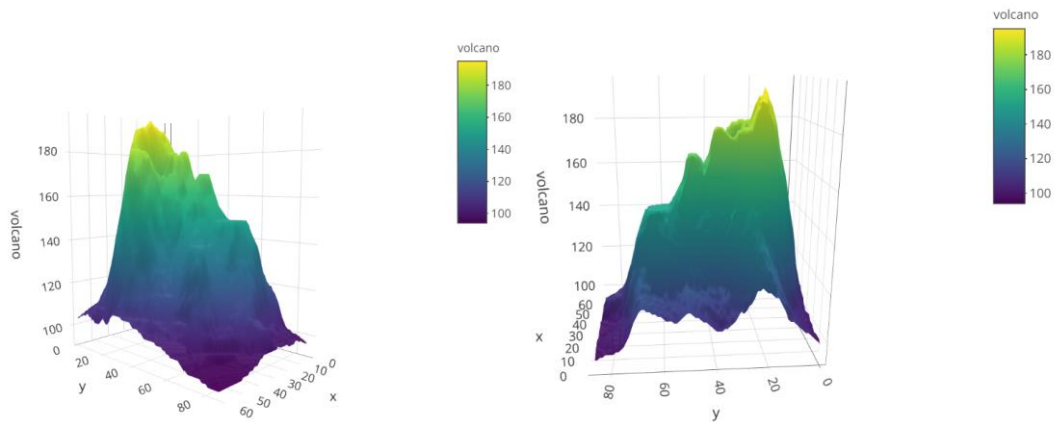
Spôsob vizualizácie grafov:

Statické vizualizácie môžu ponúkať iba vopred zostavené „pohľady“ údajov, takže na prezentovanie rôznych pohľadov na tie isté informácie sú často potrebné viaceré statické zobrazenia. Taktiež existuje obmedzenie v počte dimenzií dát, kedy všetky vizuálne prvky musia byť zobrazené na rovnakom povrchu v rovnakom čase. Ilustrovanie viacrozmerných dátových súborov v statických obrázkoch je známe svojou náročnosťou.

Dynamické, interaktívne vizualizácie môžu ľuďom umožniť sprístupniť údaje rôznym publikám, od tých ktorí iba prehliadajú alebo skúmajú súbor údajov, až po tých, ktorí pristupujú k vizualizácii s konkrétnou otázkou pri hľadaní odpovede. Interaktívna vizualizácia, ktorá poskytuje prehľad údajov spolu s nástrojmi na detailnejšie skúmanie, môže úspešne plniť mnoho úloh naraz a riešiť rôzne záujmy rôznych cieľových skupín. [28]

Obrázok č. 4 zobrazuje 3D povrchový graf s použitím súboru údajov `volcano`, čo je vstavaný súbor údajov v R. Tento súbor údajov predstavuje topografickú výšku časti sopky Maunga Whau v Aucklande na Novom Zélande. Každý prvok v množine údajov zodpovedá nadmorskej výške na konkrétnom mieste na mriežke. Poistní matematici používajú povrchové grafy na vizualizáciu geografických prvkov súvisiacich s rizikami, ako sú oblasti náchylné na záplavy, čo pomáha pri hodnotení rizík a stanovovaní cien. Pohyb alebo interak-

cia s grafom umožňuje aktuárom dynamicky skúmať údaje z rôznych perspektív. Toto dynamické skúmanie môže odhaliť skryté vzorce alebo vzťahy, ktoré nemusia byť zrejmé zo samotných statických pohľadov.



Obrázok 4: Vizualizácia 3D grafu z 2 rôznych uhlov, zdroj [8]

3.5.2 Čo je animácia?

Ak je obrázok hodný tisíc slov, tak animácia je hodná tisícich obrázkov. Animácia predstavuje optimálny spôsob využitia aspektu ľudskej psychiky známeho ako „selektívna pozornosť“ – ľudia pohotovo reagujú (presúvaním ťažiska svojej pozornosti) na akýkoľvek pohyb, vrátane zmeny vo farbe, veľkosti alebo polohe. Animácia je vhodná na vizualizáciu rozsahu neistoty a vývoja, čo sú dva z kľúčových poistno-matematických fenoménov.

Ak vezmeme príklad z odvetvia počítačových hier, animáciu možno efektívne využiť pri vizualizácii procesu simulácie. Skutočne, s rastúcim významom DFA (Deterministic Finite Automata) a iných neanalytických modelovacích techník, sa simulácie postupne stávajú technikou pre väčšinu poistných matematikov. V súčasnosti sú však všetky medzikroky používané pri simulácii pred používateľom skryté – takéto množstvo informácií sa v podstate zahodí. Použitie animácie môže zabrániť tomuto „plytvaniu“ medzi výpočtami. Animované zobrazenie krokov simulácie môže používateľovi pomôcť pri vizualizácii dynamiky simulovaného procesu respektíve oceniť rozsah neistoty v simulovaných scenároch. [23]

3.5.3 Cieľ a účel vizualizácie

Účelom vizualizácie údajov je objaviť vzory, vzťahy, hierarchie a rozdiely, ktoré by bolo ťažké alebo nemožné odhaliť len na základe štatistických postupov alebo preskúmaním textových alebo tabuľkových foriem prezentácie údajov. Dôležité je si uvedomiť, že vizuali-

zácia údajov obvykle zohráva v analytickom procese indukívnu rolu, identifikuje pozorovania a zistenia, ktoré samy o sebe majú osobitnú hodnotu, ale zároveň môžu slúžiť aj ako východiská pre ďalšie hypotézy a skúmanie pomocou analytických metód. To znamená, že použitie vizualizácie údajov zamerané na prieskum môže byť samo o sebe výsledkom alebo predchodcom ďalších analýz.

Vizualizácia vysvetľuje vzorce, trendy alebo vzťahy zahŕňajúce premenné, ktoré nás zaujímajú. Vizualizácie, ktoré nepochádzajú z nespracovaného súboru údajov, ale skôr z výskumnej otázky alebo obchodného cieľa, môžu graficky zobrazit' alternatívne hypotézy, čo umožňuje používateľovi odhadnúť, ktorá je najpravdepodobnejšia. Schopnosť vysvetľovať sa týka aj komunikácie samotných zistení s cieľom osloviť širšie publikum, efektívne usmerniť používateľa na oblasť témy a podporiť interpretáciu, závery a rozhodnutia v takom rozsahu, ktorý by nebol dosiahnuteľný prostredníctvom textových alebo číselných komunikačných formátov. [38]

Zhrnutie základných cieľov a účelov vizualizácie:

- schopnosť rýchlo absorbovať informácie, zlepšovať prehľad a robiť rýchlejšie rozhodnutia;
- lepšie pochopenie ďalších krokov, ktoré je potrebné podniknúť na zlepšenie organizácie;
- zlepšená schopnosť udržať záujem publika o informácie, ktorým rozumie;
- jednoduchá distribúcia informácií, ktorá zvyšuje možnosť zdieľať poznatky so všetkými zúčastnenými;
- zvýšená schopnosť rýchlo konať na základe zistení, a teda dosiahnuť úspech s vyššou rýchlosťou a menším počtom chýb. [4]

3.6 Prístupy akými možno prekonať chaos v údajoch

Súčasná technologická prostriedky a stratégie sa stávajú základnými nástrojmi v boji proti informačnému neporiadku. V tejto časti preskúmame metódy a techniky, ktoré nám umožňujú lepšie porozumieť a usporiadať tento zdanlivo nekonečný tok údajov.

3.6.1 Storytelling a jeho základné techniky

Aktuárstvo je špecifické vytváraním algoritmov, prácou s dátami, programovacími jazykmi a podobne. Pre aktuára je však dôležité aby dokázal presvedčiť svoje publikum,

vysvetliť aké výsledky dosiahol a to nie len svojim kolegom a nadriadenému ale aj nezainteresovaným osobám. Takto sa dostávame k „storytellingu“ ako jednému z hlavných trendov novodobého prezentovania.

„Storytelling“ je proces umenia a komunikácie, kde sa príbehy využívajú s úmyslom zaujať, informovať alebo presvedčiť publikum. Tento spôsob komunikácie využíva rôzne naratívne prvky, vrátane postáv, deja, miesta a konfliktu, na prenesenie správ, hodnôt alebo skúseností. Hlavným zámerom „storytellingu“ je vytvoriť emocionálny a významný zážitok, ktorý oslovi publikum a umožní im lepšie pochopiť predstavovanú tému.

Je prirodzené, že poistní matematici musia disponovať matematickými a štatistickými schopnosťami. Použitie rozprávania môže slúžiť ako prostriedok na pútavé sprostredkovanie informácií a urobenie ich ľahko zapamätateľnými, naratívny prístup zároveň dokáže emocionálne zaujať publikum a motivovať ho k akcii. [47]

Poistní matematici sú ponorení do racionálneho myslenia, takže sú zvyknutí tráviť potrebný čas hľadaním a analýzou údajov. Môže byť ľahké zabudnúť na to, aké to je byť klientom – niekým, komu chýbajú nástroje na tento druh zámerného myslenia. Preto je pre poistných matematikov dôležité rozvíjať empatiu voči ich publiku predtým, než prezentujú informácie založené na údajoch.

Potrebné je zvážiť tieto otázky:

1. Aké je zloženie vášho publika? Čo potrebujú vedieť?
2. Aké tri veci chcete, aby vaše publikum po vypočutí vašej prezentácie pochopilo? Pomenujte ich pomocou maximálne 30 slov.
3. Akým výzvam okrem projektu, na ktorom sa prezentujete, čelí vaše publikum?

Vizualizácia údajov zohráva dôležitú úlohu pri „storytellingu“. Sila vizualizácie údajov – tiež vedecky dokázaná – spočíva v tom, že vizuály stimulujú náš mozog rôznymi spôsobmi. Podľa vedy o rozprávaní príbehov si ľudia počas prvého kroku predstavia samých seba v príbehu, ktorý rozprávame. V druhom kroku začnú rozvíjať svoje vlastné nápady a v treťom kroku dochádza k uvoľneniu dopamínu v mozgu, ktorý im pomáha zapamätať si príbeh a konať. Počas štvrtého a piateho kroku si mozog vytvára spomienky. Dôležitá je teda vizuálna zložka príbehu. [2], [25]

Efektívny poistný matematik by mal byť schopný dokázať aj presvedčiť. Aby sa dosiahla táto rovnováha, netreba pri prezentovaní údajov zabudnúť na dve pravidlá:

1. Slová nad číslami

Nie je to tak, že slová sú dôležitejšie ako čísla, ale chcete ich použiť viac ako čísla. Kapacita našej pamäte je obmedzená, najmä pokiaľ ide o čísla. Pokiaľ ide o čísla, ktoré zdieľame treba byť selektívny a zdieľať ich v kontexte slov. Pre mnohé publikum je príbeh vyrozprávaný slovami zmyslupnejší ako samotné čísla.

2. Vytvorte zmysel

Je nutné si pamätať, že čísla nie sú podstatné – jednoducho nám pomáhajú dosiahnuť náš cieľ prezentovať naše zistenia. Aktuárstvo je veda a presvedčanie je umenie – ale byť presvedčivý svojou aktuárskou prácou je remeslo. Informácie boli spracované podľa zdrojov [7], [32], [47].

Sedem techník „storytellingu“:

1. Monomyth

Monomýtus, známy aj ako cesta hrdinu, predstavuje štruktúru príbehu, ktorú nájdeme v mnohých ľudových rozprávkach, mýtoch a náboženských textoch z rôznych kútov sveta. Hrdina v monomýte je vyzvaný, aby opustil svoj domov a podstúpil náročnú cestu do neznámeho. Po prekonaní veľkých skúšok sa vracia domov s odmenou alebo novou múdrosťou, ktorá môže pomôcť jeho komunite. Použitie monomýtu pri formovaní prezentácie môže pomôcť vysvetliť, čo nás priviedlo k múdrosti, ktorú sa snažíme zdieľať.

Účel:

- zobrat' divákov na cestu;
- ukázať výhody riskovania;
- ukážka toho, ako ste sa naučili nejakú novú múdrosť.

2. The mountain

Štruktúra tejto techniky je spôsob mapovania napätia a drámy v príbehu. Je to podobné ako monomýtus, pretože nám pomáha vytvárať zápletky, keď sa isté udalosti odohrali v príbehu. Nemusí mať nevyhnutne šťastný koniec. Prvá časť príbehu je daná nastaveniu scény a po nej nasleduje len séria malých výziev a stúpajúca akcia pred vyvrcholením závere. Je to trochu ako televízny seriál – každá epizóda má svoje výhody a pády, všetko graduje do veľkého finále na konci obdobia.

Účel:

- ukázať, ako ste prekonali sériu výziev;
- pomaly budovať napätie;

- dosiahnutie uspokojivého záveru.

3. Nested loops

Vnorené slučky sú technikou rozprávania, kde navrstvíme tri alebo viac príbehov do seba. Najdôležitejší príbeh sa umiestňuje do jadra našej správy a príbehy okolo neho sa rozpracujú alebo vysvetlia ústredný princíp.

Prvý príbeh, ktorý začnete, je posledný, ktorý dokončíte, druhý príbeh, ktorý začnete, je predposledný atď. Predstavte si, že vám priateľ hovorí o múdrom človeku vo svojom živote. Nieko, kto ho naučil dôležitú lekciu. Prvé slučky sú príbehom vášho priateľa, druhé slučky sú príbehom múdreho človeka. V strede je dôležitá lekcia.

Účel:

- vysvetlenie procesu, ako ste sa inšpirovali a dospeli k záveru;
- použitie analógií na vysvetlenie ústredného pojmu;
- ukazuje, ako sa odovzdával kus múdrosti vám.

4. Sparklines

Tie najlepšie prejavy sú úspešné pretože kontrastujú náš bežný svet a ideálny, vylepšený svet. Porovnávajú, čo je a čo by mohlo byť. Týmto moderátor upriamuje pozornosť na problémy, ktoré máme v našej spoločnosti, v našom osobnom živote, naše podniky. Moderátor vytvára túžbu po zmene v publiku. Je to vysoko emocionálna technika, ktorá motivuje publikum, aby vás podporilo.

Účel:

- inšpirovať publikum k akcii;
- vytvoriť nádej a vzrušenie;
- vytváranie sledovania.

5. In medias res

In medias res rozprávanie je vtedy, keď začínate rozprávanie v zápale akcie, predtým ako začnete odznova, na začiatku vysvetlite, ako ste sa tam dostali. Tým, že publikum vtiahnete priamo do najvzrušujúcejšej časti vášho príbehu, bude uchvátené od začiatku a zostane zaujaté, aby zistilo, čo sa stane. Tu však treba byť opatrní, nechcete rozdávať veľkú časť akcie okamžite. Dajte svojmu publiku dostatok informácií, aby ste ich udržali v kontakte, keď sa vrátite späť a nastavte scénu svojho príbehu. Toto však funguje len pri kratších prezentáciách, ak sa to natiahne príliš dlho, publikum sa stane frustrované a stráca záujem.

Účel:

- upútanie pozornosti od začiatku;
- udržať túžbu publika;
- zameranie pozornosti na kľúčový moment vo vašom príbehu.

6. Converging ideas

Konvergujúce myšlienky predstavujú rečovú štruktúru, ktorá ilustruje, ako sa rôzne myšlienky zlúčili a spoločne vytvorili jeden výsledok alebo koncept. Táto technika môže byť využitá na vysvetlenie vzniku hnutia alebo na ukázanie, ako jeden nápad vzišiel z viacerých významných myšlienok smerujúcich k jednému cieľu. Konvergujúce nápady sú podobné štruktúrou ako vnorené slučky, ale skôr než rámovaním jedného príbehu komplementárnymi príbehmi môže ukázať, ako rovnako dôležité príbehy dospeli k jedinému silnému záveru. Táto technika by mohla byť použitá na rozprávanie príbehov o niektorých z najväčších svetových partnerstiev – napr. webových vývojári Larry Page a Sergey Brin.

Účel:

- ukázať, ako sa spojili výnimočné mysle;
- ukázať, ako došlo k vývoju v istom bode v histórii;
- ukázať, ako sa vytvorili symbiotické vzťahy.

7. False start

Príbeh s falošným začiatkom je, keď začnete rozprávať zdanlivo predvídateľný príbeh pred nečakaným narušením a začnete odznova. Prilákate svoje publikum do falošného pocitu bezpečia a potom ich šokujete otočením stolov.

Tento formát je ideálny na zdieľanie príbehu o období, keď ste čelili neúspechu, následne sa k tomu vrátili a získali nový pohľad. Je vhodný na diskusiu o lekciách, ktoré ste získali z tejto skúsenosti, alebo o inovatívnych spôsoboch, ako ste vyriešili problém. Jeho výhodou je, že rýchlo pritiahne pozornosť a prekvapí publikum. Informácie boli spracované podľa zdrojov [21], [25], [32].

Účel:

- narušenie očakávaní publika;
- ukázanie výhod flexibilného prístupu;
- udržiavanie zapojenia publika.

3.6.2 Informačná grafika

Informačná grafika sa používa na komunikáciu zložitých vecí vo vizuálne jednoduchom médiu, ktorému môže porozumieť ktokoľvek. Tieto zložité veci, zahŕňajú finančné štatistiky, časové harmonogramy a dokonca aj technické procesy. Informačnú grafiku možno použiť na zobrazenie vzťahov medzi dvomi alebo viacerými prvkami, ako je napríklad vzťah príčiny a následku.

Niekedy môžu určité grafy vyzerat' monotónne a vizuálne nezaujímavo, až natoľko, že oslabujú komunikáciu informácií, ktoré sa snažíme preniesť. Naopak, moderná infografika predstavuje umelecké dielo. Pre efektívnu prezentáciu je potrebná kreativita v oblasti jazykového umenia, schopnosť vizualizovať prostredníctvom grafického dizajnu a využívanie technológie. [6], [17]

Začnime s tromi základnými prvkami informačnej grafiky:



Obrázok 5: Vizualizácia základných prvkov informačnej grafiky, zdroj: [44]

Bohatý obsah dáva grafike zmysel. „Pozývajúca“ vizualizácia interpretuje obsah a zdôrazňuje podstatu informácií pre čitateľa. Sofistikované prevedenie oživuje obsah a grafiku.

Pri vytváraní grafov je najlepšie systematicky dodržiavať štyri základné kroky: prieskum, úprava, graf a kontrola.

Prieskum:

- prieskum z aktuálnych dôveryhodných zdrojov;
- použiť nezávislý zdroj pre sporné údaje, ktorý je otvorený interpretácii, ako napríklad podiel na trhu, aby sme sa vyhli zaujatosti;
- v prípade potreby získať povolenie na používanie údajov.

Úprava údajov:

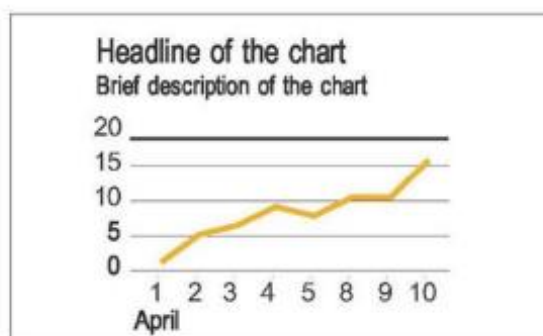
- identifikácia svojho kľúčového posolstva;
- výber najlepšieho rádu údajov, ilustrácia svojho názoru, napríklad podiel na trhu vs. celkový príjem;
- filtrácia a zjednodušenie údajov, aby sme poskytli podstatu údajov zamýšlanému publiku;
- vykonanie číselnej úpravy nespracovaných údajov, aby sme zlepšili náš názor, napríklad absolútne hodnoty vs. percentuálna zmena.

Absolute values		Percentage change	
A	B	A	B
10	100		
20	110	+100%	+10%
30	120	+200%	+20%

Obrázok 6: Vizualizácia úpravy údajov - percentuálna zmena absolútnych hodnôt premennej A a B, zdroj: [44]

Tvorba grafu:

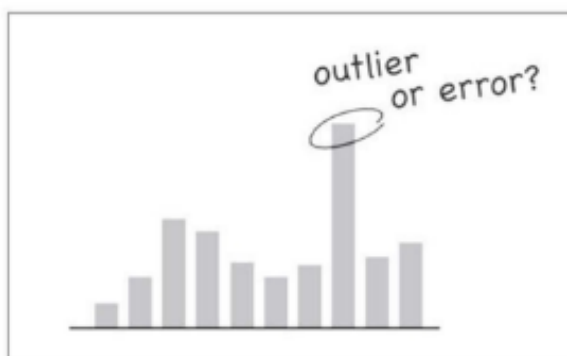
- výber správneho grafu na prezentáciu údajov, napríklad čiara na zobrazenie trendu alebo čiara na zobrazenie diskretných veličín;
- voľba vhodného nastavenia grafu, napríklad mierka, prírastky osi y a základná čiara;
- označenie grafu, napríklad titul, popis, legendy a zdrojový riadok;
- použitie farby a typografie, čím získame kľúčové posolstvo.



Obrázok 7: Ukážka atribútov grafu, zdroj: [44]

Kontrola grafu:

- skontrolovať vykreslené údaje so svojimi zdrojmi;
- použiť úsudok na posúdenie, či náš graf dáva zmysel;
- skúsiť sa na graf pozrieť z pohľadu čitateľa;
- overenie svojich údajov pomocou ďalších zdrojov a konzultácia s odborníkmi z našej oblasti v prípade sporného obsahu a odľahlých hodnôt.



Obrázok 8: Vizualizácia posúdenia a kontroly grafu,
zdroj: [44]

Slová verzus grafy:

Grafy sú mocným nástrojom, ktorý v sebe približuje sériu čísel. Čísla v grafe sprostredkujú informácie čitateľovi vizuálne aj naratívne. Rovnaký súbor čísel pôsobí konkrétnejšie a presnejšie, keď je graficky znázornený, než keď je opísaný v texte alebo titulku. Obrázok č. 10 naľavo zobrazuje čísla v texte a napravo sú graficky zobrazené čísla. [42]



Obrázok 9: Porovnanie vizualizácie číselných údajov,
zdroj: [44]

Výstupné formáty infografiky:

1. **Statické** - akákoľvek infografika, ako je definovaná vyššie, navrhnutá so zámerom na tlač (plagáty, schémy sprevádzajúce články v časopisoch resp. v novinách, v rámci reklám atď.), alebo digitálne použitie (na webových stránkach alebo pre zobrazenia obrazovky ako digitálne displeje) bez zahrnutia akéhokoľvek pohybu resp. animovaných prvkov alebo funkcií.

2. **Interaktívne** - v tomto prípade sa jedná o infografiku s prídavkom prvkov interakcie, ako sú posuvné lišty a tlačidlá prehrávania alebo akékoľvek iné pokročilé ovládacie prvky. Tieto interakcie umožňujú užívateľom prehrať, posunúť sa vpred, zastaviť, zopakovať alebo odhaliť konkrétne informácie, údaje resp. pohyb z displeja podľa ich preferencií pre oveľa pútavejšie a informatívne skúsenosti.
3. **Animované** - akákoľvek infografika podľa definície vyššie, navrhnutá so zámerom pre zobrazenie informácií na obrazovke (Motion grafické videá: na webových stránkach s videami ako sú YouTube, televízne reklamy, múzejné expozície a kiosky atď.), kde majú prvky zvyčajne konštantný pohyb a dáta sú v animovanej podobe. Pohyb alebo pohyby sú zvyčajne vytvorené pomocou počítačového animačného softvéru alebo programovacieho jazyka. [1], [17], [15]

3.7 Jazyk R ako nástroj na vizualizáciu údajov

Jazyk R je výkonný a všestranný programovací jazyk, ktorý sa široko používa na analýzu a vizualizáciu údajov. Jeho rozsiahly ekosystém balíkov a knižníc, špeciálne vyvinutých pre vizualizáciu dát, z neho robí populárnu voľbu pre vedcov, štatistikov a výskumníkov.

Jedným z hlavných prínosov používania R pre vizualizáciu údajov je jeho úzka integrácia s analýzou údajov. V rámci jediného prostredia je možné jednoducho manipulovať s údajmi, vykonávať štatistické analýzy a vytvárať vizualizácie.

Možnosti vizualizácie dát v R sú vysoko prispôsobiteľné. Môžeme doladiť každý aspekt svojich grafov, od farieb a mierok až po štítky a anotácie. Navyše máme možnosť vytvárať vlastné vizualizácie alebo upravovať existujúce balíky podľa našich konkrétnych požiadaviek.

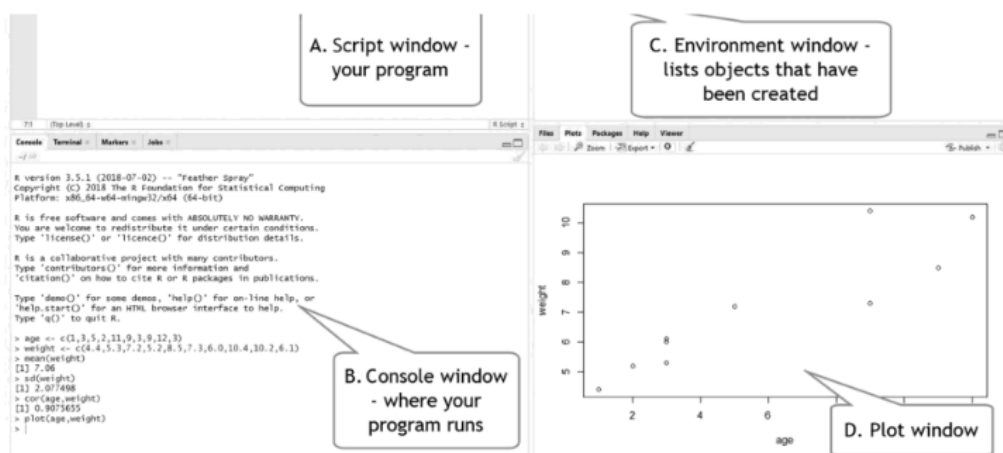
Jazyk R je vhodný na vytváranie grafiky v publikačnej kvalite. So správnymi balíkmi a nastaveniami dokážeme vytvárať obrázky s vysokým rozlíšením vhodné pre vedecké práce, správy a prezentácie. Má aktívnu a rozsiahlu používateľskú komunitu, čo znamená, že existuje bohatá dokumentácia, návody a podpora pre používateľov.

Spomínaný programovací jazyk možno integrovať aj s inými nástrojmi na analýzu a vizualizáciu údajov, ako je R Markdown (viď kap. 3.11.2), čo z neho robí všestrannú voľbu pre rôzne pracovné postupy.

Štandardné rozhranie pre jazyk R je veľmi základné a ponúka o niečo viac ako „command prompt“ v preklade príkazový riadok, na zadávanie kódov. Pre projekty v praxi

chceme komplexnejší nástroj na písanie kódu a prezeranie výstupu. Pre R bolo vyvinutých niekoľko takýchto nástrojov, nazývaných Integrated Development Environments (IDE), vrátane Eclipse so StatET, Visual Studio pre R a RStudio Desktop. [13], [18]

Vzhľadom na to, že praktickú časť budeme vykonávať v RStudiu, opíšeme toto prostredie, ktoré disponuje rôznymi oknami a kartami s nástrojmi na importovanie údajov, písanie kódu, ladenie chýb, vizualizáciu výstupu a tvorbu správ.



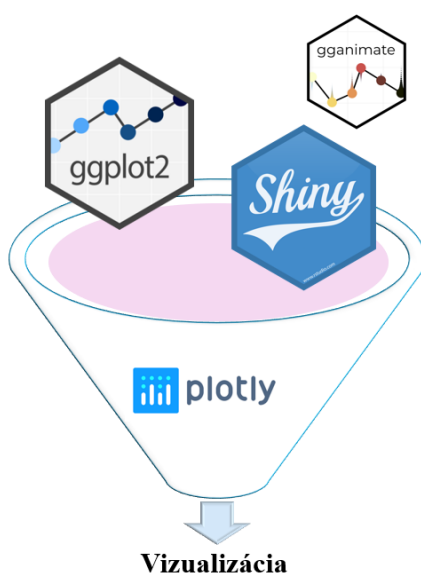
Obrázok 10: RStudio prostredie, zdroj: [20]

- A. **Skriptovacie okno** - je centrálna časť rozhrania RStudio, kde píšeme, upravujeme a spúšťame svoj R kód. Je navrhnuté tak, aby zefektívnilo proces kódovania, uľahčilo analýzu údajov a pomohlo nám vytvoriť dobre zdokumentované a reprodukovateľné analýzy alebo správy. Je to kľúčový pracovný priestor pre každého, kto pracuje s R na analýze údajov alebo programovaní.
- B. **Okno konzoly** - je základný komponent RStudia, ktorý poskytuje interaktívne a dynamické prostredie pre R programovanie a analýzu dát. Je to univerzálny pracovný priestor, kde môžeme rýchlo testovať a spúšťať kód, preskúmavať údaje a získať okamžitú spätnú väzbu od interpretéra jazyka R.
- C. **Okno prostredia** - Je užitočným nástrojom na monitorovanie a správu pracovného priestoru v R. Pomáha nám sledovať údaje a objekty, s ktorými pracujeme, poskytuje rýchly prístup k informáciám o týchto objektoch a umožňuje efektívne riadiť náš pracovný priestor. To môže byť užitočné najmä pri práci na projektoch analýzy údajov a keď chceme zabezpečiť, aby bola relácia R čistá a dobre organizovaná.
- D. **Grafické okno** - Je funkčný priestor v RStudiu určený na zobrazenie výsledkov R kódu vo forme grafických reprezentácií a tabuliek, umožňujúcich vizualizáciu a skú-

manie údajov. Taktiež umožňuje vytvárať kvalitné vizuálne prezentácie analýz s interaktívnymi funkciami a exportom do rôznych formátov na ďalšie využitie v správach, prezentáciách alebo dokumentoch.

R prichádza s rozsiahlymi funkciami a niektoré z jeho najzaujímavejších funkcií sú dostupné ako voliteľné moduly, ktoré si môžeme stiahnuť a nainštalovať. Existuje viac ako 10 000 užívateľských knižníc, nazývaných balíčky, ktoré si môžeme stiahnuť zo stránky <http://cran.r-project.org/web/packages>. Poskytujú obrovskú škálu nových schopností, od analýzy geopriestorových údajov cez spracovanie hmotnostného spektra až po analýzu psychologických testov. Jeden súbor balíčkov, spoločne nazývaný *tidyverse*, si zaslúži osobitnú pozornosť. Ide o kolekciu, ktorá ponúka efektívny, konzistentný a intuitívny prístup k manipulácii a analýze údajov. Výhody, ktoré ponúkajú balíčky *tidyverse* (s názvami ako *tidyr*, *dplyr*, *lubridate*, *stringr* a *ggplot2*) menia spôsob, akým dátoví vedci píšu kód v R. [20], [50]

V našej práci sa budeme zameriavať na využívanie vizualizačných balíčkov, ktoré sú zobrazené na obrázku č. 11 nižšie. Každý balíček si bližšie opíšeme v nasledujúcich kapitolách.



Obrázok 11: Zobrazenie využitých vizualizačných balíčkov v R programe, zdroj: vlastné spracovanie

3.8 R *ggplot2*

Ide o balík R určený na tvorbu štatistickej alebo dátovej grafiky, avšak na rozdiel od väčšiny iných grafických balíkov sa vyznačuje hlbokou základnou gramatikou. Táto gramatika, založená na Grammar of Graphics, sa skladá zo sady nezávislých komponentov, ktoré môžu byť zložené mnohými rôznymi spôsobmi. Vďaka tomu je *ggplot2* veľmi výkonný, pretože môžeme vytvárať nové grafiky, ktoré sú presne prispôbené nášmu problému. Je pre nás dôležité oboznámiť sa s balíčkom *ggplot2*, pretože nasledujúce balíčky *plotly* a *gganimate* s ním úzko súvisia a práve na ne upriamime našu pozornosť v praktickej časti. [42]

Kľúčové vlastnosti balíka *ggplot2*:

1. **Gramatika grafiky:** *ggplot2* je postavený na koncepte Grammar of Graphics, ktorý poskytuje štruktúrovaný spôsob uvažovania o vizualizácii dát. Rozdeľuje graf na komponenty, ako sú údaje, estetika, geometrie, fazety a štatistiky. Tento prístup uľahčuje vytváranie zložitých vizualizácií špecifikovaním komponentov grafu krok za krokom.
2. **Konštrukcia vrstveného grafu:** *ggplot2* vytvára grafy vrstveným spôsobom. Začneme s údajmi, definujeme estetiku (ako sú premenné mapované na vizuálne vlastnosti) a potom pridáme vrstvy geometrií (napr. body, čiary, pruhy), aby sme vytvorili graf. Tento vrstvený prístup umožňuje vytvárať sofistikované vizualizácie.
3. **Estetika a mapovanie:** estetické mapovania definujú ako sú premenné vizuálne reprezentované v grafe. Premenné sa dajú mapovať na estetiku, ako je farba, veľkosť, tvar a poloha, čo umožňuje vizuálne kódovať údaje rôznymi spôsobmi.
4. **Rozsiahle typy grafov:** *ggplot2* podporuje širokú škálu typov grafov, vrátane bodových grafov, čiarových grafov, stĺpcových grafov, histogramov, krabicových grafov, husľových grafov, tepelných máp a ďalších. Je vysoko všestranný a prispôsobiteľný rôznym potrebám vizualizácie údajov.
5. **Transformácia údajov:** *ggplot2* umožňuje transformáciu údajov pomocou funkcií ako `filter()`, `mutate()`, `group_by()` a `Summary()`, čo umožňuje predspracovanie a manipuláciu s údajmi priamo vo vykresľovacej linke.
6. **Fazetovanie:** *ggplot2* podporuje fazetovanie, ktoré umožňuje vytvárať malé násobky alebo mriežku grafov na základe jednej alebo viacerých premenných. Je to užitočné pri vizualizácii podmnožín údajov.

7. **Témy a prispôsobenie:** vzhľad grafov `ggplot2` si môžeme prispôbiť pomocou tém. Motívy ovládajú vlastnosti, ako sú písma, farby, čiary mriežky a ďalšie vizuálne prvky. K svojim grafom môžeme tiež pridať názvy, štítky a anotácie.
8. **Integrácia s dátovými rámcami:** `ggplot2` je navrhnutý tak, aby bezproblémovo pracoval s dátovými rámcami. Dátové rámce môžeme použiť na ukladanie a manipuláciu s údajmi, čo uľahčuje vytváranie vizualizácií zo štruktúrovaných množín údajov.
9. **Ekosystém ggplot2:** v prostredí `ggplot2` sa nachádzajú doplnky a balíky ako `ggplot2()`, `ggthemes()` a `ggtree()`, ktoré využívajú a rozširujú funkcie `ggplot2` na špecifické účely.
10. **Prieskum údajov a komunikácia:** `ggplot2` je výkonný nástroj na prieskum údajov a komunikáciu. Umožňuje vytvárať zmysluplné vizualizácie, ktoré odhaľujú pohľady na dáta a uľahčujú efektívnu dátovú komunikáciu. [18], [29]

3.8.1 Syntax `ggplot2` a prispôsobenie grafu

Syntax `ggplot2` sa opiera o dva základné koncepty: geometrické objekty a gramatiku grafiky. Geometrické objekty predstavujú rôzne vizuálne štruktúry, ktoré slúžia na vizualizáciu údajov. Gramatika grafiky je syntax, ktorú používame pre rôzne estetiky grafu, ako je napríklad mierka súradníc, písma, farebné motívy atď. Pomocou vrstveného konceptu Grammar of Graphics je možné v `ggplot2` zostaviť graf vo vrstvách.

Údaje (`data`), sú komponenty, ktoré chceme vizualizovať a súbor estetických mapovaní popisuje, ako sa premenné v údajoch mapujú na estetické atribúty, ktoré môžeme vnímať. Geometrické objekty (`geoms`), predstavujú to, čo skutočne vidíme na grafe: body, čiary, polygóny atď. Štatistické transformácie (`stats`), sumarizujú údaje mnohými užitočnými spôsobmi. Napríklad binovanie a počítanie pozorovaní na vytvorenie histogramu alebo zhrnutie 2D vzťahu s lineárnym modelom. Štatistiky sú voliteľné, ale veľmi užitočné. Mierky (`scales`) mapujú hodnoty v dátovom priestore na hodnoty v estetickom priestore, či už ide o farbu, veľkosť alebo tvar. Miery vykresľujú legendu alebo osi, ktoré poskytujú inverzné mapovanie, aby bolo možné z grafu prečítať pôvodné hodnoty údajov. Súradnicový systém (`coord`), popisuje, ako sú súradnice údajov mapované do roviny grafiky. Poskytuje tiež osi a mriežky, aby bolo možné čítať graf. Bežne používame karteziánsky súradnicový systém, ale je k dispozícii množstvo ďalších, vrátane polárnych súradníc a mapové projekcie.

Špecifikácia fazetovania (*facet*) popisuje, ako rozdeliť údaje do podmnožín a ako tieto podmnožiny zobraziť ako malé násobky. Toto je tiež známe ako kondicionovanie alebo mriežkovanie/mrežovanie. [27], [42]

3.8.2 Proces tvorby statického grafu pomocou *ggplot2*

Inštalácia



Balík *ggplot2* nájdeme na CRAN (vďaka čomu je jeho inštalácia veľmi jednoduchá):

```
install.packages("ggplot2")
```

Potom ho môžeme jednoducho načítať s nasledujúcim kódom:

```
library(ggplot2)
```

Alebo môžeme začiarknuť políčko pred názvom balíka na paneli balíkov RStudio:

	Name	Description	Version		
<input checked="" type="checkbox"/>	ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.4.4		

Obrázok 12: Zobrazenie balíka *ggplot2* v paneli balíkov RStudio, zdroj: vlastné spracovanie

Geometrické objekty balíka *ggplot2* a vizualizácia grafu

Geometrické objekty, označované aj ako *geoms*, sú zodpovedné za skutočné vykresľovanie vrstiev v grafe a ovplyvňujú typ grafu, ktorý vytvárame. Dôležitým aspektom je správny výber *geoms*, pretože to ovplyvňuje vzhľad a štruktúru našej vizualizácie. Je dôležité mať jasnú predstavu o tom, ako chceme svoje údaje vizualizovať a aký dojem chceme dosiahnuť.

Mali by sme tiež poznať premenné, ktoré chceme vo svojej grafike zobraziť. V balíku *ggplot2* je k dispozícii mnoho rôznych geometrických objektov, z ktorých si môžeme vybrať podľa našich požiadaviek. V podstate sa geometrické objekty delia podľa toho, koľko premenných chceme vizualizovať. [18]

Jednorozmerné, dvojrozmerné a trojrozmerné objekty, ktoré sa používajú na pochopenie a vizualizáciu charakteristík jednej premennej sú tieto:

One Variable	Two Variables
Continuous Variable <ul style="list-style-type: none"> • geom_area • geom_density • geom_dotplot • geom_freqpoly • geom_histogram 	Continuous X, Continuous Y variable <ul style="list-style-type: none"> • geom_blank • geom_jitter • geom_point • geom_point • geom_quantile • geom_rug • geom_smooth • geom_text
Discrete Variable <ul style="list-style-type: none"> • geom_bar 	Discrete X, Continuous Y variable <ul style="list-style-type: none"> • geom_bar • geom_boxplot • geom_dotplot • geom_violin
Three Variables <ul style="list-style-type: none"> • geom_contour • geom_raster • geom_tile 	Continuous Bivariate Distribution <ul style="list-style-type: none"> • geom_bin2d • geom_density2d • geom_hex

Obrázok 13: Ukážka *geoms* pri jedno-, dvoj- a trojrozmerných objektoch, zdroj: [42]

Ako už bolo spomenuté, jednou z veľkých výhod `ggplot2` je, že môžeme zmeniť takmer všetky prvky podľa našich individuálnych potrieb. V nasledujúcej časti si ukážeme niekoľko spôsobov, ako sa dajú prispôsobiť vizualizácie.

Veľmi praktickou funkciou `ggplot2` je spôsob, akým môžeme „podmnožiť“ údaje pomocou funkcie `subset()`. Stačí do grafu pridať volanie funkcie podmnožiny, aby sme si vizualizovali určité hodnoty. Podmnožinu, ktorú chceme vizualizovať, je možné nastaviť pre všetky *geoms*:

```
ggplot(subset(iris, Species == "setosa")) +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width))
```

V tomto kóde sme najprv použili funkciu `subset()` na vytvorenie podmnožiny dátového rámca `iris`, kde hodnota pre premennú `Species` je "setosa". Potom sme túto podmnožinu použili vo funkcii `ggplot`, kde sme definovali typ grafu a estetické mapovanie v `geom_point()` pre vizualizáciu dvojice hodnôt `Sepal.Length` na osi *x* a `Sepal.Width` na osi *y*. V tomto kóde je operátor (+) využívaný v knižnici `ggplot2` na postupné pridávanie rôznych prvkov do grafu. Použitie tohto operátora umožňuje "spájať" viacero príkazov, ktoré definujú rôzne časti grafu, do jedného celku. [22]

V `ggplot2` je veľmi jednoduché pridávať názvy do grafov. Preto poskytuje funkciu `ggtitle()`. Môžeme ho jednoducho pridať do grafu pomocou operátora plus (+).

```
d <- ggplot(iris, aes(Species, Sepal.Length, fill = Species)) +  
  geom_bar(stat = "identity")  
d + ggtitle("Iris data: Species vs Sepal Length")
```

Zmena označenia osí je rovnako jednoduchá ako pridanie názvu, pretože `ggplot` ponúka funkcie `scale_x_continuous()` a `scale_y_continuous()`. Tieto funkcie možno použiť na zmenu všetkých nastavení týkajúcich sa osí.

Poskytovanie jednoduchého spôsobu, ako vymeniť osi x a y v grafe, je možné pomocou funkcie `coord_flip()` v `ggplot2`, čo umožňuje rýchlu zmenu orientácie grafu. Balíček ponúka so svojou možnosťou motívov spôsob, ako dosiahnuť, aby vizualizácie vyzerali oveľa lepšie a vynikli v porovnaní s takmer každým grafom vytvoreným pomocou základného vykresľovania. Okrem toho je tu veľmi zaujímavý balíček od Jeffreyho B. Arnolda s názvom `ggthemes`. Tento balík pridáva celý rad ďalších možností k existujúcim témam a mierkam `ggplot2`. Obsahuje napríklad tému podľa návrhu časopisu *Economist* alebo tému podľa návrhu denníka *Wall Street Journal*. [20], [27]

3.8.3 Zobrazenie grafov `ggplot2` a ich výstupný formát

1. Uloženie grafu do súboru bez jeho vykreslenia

V niektorých prípadoch môže byť žiaduce uložiť graf priamo z kódu bez manuálneho ukladania. Tento prístup sa môže ukázať veľmi užitočným, najmä ak máme spustené skripty na vytváranie štandardných grafov. V tomto prípade môžeme do našich skriptov vložiť kód, aby sme vykonali analýzu a vytvorili štandardné grafy. Existuje niekoľko rôznych funkcií, ktoré nám umožňujú uložiť naše grafy v rôznych formátoch súborov. Tieto funkcie pracujú otvorením grafického zariadenia, vykreslením grafu na tomto zariadení a následným uzavretím zariadenia. Tým sa zabezpečí, že graf sa nezobrazí priamo v konzole R, ale bude uložený priamo v grafickom zariadení. Medzi najdôležitejšie formáty súborov, ktoré môžeme vytvárať, patria PNG, PDF a JPEG.

- `png(filename, width, height)`
- `pdf(file, width, height)`
- `jpeg(filename, width, height)`

2. R Markdown

Pri používaní R Markdownu na tvorbu správ alebo dokumentov máme možnosť vložiť grafy vytvorené pomocou `ggplot2` priamo do našich dokumentov.

3. Shiny

V aplikácii Shiny môžeme použiť `ggplot2` na vytváranie grafov a ich zobrazenie. Grafy môžu byť aktualizované dynamicky na základe vstupu používateľa. [22], [23]

3.9 R *plotly*

Existujú dva hlavné spôsoby, ako vytvoriť objekt `plotly`: buď pretransformovaním objektu `ggplot2` pomocou `ggplotly()` na objekt `plotly`, alebo priamou inicializáciou objektu `plotly` pomocou funkcií `plot_ly()/plot_geo()/plot_mapbox()`. Oba prístupy sú navyše implementáciou Grammar of Graphics a oba sú poháňané knižnicou grafov JavaScript `plotly.js`, takže mnohé z rovnakých konceptov a nástrojov, ktoré sa naučíme pre jedno rozhranie, možno znova použiť v druhom. [9]

- graf založený na `ggplot2`

```
plot(method, data_idx = 1, output_idx = NULL, aggr_channels = "sum",
      as_plotly = FALSE, same_scale = FALSE)
```

- graf založený na `plotly`

```
plot(method, data_idx = 1, output_idx = NULL, aggr_channels = "sum",
      as_plotly = TRUE, same_scale = FALSE)
```

Kľúčové vlastnosti balíka *plotly*:

1. **Interaktívna vizualizácia:** `plotly` nám umožňuje vytvárať interaktívne grafy. Používatelia môžu priblížiť, posúvať, umiestniť kurzor myši na dátové body, získať informácie a vybrať podmnožiny údajov na bližšie preskúmanie, čím sa zvýši interaktivita a používateľský zážitok z vizualizácií.
2. **Integrácia s `ggplot2`:** `plotly` je možné použiť v spojení s `ggplot2`, ďalším populárnym balíkom vizualizácie údajov v R. To nám umožňuje vziať naše statické grafy `ggplot2` a previesť ich na interaktívne grafy pomocou funkcie `ggplotly()`.

3. **Bohaté typy grafov:** `plotly` podporuje širokú škálu typov grafov vrátane bodových grafov, čiarových grafov, stĺpcových grafov, koláčových grafov, bublinových grafov, rámečkových grafov, tepelných máp, 3D plošných grafov, choropletových máp a ďalších. Vďaka tejto rozmanitosti je univerzálny pre rôzne potreby vizualizácie údajov.
4. **Webové dashboardy (webové tabule):** s pomocou `plotly` môžeme vytvárať interaktívne webové dashboardy, ktoré obsahujú viacero grafov a vizualizácií. Tieto dashboardy je možné zdieľať online, čo ich robí užitočnými pre prezentáciu a komunikáciu dátových informácií.
5. **Prispôbenie:** pomocou `plotly` si dokážeme prispôsobiť vzhľad svojich grafov. To zahŕňa zmenu farieb, písma, štítkov, nadpisov, legend a ďalších. Umožňuje doladenie vizuálnych prvkov našich grafov pomocou rozhrania API („Application Programming Interface“).
6. **Spolupráca a zdieľanie:** interaktívne vizualizácie v `plotly` môžeme zverejniť v cloudovom prostredí `plotly`, čo nám umožňuje spolupracovať s kolegami a zdieľať naše grafy s širším publikom.
7. **Vstavané miniaplikácie:** `plotly` poskytuje miniaplikácie a ovládacie prvky, ktoré možno pridať do našich interaktívnych informačných panelov, čo používateľom umožňuje filtrovať, triediť alebo upravovať údaje zobrazené za behu.
8. **Integrácia R Markdown:** `plotly` možno integrovať do dokumentov R Markdown, čo nám umožní vytvárať dynamické a interaktívne zostavy, ktoré sú efektívne pre príbehy založené na údajoch.
9. **Komunita a dokumentácia:** `plotly` má aktívnu používateľskú komunitu a komplexnú dokumentáciu, vďaka čomu je ľahké nájsť podporu pre naše projekty pri vizualizácii údajov.
10. **Kompatibilita:** hoci má `plotly` väzby R, neobmedzuje sa len na R. Je to všestranná knižnica s implementáciami v iných programovacích jazykoch, ako sú Python a JavaScript, takže naše vizualizácie môžu byť konzistentné v rôznych prostrediach. [1], [6], [23]

3.9.1 Syntax R plotly a prispôsobenie grafu

V jazyku R je k dispozícii všestranný balík `plotly`, ktorý umožňuje tvorbu interaktívnych a esteticky príťažlivých vizualizácií údajov. Syntax funkcie `plot_ly` zahŕňa niekoľko kľúčových argumentov, ktoré si opíšeme nižšie.

```
plot_ly (data, x, y, type, mode, color, size, ....)
```

- *data* - dátový rámec
- *x,y* - hodnoty osí x a y
- *type* - určuje typ grafu alebo stopy, ako napríklad „histogram“, „rozptyl“, „pruh“, „box“, „teplotná mapa“, „histogram“, „histogram2d“, „histogram2dcontour“, „koláčový graf“, „obrys“, „scatter3d“
- *mode* - určuje režim, ako napríklad čiara, body, značky
- *color* - určuje farbu dátových bodov, ktorá je zvyčajne funkciou `I()`, aby sa predišlo zmene mierky
- *colors* - paleta `colourbrewer`, vektor farieb v hexa formáte
- *size* - názov alebo výraz, ktorý definuje veľkosť dátových bodov

Na formátovanie osí môžeme použiť ďalšie funkcie ako `layout()`. Do grafu môžeme pridať aj anotácie vďaka funkcii `add_annotatons()`. Funkcia `add_annotatons()` obsahuje nasledujúce argumenty.

- *x* - nastavenie anotácií x polohy súradníc (hodnota osí x na základe súboru údajov)
- *y* - nastavenie anotácií y polohy súradníc (hodnota osí y na základe súboru údajov)
- *text* - text spojený s anotáciou
- *showarrow* - 0 alebo 1 v závislosti od toho, či sa má šípka zobrazit' alebo nie
- *xref* - hovorí `plotly`, aby nastavil x odkaz na graf (v tomto prípade $x=0$ znamená najviac vľavo, $x=1$ znamená najviac vpravo)
- *yref* - hovorí `plotly`, aby nastavil y odkaz na graf (v tomto prípade $y=0$ znamená spodok, $y=1$ znamená horný) [9], [28]

Funkcia `plot_ly()` má množstvo argumentov, ktoré sú jedinečné pre balík R (napr. farba, ťah, rozsah, symbol, typ čiary atď.) a uľahčujú kódovanie dátových premen-

ných ako vizuálnych vlastností (napr. farba). Štandardne tieto argumenty mapujú hodnoty dátovej premennej na vizuálny rozsah definovaný množným číslom argumentu.

3.9.2 Proces tvorby interaktívneho grafu pomocou R plotly

Inštalácia a načítanie R Plotly



Balík plotly nainštalujeme pomocou tohto kódu:

```
install.packages("plotly")
```

Potom ho môžeme jednoducho načítať s nasledujúcim kódom:

```
library(plotly)
```

Alebo môžeme začiarknuť políčko pred názvom balíka na paneli balíkov RStudio:

	Name	Description	Version		
<input checked="" type="checkbox"/>	plotly	Create Interactive Web Graphics via 'plotly.js'	4.10.2		

Obrázok 14: Zobrazenie balíka *plotly* v paneli balíkov RStudio, zdroj: vlastné spracovanie

Variety vizualizácie *plotly*

V tejto fáze sa rozhodneme, ktorý typ grafu najlepšie a najefektívnejšie zobrazí naše údaje. Niektoré typy sme nižšie popísali, nejedná sa však o všetky. Vybrali sme tie najzákladnejšie ale aj tie, s ktorými sa stretneme v praktickej časti.

Jednoduché stopy (*Simple traces*)

V `plotly` je `trace` (stopa) údajový objekt, ktorý určuje, ako sa má vykresliť konkrétna množina údajov. Obsahuje informácie o údajoch, ako sú súradnice x a y , typ grafu (rozptyl, pruh, čiara atď.) a rôzne možnosti štýlu.

1. Bodová stopa (*Scatter Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="scatter"[, ...])  
add_trace(p, type="scatter"[, ...])
```

Typ bodového sledovania zahŕňa čiarové grafy, bodové grafy, textové grafy a bublinové grafy. Údaje vizualizované ako bod rozptylu alebo čiary sú nastavené v x a y . Text (zo-

brazuje sa buď na grafe alebo iba pri umiestnení kurzora myši) je cez „text“. Bublinové grafy sa dosahujú nastavením `marker.size` a/alebo `marker.color` na číselné polia.

2. Stĺpcová stopa (*Bar Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="bar" [, ...])
add_trace(p, type="bar" [, ...])
```

Údaje vizualizované pomocou rozpätia pruhov sú nastavené ako y , ak je „orientácia“ nastavená na v (predvolená hodnota) a štítky sú nastavené na x . Nastavením „orientácie“ na h sa role vymenia.

3. Koláčová stopa (*Pie Trace*)

Inicializuje sa pomocou `plot_ly()` or `add_trace()`:

```
plot_ly(df, type="pie" [, ...])
add_trace(p, type="pie" [, ...])
```

Údaje vizualizované podľa sektorov koláča sú nastavené v „hodnotách“. Označenia sektorov sú nastavené v „štítkoch“. Farby sektorov sú nastavené v `marker.colors`.

4. Stopa teplotnej mapy (*Heatmap Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="heatmap" [, ...])
add_trace(p, type="heatmap" [, ...])
```

Údaje, ktoré definujú mapovanie hodnôt teplotnej mapy na farbu, sú špecifikované v osi z . Tieto údaje môžu byť usporiadané buď ako 2D zoznam, alebo ako 1D pole hodnôt. V prípade, že zoznam v osi z je 2D, kde má n riadkov a m stĺpcov, výsledná teplotná mapa bude mať zvyčajne n oddielov pozdĺž osi y a m oddielov pozdĺž osi x . Inak povedané, bunka v i -tom riadku/ j -tom stĺpci v zozname z je mapovaná na i -tý oddiel osi y (počínajúc od spodnej časti grafu) a j -tý oddiel osi x (počínajúc zľava od grafu). Toto usporiadanie je možné zmeniť pomocou funkcie `transpose()`. Okrem toho, os x (y) môže byť definovaná buď s m alebo $m + 1$ (n alebo $n + 1$) prvkami. Ak je m (n), potom súradnice zodpovedajú stredu buniek teplotnej mapy a bunky majú rovnakú šírku. V prípade $m + 1$ ($n + 1$), súradnice zod-

povedajú okrajom buniek teplotnej mapy. Ak sú údaje v osi z uvedené ako 1D zoznam, súradnice x a y musia byť explicitne definované v osiach x a y pre vytvorenie trojíc údajov. [35]

Distribučné stopy (*Distribution Traces*)

V `plotly` môžeme vytvárať distribučné stopy na vizualizáciu distribúcie údajov pomocou rôznych typov grafov, ako sú histogramy a krabicové grafy.

1. Krabicová stopa (*Box Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="box"[, ...])  
add_trace(p, type="box"[, ...])
```

Každý box sa rozprestiera od kvartilu 1 (Q1) po kvartil 3 (Q3). Druhý kvartil (Q2, t.j. medián) je označený čiarou vo vnútri rámčeka. Grafy rastú smerom von od okrajov škatúľ, štandardne presahujú $\pm 1,5$ -násobok medzikvartilového rozsahu (IQR: Q3-Q1). Hodnoty a pozície zodpovedajúce jednotlivým poliam možno zadať pomocou dvoch podpisov. Prvý podpis očakáva, že používatelia dodajú vzorové hodnoty v dátovom poli y pre zvislé polia (x pre vodorovné polia). Zadaním pol'a $x(y)$ sa vykreslí jeden rámček na každú odlišnú hodnotu $x(y)$. Ak nie je uvedený žiadny zoznam $x(y)$, vykreslí sa jeden rámček. V tomto prípade je rámček umiestnený so stopou „name“ alebo x_0 (y_0), ak je k dispozícii. [9], [14]

2. Stopa histogramu (*Histogram Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="histogram"[, ...])  
add_trace(p, type="histogram"[, ...])
```

Vzorové údaje, z ktorých sa vypočítava štatistika, sa nastaví v x pre vertikálne rozložené histogramy a v y pre horizontálne rozložené histogramy. Ak nie sú poskytnuté žiadne agregáčnne údaje, hodnoty sú nastavené na $xbins$ a $ybins$.

Finančné stopy (*Finance Traces*)

V jazyku R nám všestranný balík `plotly` umožňuje vytvárať interaktívne vizualizácie finančných a časových radov. Tento balík je užitočný pre vytváranie rôznorodých finančných grafov a vizualizácií v kontexte analýzy cien akcií, portfólia a hodnotenia rizík.

1. Stopa Vodopád (*Waterfall Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="waterfall"[, ...])  
add_trace(p, type="waterfall"[, ...])
```

Graf vodopádu je užitočným nástrojom na vizualizáciu príspevku jednotlivých prvkov (pozitívnych alebo negatívnych) v stĺpcovom grafe. Údaje, ktoré sú zobrazené pomocou rozpätia pruhov, sú definované ako y , ak je parameter „orientácia“ nastavený na v (čo je predvolená hodnota), pričom štítky sú nastavené na x . Ak je „orientácia“ nastavená na h , dochádza k výmene úloh.

3D stopy (*3D Traces*)

V `plotly` môžeme vytvárať 3D stopy na vizualizáciu údajov v troch rozmeroch, čo je užitočné na vizualizáciu 3D údajov, ako sú povrchové grafy, bodové grafy a iné typy 3D vizualizácií.

1. 3D bodová stopa (*Scatter3D Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="scatter3d"[, ...])  
add_trace(p, type="scatter3d"[, ...])
```

Údaje, ktoré sa zobrazujú ako body rozptylu alebo čiary v trojrozmernom priestore, sú definované pomocou parametrov x , y a z . Text, ktorý sa zobrazuje buď priamo na grafe alebo sa zobrazuje po umiestnení kurzora myši, je určený pomocou parametra „text“. Veľkosť a/alebo farba bublín sa upravujú nastavením `marker.size` a `marker.color`. Projekcie sú nastavované cez parameter „projekcia“.

2. Povrchová stopa (*Surface Trace*)

Inicializuje sa pomocou `plot_ly()` alebo `add_trace()`:

```
plot_ly(df, type="surface"[, ...])  
add_trace(p, type="surface"[, ...])
```

Súradnice povrchu sú definované pomocou parametra z . Údaje v parametri z by mali byť vo formáte 2D zoznamu. Súradnice v parametroch x a y môžu byť buď 1D zoznamy ale-

bo 2D polia (napríklad na zobrazenie grafu parametrických plôch). Ak nie sú uvedené v parametroch x a y , predpokladá sa, že súradnice x a y sa lineárne zvyšujú od 0 s krokom jednotky. Farebná škála sa predvolene prispôsobuje hodnotám v parametri z . Na vlastné farebné škály sa používa parameter `surfacecolor`. [8], [31], [34]

3.9.3 Zobrazenie a výstupný formát grafu R plotly

Máme viacero možností, ako exportovať interaktívne grafy do rôznych výstupných formátov. Môžeme si vybrať ten, ktorý najviac vyhovuje našim požiadavkám na zdieľanie alebo vkladanie vizualizácií. Tu sú základné formáty výstupu pre grafy `plotly` v R:

1. Statické obrázky

Akýkoľvek `plotly` objekt možno uložiť ako statický obrázok pomocou funkcie `orca()`. Ak ho chceme použiť, budeme potrebovať `orca(CLI)`, nástroj príkazového riadka. CLI je možné nainštalovať cez `node.js` alebo `conda`.

Export obrázka z prehliadača je vhodná možnosť, ak potrebujeme pred exportom vykonať úpravy alebo ak chceme, aby ostatní zdieľali našu prácu. V štandardnom nastavení, kliknutím na ikonu „Stiahnuť graf“ v paneli nástrojov sa obrázok stiahne vo formáte PNG a použije sa aktuálna výška a šírka grafu, pričom tieto predvolené hodnoty je možné upraviť prostredníctvom konfigurácie grafu.

2. Webová stránka (HTML)

Každý widget možno uložiť ako samostatný súbor HTML pomocou funkcie `htmlwidgets::saveWidget()`. V predvolenom nastavení vytvára úplne samostatný súbor HTML, čo znamená, že všetky potrebné súbory závislostí JavaScript a CSS sú zviazané v súbore HTML. Vďaka tomu je veľmi jednoduché zdieľať miniaplikáciu ako jeden súbor HTML.

3. R Markdown

Existuje možnosť integrovať `plotly` do dokumentov R Markdown, čo umožňuje vytváranie dynamických a interaktívnych správ, ktoré sú efektívne pre príbehy založené na údajoch. Grafy `plotly` môžeme vložiť do dokumentov R Markdown a následne vykresliť dokument do HTML, PDF alebo prezentácie.

4. Shiny

V aplikácii Shiny môžeme zobraziť grafy `plotly` priamo vo svojej aplikácii. Interaktivita `plotly` je v Shiny zachovaná, čo umožňuje používateľom interagovať s grafmi. Na zobrazenie grafov `plotly` v aplikácii Shiny zvyčajne použijeme funkcie `plotlyOutput()` a `renderPlotly()`. [23], [39]

3.10 R *ganimate*

Niektor by mohol tvrdiť, že animácie predstavujú posvätný grál v oblasti vizualizácie údajov. Na rozdiel od statických obrázkov sa animácie pohybujú, menia sa a zobrazujú zmeny v údajoch v čase (alebo v iných premenných). Na prvý pohľad sa môžu animácie zdať veľmi zložité, ale v skutočnosti nie je ich vytvorenie také náročné. To, čo vnímame ako nepretržitý pohyb, je v skutočnosti séria jednotlivých obrázkov zobrazených postupne vo veľmi rýchlej sekvencii. [30]

Rovnako platí, že animácie údajov sú viacnásobné grafické znázornenia idúce jedna po druhej podľa logickej postupnosti (najčastejšie časovej osi). Všetko, čo sme opísali, môžeme v R realizovať pomocou balíka `ganimate`. Aj keď sú animácie jednou z najsofistikovanejších vizualizačných techník, vyžadujú si aj maximálnu opatrnosť. Analytici si musia vždy položiť otázku, či animácia robí údaje zrozumiteľnejšími alebo ide o nadbytočné zdobenie, ktoré môže viac rozptýliť, ako zlepšiť.

Kľúčové vlastnosti balíka *ganimate*:

1. **Integrácia s *ggplot2*:** `ganimate` sa bezproblémovo integruje s `ggplot2`, jedným z najpoužívanejších balíkov vizualizácie dát v R. Pridáva možnosti animácie do grafov `ggplot2`, ktorý sa riadi princípmi Grammar of Graphics pre vytváranie grafov.
2. **Typy animovaných grafov:** `ganimate` nám umožňuje vytvárať širokú škálu animovaných typov grafov, vrátane animovaných bodových grafov, čiarových grafov, pruhových grafov, tepelných máp a ďalších. Môžeme animovať údajové body, čiary alebo iné geometrie.
3. **Prechodové rámy:** `ganimate` funguje na základe konceptu prechodových rámcov. Umožňuje nám určiť, ako sa naše údaje menia z jedného rámca na druhý, čo definuje animáciu. Na základe údajov môžeme vytvárať plynulé prechody alebo postupné animácie.

4. **Flexibilné ovládanie animácie:** máme kontrolu nad procesom animácie, čo nám umožňuje prispôbiť nastavenia animácie, ako je trvanie animácie, funkcie nábehu a interpolácia snímok.
5. **Rozšíriteľné a prispôsobiteľné:** `gganimate` je možné rozšíriť o rôzne témy a možnosti animácií. Môžeme vytvoriť vlastné animácie alebo použiť preddefinované témy, aby sme sa ubezpečili, že naše animácie budú vizuálne príťažlivé.
6. **Dynamický prieskum údajov:** animácia je výkonný nástroj na skúmanie dynamických údajov. `Gganimate` pomáha identifikovať vzory a trendy v našich údajoch, ktoré nemusia byť okamžite zrejmé v statických vizualizáciách.
7. **Prípady použitia:** `gganimate` je cenný pre širokú škálu aplikácií, ako je zobrazovanie časových zmien v súboroch údajov, ilustrovanie vývoja priestorových údajov a vysvetľovanie dynamických procesov vo vedeckom výskume alebo prezentáciách.
8. **Skúmanie údajov a rozprávanie príbehov:** animácia je výkonný nástroj na skúmanie dynamických údajov, znázorňovanie zmien v priebehu času a sprostredkovanie príbehov prostredníctvom vizualizácie údajov. Pomocou `gganimate` je možné identifikovať vzory a trendy v našich údajoch, ktoré v statických vizualizáciách nemusia byť okamžite zrejmé. [24], [40]

3.10.1 Syntax R gganimate a prispôsobenie animácie

Balík `gganimate` používa gramatiku grafiky rovnako ako balík `ggplot2`, jedná sa však o rozdielne funkcie. Túto všeobecnú syntax je možné prispôbiť tak, aby vyhovovala našim špecifickým požiadavkám na údaje, graf a animáciu. Celý rad prechodových funkcií a nastavení na vytváranie dynamických a vizuálne príťažlivých animácií z grafov `ggplot2` je možné nájsť v balíku `gganimate`.

Funkcia *transition*

Definuje, ako sa vrstvy údajov menia s časom. Vrstvy môžu byť statické alebo animované rovnakým spôsobom, akým je možné vrstvy rozdeliť alebo opakovať naprieč fázami. V balíku `gganimate` sú prechody základným konceptom, ktorý riadi, ako animácia postupuje z jedného stavu do druhého. Tieto prechody definujú správanie animácie a umožňujú určiť, ako sa údaje v priebehu času menia. Prechody sú kľúčovou súčasťou vytvárania

dynamických a pútavých vizualizácií údajov. Tu sú niektoré z bežne používaných prechodov v `gganimate`:

1. `transition_states()`: sa používa, keď máme kategorickú premennú, ktorá definuje rôzne stavy v údajoch. Často sa používa na vytváranie animácií, ktoré ukazujú, ako sa menia údaje medzi jednotlivými kategóriami. Môžeme ho napríklad použiť na animáciu údajov za rôzne roky, mesiace alebo akúkoľvek kategorickú premennú.
2. `transition_time()`: je navrhnutý pre údaje založené na čase. Používa sa, keď údaje majú časovú premennú a chceme vytvoriť animácie založené na čase. Môžeme určiť časovú jednotku (napr. roky, mesiace, dni), aby sme ovládali podrobnosť času animácie.
3. `transition_states_and_ranges()`: Umožňuje špecifikovať stavy aj časy prechodu manuálne kombinovaním `transition_states()` s `transition_manual()`. Je to užitočné pri kontrole animácie prostredníctvom definovania konkrétnych stavov a zároveň aj časových bodov.
4. `transition_layers()`: sa používa na vytváranie animácií, ktoré zahŕňajú vrstvenie rôznych grafov na seba. Umožňuje zobrazit' vývoj viacerých vrstiev v animácii.
5. `transition_reveal()`: sa používa na vytvorenie špecifického typu animačného prechodu, ktorý odhaľuje údaje v priebehu času. Tento prechod je užitočný najmä vtedy, keď chceme postupne odhaľovať dátové body v grafe, aby to vyzeralo, akoby sa dáta v priebehu animácie odkrývali. Často sa používa na animácie údajov založené na čase alebo na zdôraznenie časového aspektu údajov.

Funkcia *view*

Funkcia `view()` slúži na ovládanie pomeru strán, úrovne priblíženia a posúvania (úprava viditeľnej oblasti) animovaného grafu. Táto funkcia je užitočná najmä vtedy, keď chceme prispôbiť zobrazenie animácie, aby sme zdôraznili špecifické aspekty našich údajov.

1. `view_follow()`: sa používa na vytvorenie animácie, ktorá sleduje konkrétny údajový bod alebo oblasť záujmu, keď sa pohybuje v animácii. Táto funkcia je užitočná, keď potrebujeme zachovať zaostrenie na konkrétny údajový bod alebo oblasť, kde sa naše údaje časom vyvíjajú.

2. `view_step()`: sa používa na vytvorenie animácie s postupnými zmenami v zobrazení. Táto funkcia je užitočná, keď meníme zobrazenie v sérii krokov, ako je posúvanie cez veľkú oblasť, priblíženie, oddialenie alebo prispôbenie zobrazenia špecifickým oblastiam našich údajov.
3. `view_zoom()`: slúži na vytvorenie animácie s postupnými zmenami úrovne priblíženia pohľadu. Táto funkcia je užitočná pri priblížení alebo oddialení zobrazenia, aby sme počas animácie zdôraznili špecifické aspekty údajov.
4. `view_static()`: sa používa na vytvorenie animácie so statickým zobrazením, čo znamená, že zobrazenie zostáva rovnaké počas celej animácie. Dokážeme zdôrazniť konzistentnosť zobrazenia a zachovať zameranie na konkrétnu oblasť údajov bez akéhokoľvek približovania alebo posúvania.

Funkcia *shadow*

Do údajových bodov alebo iných prvkov v grafe môžeme pridať tieň, aby sme vytvorili vizuálny efekt, ktorý simuluje hĺbku a rozmer. Pridanie tieňov do animovaných vizualizácií `ggplot2` môže zvýšiť ich vizuálnu príťažlivosť. Aj keď samotný `gganimate` neposkytuje vstavanú funkciu na pridávanie tieňov, tento efekt môže dosiahnuť prispôbením grafu a animácie `ggplot2`.

1. `shadow_wake()`: vytvorenie efektu prebudenia si možno predstaviť ako simuláciu stopy alebo cesty, ktorú za sebou zanechal pohybujúci sa objekt počas animácie. Tento efekt sa často používa na zdôraznenie pohybu a dráhy objektu, ako je dátový bod alebo tvar, v priebehu času.
2. `shadow_trail()`: sa využíva na vytvorenie efektu „stopy“ pridaním vizuálnej stopy alebo cesty za pohybujúcim sa objektom v animácii. Tento efekt je užitočný najmä vtedy, keď potrebujeme zdôrazniť trajektóriu alebo cestu objektu, napríklad dátového bodu, keď sa pohybuje v animácii.
3. `shadow_mark()`: sa používa na pridanie tieňa alebo značky v každom časovom bode počas animácie. Tento tieň alebo značka môže predstavovať históriu alebo cestu dátového bodu, zdôrazňujúc jeho pohyb a trajektóriu v priebehu času.

Funkcia *enter/exit*

Danú funkcie môžeme použiť na ovládanie animácií vstupu a výstupu údajových bodov v animovanom grafe `ggplot2`. Tieto možnosti určujú, ako sa údajové body zobrazujú

a miznú pri prechode do animácie a von z nej. To je užitočné pri vytváraní hladkých a vizuálne príťažlivých animácií.

1. `enter_manual()` a `exit_manual()`: slúžia na manuálne určenie spôsobu, akým dátové body vstupujú a opúšťajú animáciu. Funkcia umožňuje mať jemnú kontrolu nad výskytom a zmiznutím údajových bodov v konkrétnych stavoch animovaného grafu.
2. `enter_appear()` a `exit_disappear()`: sa používajú na ovládanie toho, ako dátové body vstupujú a opúšťajú animáciu tak, že sa budú objavovať a miznúť. Tieto funkcie umožňujú určiť, ako dátové body prechádzajú do grafu a von z neho v konkrétnych stavoch, čím vytvárajú vizuálny efekt, keď sa zdá, že sa objavujú a miznú.
3. `enter_fade()` a `exit_fade()`: zobrazujú ako dátové body vstupujú do animácie a opúšťajú ju, a to tak, že sa postupne stmievajú. Tieto funkcie umožňujú určiť, ako dátové body prechádzajú do grafu a von z neho zmenou ich nepriehľadnosti.
4. `enter_grow()` a `exit_shrink()`: ide o vyobrazenie toho, ako dátové body vstupujú a opúšťajú animáciu tak, že sa ich veľkosť zväčší, keď sa objavia, a zmenšia sa, keď miznú. Tieto funkcie umožňujú určiť, ako dátové body prechádzajú do grafu a von z neho zmenou ich veľkosti.
5. `enter_recolour()` a `exit_recolour()`: slúžia na ovládanie toho, ako dátové body vstupujú a opúšťajú animáciu, a to zmenou ich farby, keď sa objavujú a miznú. Tieto funkcie umožňujú určiť, akým spôsobom budú dátové body prechádzať do grafu a von z neho zmenou ich farieb.

Medzi ďalšie funkcie patrí: `enter_fly()` a `exit_fly()`, `enter_drift()` a `exit_drift()`, `enter_reset()` a `exit_reset()`.

Funkcia *easy_aes*

Umožňuje určiť, ako by sa mala estetika animovať počas prechodu medzi rôznymi stavmi animácie. Funkcia `easy_aes()` poskytuje kontrolu nad uvoľnením estetiky, ktorá určuje, ako sa mení v priebehu času. Uvoľnenie definuje rýchlosť a vzor prechodu. Objekty v skutočnom živote sa nielen okamžite spúšťajú a nezastavujú ale takmer nikdy sa nepohybujú konštantnou rýchlosťou.

Funkcia `easy_aes()` modeluje:

- funkciu druhej mocniny (quadratic) - pomocou funkcie `easy_aes()` môžeme do grafov ľahko pridávať estetické prvky charakteristické pre kvadratické vzory, ako sú paraboly.
- funkciu tretej mocniny (cubic) - funkcia `easy_aes()` modeluje funkciu tretej mocniny, čo znamená, že vytvára model, ktorý vyjadruje závislosť medzi premennými tak, že hodnota vysvetľovanej premennej (závislá premenná) je kubickou funkciou hodnoty vysvetľujúcej premennej (nezávislá premenná).
- funkciu štvrtej mocniny (quartic) - funkcia `easy_aes()` modeluje funkciu štvrtej mocniny, čo znamená, že výsledná hodnota (y) pre danú hodnotu (x) bude vypočítaná pomocou umocňovania hodnoty (x) na štvrtú mocninu.
- $\pi/2$ kruhový oblúk (circular) - funkcia `easy_aes()` modeluje $\pi/2$ kruhový oblúk v priestore súradníc, ktorý sa nazýva "circular".
- exponenciálnu funkciu (exponential) - v kontexte modelovania exponenciálnej funkcie znamená, že `easy_aes()` je schopná nastaviť estetické vlastnosti grafu tak, aby sa vizuálne zobrazila exponenciálna závislosť medzi premennými. To môže zahŕňať nastavenie farby, tvaru alebo veľkosti bodov alebo čiar tak, aby vizuálne odrážali vzor exponenciálneho rastu alebo poklesu v údajoch.
- elastické uvoľňovanie energie (elastic) - znamená, že tento nástroj sa používa na nastavenie estetických vlastností (ako farby, veľkosti, tvary atď.). Pojem "elastické uvoľňovanie energie" môže byť metaforou, ktorá opisuje flexibilitu alebo dynamiku, ktorú táto funkcia poskytuje pri manipulácii s grafickými vlastnosťami.
- odskakovanie lopty (bounce) - pokiaľ ide o "odskakovanie lopty", to znamená, že lopta sa v animácii odrazí od povrchu (napríklad od zeme) a potom stúpne naspäť. Informácie boli spracované podľa zdrojov [19], [30].

3.10.2 Inštalácia balíka *R ganimate*

Inštalácia



Balík `gganimate` nainštalujeme pomocou nižšie uvedeného kódu :

```
install.packages("gganimate")
```

Tento balík používa `gganimate` na konverziu série obrázkov na animácie GIF, ktoré sa zobrazia na obrazovke. Potom ho môžeme jednoducho načítať s nasledujúcim kódom:

```
library(gganimate)
```

Alebo môžeme začiarknuť políčko pred názvom balíka na paneli balíkov RStudio:

	Name	Description	Version		
<input checked="" type="checkbox"/>	gganimate	A Grammar of Animated Graphics	1.0.9		

Obrázok 15: Zobrazenie balíka *gganimate* v paneli balíkov RStudio, zdroj: vlastné spracovanie

3.10.3 Vykresľovanie a výstup objektu R *gganimate*

Vykresľovanie objektu R *gganimate*

Pomocou funkcie `animate()` dokážeme pohodlne vykresliť a prezerat' zhuk objektu bez toho, aby sme ho uložili ako súbor alebo sa spoliehali na externé prehliadače. To uľahčuje experimentovanie s rôznymi animáciami a prechodmi pri práci v R.

Funkcia `print()` sa zvyčajne nepoužíva na interaktívne prezeranie animácií v jazyku R, častejšie sa používa na zobrazenie statických vizualizácií.

Ďalšou možnosťou ako dokážeme vykresliť objekt `gganimate` je pomocou funkcie `knit_print.gganimate`, zvyčajne to urobíme v dokumente, ktorý podporuje vykresľovanie, ako je napríklad dokument R Markdown. Tým sa do výstupného dokumentu vloží veľká animácia a môžeme si prispôsobiť formát dokumentu podľa potreby (napr. HTML, PDF alebo iné formáty podporované R Markdown). [23]

Export animácie

Balík `gganimate` poskytuje niekoľko možností vykresľovania, ktoré sa používajú na export a uloženie animácií v rôznych formátoch. Tieto možnosti vykresľovania umožňujú vybrať najlepší formát pre špecifické potreby.

1. `gifski_renderer()`: sa používa na export animácií vo formáte GIF. Je to populárna voľba na vytváranie animovaných obrázkov GIF. Je možné prispôsobiť rôzne parametre, ako je názov súboru, kvalita a rozmery GIF.
2. `av_renderer()`: sa používa na export animácií vo video formátoch, ako sú MP4, WebM a GIF. Pri kódovaní sa spolieha na balík `av` a poskytuje dobrú kvalitu videa a menšie veľkosti súborov v porovnaní s GIF.

3. `ffmpeg_renderer()`: exportuje animácie vo video formátoch pomocou softvéru `ffmpeg`. Ponúka širokú škálu možností formátu videa a umožňuje presnú kontrolu nad kvalitou a nastaveniami videa.
4. `magick_renderer()`: sa používa na export animácií v rôznych formátoch, vrátane GIF, MP4, WebM a ďalších. Na vykresľovanie využíva balík `magick` a poskytuje možnosti na ovládanie rôznych aspektov procesu vykresľovania.
5. `rsvg_renderer()`: sa používa na export animácií ako sekvencie súborov SVG (Scalable Vector Graphics). Aj keď je to pre animácie menej bežné, môže to byť užitočné pri vytváraní vektorových animácií.

Výstup objektu *gganimate*

V `gganimate` používame funkciu `anim_save()` na určenie výstupného formátu a uloženie animovaných grafov do súboru. Funkcia umožňuje ukladať animácie vo formáte GIF, MP4, WebM a ďalších.

Funkcia `last_animation()` sa používa na prístup k poslednej vykreslenej animácii. Po vykreslení animácie pomocou funkcie `animate()` môžeme použiť `last_animation()` na odkazovanie na naposledy vykreslenú animáciu, ktorá umožní vykonávať ďalšie akcie alebo ju uložiť v rôznych formátoch.

Funkcia `frame_vars()` umožňuje pristupovať a pracovať s premennými spojenými s jednotlivými snímkami animácie. Táto funkcia sa používa na prezeranie a manipuláciu s údajmi špecifickými pre rám.

`split_animation()` je funkcia v balíku `gganimate`, ktorá umožňuje rozdeliť animáciu do viacerých podanimácií na základe premennej zoskupenia. To je užitočné, keď potrebujeme vytvoriť samostatné animácie pre rôzne skupiny údajov. [29], [30]

Tvorba grafu *ggplot2* a animácie

Balík `ggplot2` je účinný pre tvorbu statických grafov, avšak nie je schopný spracovať animácie, ktoré sú užitočné pri zobrazení vývoja údajov indexovaných podľa času. Na túto úlohu slúži balík `gganimate`, ktorý je navrhnutý na podobnom princípe, ale pre animácie. Vytvára grafy pre každé časové obdobie a potom ich interpoluje, aby vytvoril plynulú animáciu.

V predchádzajúcom kroku sme si ukázali ako nainštalovať a spustiť balík `gganimate`, syntax a prispôbenie, ako aj vykresľovanie animácie. Predpokladajme, že sme si

importovali dáta a teraz z nich chceme spraviť animáciu. Pred tým než vytvoríme animáciu je potrebné vytvoriť statický graf pomocou balíčka `ggplot2`. Tento základný graf predstavuje počiatočný stav našej animácie a zostáva konštantný počas celej animácie.

Balík `gganimate` umožňuje vytvárať animácie pomocou `ggplot2`. Animácia je v podstate postupnosť statických obrázkov, ktoré sa zobrazujú postupne. V ďalšom kroku pridáme prechodové vrstvy a animácie pomocou funkcie `transition_states()`, ktorá vytvára sériu grafov. V tejto funkcii špecifikujeme premennú `XY` ako argument stavov, čo hovorí R, že táto premenná určuje sériu grafov (stavy tu nie sú o geografickej podjednotke, ale o časovej alebo znakovej podjednotke - premýšľame o rôznych stavoch hmoty).

Zároveň môžeme zadať, ako dlho chceme, aby bol každý stav viditeľný pomocou argumentu `state_length()` a ako dlho chceme, aby každý prechod medzi stavmi trval pomocou argumentu `transition_length()`. Pridáme aj funkciu `shadow_mark()`, ktorá udrží body z predchádzajúcich stavov viditeľné a pridá viac bodov navrch. Po vytvorení zavoláme objekt animácie z konzoly, aby sme ho videli v okne Viewer v RStudio.

Pre zachytenie a zopakovanie relevantných rozdielov sme zostavili tabuľku, ktorá komparuje balík `ggplot2` a balík `gganimate`. [19], [30], [42]

ggplot2	gganimate
Je určený na tvorbu statických grafov.	Umožňuje vytvárať animácie pomocou <i>ggplot2</i> .
Neposkytuje priamu podporu pre animácie.	Poskytuje prostredie na vytváranie animácií z <i>ggplot2</i> .
Grafy sú statické a nezmenia sa s časom.	Grafy môžu byť dynamické a meniť sa s časom.
Je účinný na zobrazenie statických dát.	Je vhodný na zobrazenie pohyblivých dát a časových radov.
Vytvára statické vizualizácie údajov.	Vytvára dynamické vizualizácie údajov.
Neposkytuje priamu funkcionálnosť pre animácie.	Poskytuje nástroje na tvorbu animácií z <i>ggplot2</i> grafu.

Tabuľka 1: Komparácia balíka *ggplot2* a *gganimate*, zdroj: vlastné spracovanie

3.11 R Markdown

Tento nástroj bol navrhnutý na zjednodušenie reprodukovateľnosti, pretože umožňuje integrovať výpočtový kód a textové popisy v jednom dokumente a výsledky sa automaticky generujú zo zdrojového kódu. R Markdown podporuje množstvo statických aj dynamických/interaktívnych výstupných formátov.

Dokument R Markdown sa skladá z troch hlavných častí: metadát, textu a kódu. Metadáta sú umiestnené medzi trojitými pomlčkami a používajú syntax YAML, preto sú niekedy označované ako YAML metadáta. Dôležité je poznamenať, že v YAML formáte záleží na odsadení, preto je potrebné správne odsadiť podpolia v hornom poli. Telo dokumentu nasleduje za metaúdajmi.

Existujú dva typy počítačového kódu:

- Časť kódu začína trojitými spätnými apostrofmi, napríklad ``` { r }`, pričom `r` označuje jazyk. Táto časť kódu končí rovnakým spôsobom tromi spätnými apostrofmi ````. Medzi týmito spätnými apostrofmi je možné definovať možnosti blokov (napríklad nastavenie výšky obrázka na 5 palcov: ``` { r, obr.výška=5 }`).
- Vložený R kód začína znakom `r` a končí spätnými apostrofmi ````. [13]

R Markdown je ľahký značkovací jazyk so syntaxou formátovania obyčajného textu, ktorý je možné previesť do HTML a mnohých ďalších formátov. Kombinácia R Markdown a R kódu umožňuje vytvárať reprodukovateľný výskum, ktorý dokáže automaticky aktualizovať výsledky, čísla a štatistiky priamo v dokumente. Tu sú typy výstupov, ktoré je možné vygenerovať z dokumentu R Markdown:

1. **HTML dokument:** Toto je jeden z najbežnejších výstupov. Vytvára dokument podobný webovej stránke, ktorý možno jednoducho zdieľať a prezerať v akomkoľvek webovom prehliadači.
2. **Dokument PDF:** R Markdown dokáže vytvárať dokumenty PDF konverziou Markdown na LaTeX, systém na prípravu dokumentov. To si vyžaduje, aby bola na našom systéme nainštalovaná distribúcia LaTeXu.
3. **Dokument Word:** Je možné tiež generovať dokumenty Microsoft Word priamo z R Markdown, čo je užitočné na zdieľanie zistení so spolupracovníkmi, ktorí uprednostňujú alebo vyžadujú dokumenty Word.
4. **Prezentačné snímky:** R Markdown podporuje vytváranie prezentačných snímok, ktoré môžu obsahovať naratívny text a aj výstupy R kódu. Dá sa to dosiahnuť integráciou s rámcami, ako je Beamer (pre snímky PDF), ioslides, Slidy a ďalšie pre snímky založené na HTML.
5. **Dashboardy:** Pomocou balíkov ako `flexdashboard` je možné vytvárať interaktívne dashboardy priamo z dokumentov R Markdown. Tieto ovládacie panely môžu obsahovať používateľské ovládacie prvky a dynamicky aktualizované tabuľky a grafy.

6. **Knihy a dlhé dokumenty:** Balíky ako `bookdown` rozširujú R Markdown na podporu tvorby kníh a dlhších správ, čo umožňuje komplexné štruktúry dokumentov vrátane kapitol, príloh a bibliografií.
7. **Webové stránky a blogy:** Pomocou rámcov, ako je `blogdown` a `distill`, je možné vytvárať celé webové stránky alebo blogy, kde je každý príspevok alebo stránka napísaná v R Markdown. [39]

3.11.1 R Markdown vo formáte PowerPoint prezentácie

Jednou z významných funkcií R Markdown je použitie na vytváranie prezentácií. Väčšina výsledných výstupov sú html dokumenty, ale existujú aj možnosti vytvorenia pptx alebo pdf verzie.

Vytváranie prezentácií v PowerPointe pomocou R Markdown ponúka niekoľko výhod, vďaka čomu je cenným nástrojom pre profesionálov, dátových vedcov, výskumníkov a pedagógov. Dokumenty R Markdown obsahujú kód a údaje spolu s obsahom prezentácie. Podporuje to reprodukovateľnosť, pretože niekto iný môže znova vytvoriť našu analýzu a prezentácie jednoduchým spustením dokumentu R Markdown. Ďalšou z výhod použitia PowerPoint prezentácie je, že môžeme jednoducho zahrnúť R chunk code a jeho výstup do samotných snímok bez potreby kopírovania a prilepovania informácií. To umožňuje aktualizácie grafov, tabuliek a iných vizualizácií v reálnom čase pri zmenách základných údajov alebo kódu. [45]

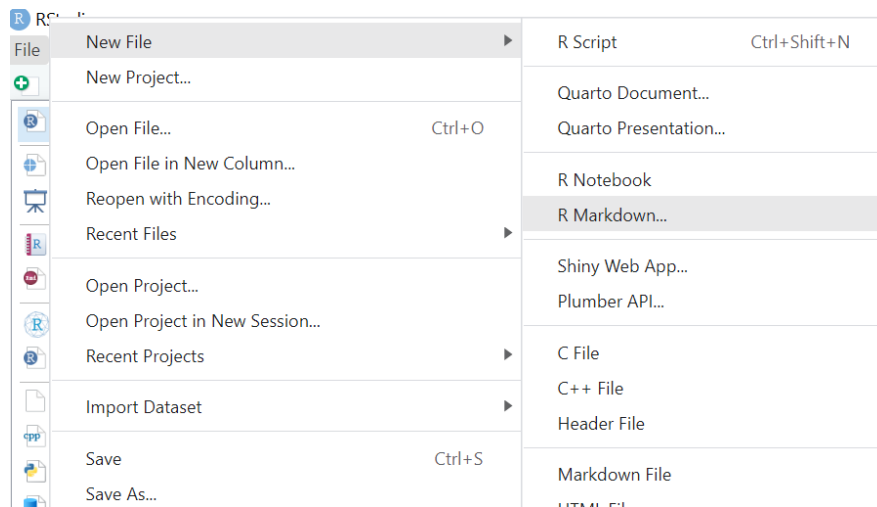
Celkovo R Markdown pre PowerPoint prezentácie ponúka výkonný a flexibilný prístup k vytváraniu dátovo orientovaných, interaktívnych a reprodukovateľných prezentácií. Jeho všestrannosť a bezproblémová integrácia s R z neho robí preferovanú voľbu pre profesionálov, ktorí potrebujú efektívne komunikovať dáta a analýzy.

Tieto dôvody robia z R Markdown vo formáte PowerPoint prezentácií atraktívnu voľbu pre tvorcov prezentácií, najmä pre tých, ktorí pracujú s dátami a kódom. Práve preto sme sa rozhodli v praktickej časti generovať výsledné výstupy ako PowerPoint prezentácie.

3.11.2 R Markdown dokument a jeho štruktúra

R Markdown je kódovanie v štýle notebooku, čo znamená, že spája prostredie podobné spracovaniu textu s prístupom k programovaciemu jazyku. Jeho cieľom je umožniť používateľovi spojiť príbeh a kúsky kódu, aby vytvoril dobre naformátovaný vedecký výstup. R

Markdown je plne integrovaný do RStudio. V ponuke Súbor na obrázku č. 16 vidíme, že máme prístup k novým R skriptom. O niečo nižšie je výber pre R Markdown.



Obrázok 16: Vytvorenie R Markdown dokumentu v RStudio, zdroj: vlastné spracovanie

Keď si vyberieme R Markdown, dostaneme súbor kde si môžeme všimnúť, že použitie # v dokumente R Markdown označuje úrovne nadpisov, ktoré budú automaticky naformátované, keď exportujeme R Markdown do iného formátu. Zvyčajne sa dokumenty R Markdown exportujú do HTML, Word alebo pdf. My v našej práci upriamime pozornosť tvorbe výstupu ako PowerPoint prezentácie. Formátovanie dokumentu R Markdown je čiastočne riadené syntaxou R Markdown a čiastočne nastaveniami v hornej časti dokumentu, známymi ako YAML, ktoré obsahujú metadáta. Metadáta sa nachádzajú medzi trojitými pomlčkami "---" a zahŕňajú základné informácie o dokumente, vid' obrázok č. 17.

```
---  
title: "r_markdown"  
output: powerpoint_presentation  
date: "2023-10-28"  
---
```

Obrázok 17: Základné informácie R Markdown súboru, zdroj: vlastné spracovanie

Tieto metadáta sa používajú pri zostavovaní dokumentu. Na rozdiel od skriptu by sme mohli začať dokument R Markdown nejakým textom o našej štúdií alebo zdroji súboru údajov, ktorý sa chystáme preskúmať. Keď chceme zahrnúť kód, vkladáme ho do tzv. kódového bloku (chunk code), kde môžeme písať kód spolu s jeho vlastnými parametrami a ovládacími prvkami. Na pridanie nového kódového bloku do R Markdown môžeme použiť klá-

vesovú skratku Option + Command + i (v IOS) alebo ctrl + shift + i vo Windows. Alternatívne môžeme využiť rozbaľovaciu ponuku Vložiť v paneli nástrojov R Markdown. [13], [24]

3.11.3 Code Chunks

Ak chceme zahrnúť kód R do dokumentov prezentácie, umiestnime ich do tzv. kódového bloku (code chunk). Syntax code chunk sa líši v závislosti od značkovacieho jazyka, ktorý používame na písanie našich dokumentov. V súboroch R Markdown začíname časť kódu napísaním hlavičky: ````{r}`. Časť kódu je uzavretá jednoducho s: `````.

Každý chunk má štítok. Keď code chunk vytvorí graf alebo sa výstup uloží do vyrovnávacej pamäte pre budúce použitie, knitr použije označenie časti pre názov nového súboru. Ak kódovému bloku nepriradíme explicitne štítok, tak mu bude automaticky priradený jeden ako: `unnamed-chunk-1`. Ak chceme explicitne priradiť štítok kódu v dokumentoch R Markdown, stačí umiestniť názov štítka do zátvoriek za r, teda ````{r ChunkLabel}`.

Časti kódu majú tiež analogické koncepty ako triedy a jedinečné identifikátory. Chunk code môže špecifikovať voliteľný identifikátor (často nazývaný „označenie časti“) hneď za jeho jazykovým modulom. Môže nastaviť triedy pre kódové a textové výstupné bloky vo výstupnom dokumente prostredníctvom možností `chunk.class.source` a `class.output`. Napríklad hlavička kódového bloku (`r` súhrnné štatistiky, `class.output = 'large-text'`) dáva kódovému bloku súhrnnú štatistiku menovky a veľký text triedy pre bloky textového výstupu. Chunk môže mať len jednu menovku, ale môže mať viacero tried. [45]

3.11.4 Štruktúra prezentácie

Prezentácie programu PowerPoint majú hierarchiu titulkov, sekcií a snímok obsahu, zatiaľ čo R Markdown dokumenty majú hierarchiu hlavičiek a obsahu. Aby bolo možné vytvoriť štruktúru prezentácie v PowerPointe, Pandoc musí určiť, ktorá úroveň hlavičky označenia by sa mala použiť pre úroveň snímky PowerPoint. Napríklad hlavička označenia `## Introduction` sa môže namapovať na snímku sekcie, zatiaľ čo `### Details` sa môže namapovať na snímku s obsahom.

V predvolenom nastavení, Pandoc nastaví úroveň snímky na najvyššiu úroveň hlavičky v hierarchii, po ktorej bezprostredne nasleduje obsah označenia niekde v dokumente.

Keď Pandoc určí úroveň snímky, dokument označenia sa rozreže na snímky podľa nasledujúcich pravidiel:

- Hlavičky na úrovni snímky vždy začínajú novú snímku.
- Hlavičky nad úrovňou snímky v hierarchii vytvárajú hlavičky sekcií.
- Hlavičky pod úrovňou snímky v hierarchii vytvárajú hlavičky na snímke.
- Vodorovné pravidlo vždy začína novú snímku.
- Titulná strana sa vytvorí automaticky z bloku názvu dokumentu (t. j. YAML).

```
---
title: Title slide (Slide 1)
output:
  powerpoint_presentation:
    slide_level: 3 # use this to override default
---

# Section header (Slide 2)

## Section header (Slide 3)

### Slide Title (Slide 4)

Slide Content (by default this content sets the slide level to three)

#### Slide content header (Slide 4)

##### Slide content header (Slide 4)
```

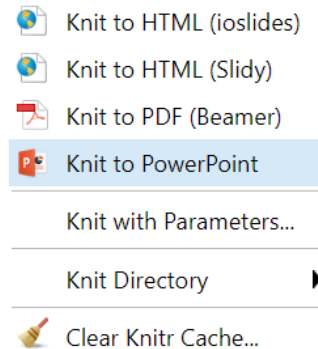
Obrázok 18: Vizualizácia tvorby prezentácie so štyrmi snímkami, zdroj:[36]

Vyššie uvedený obrázok č. 18 zobrazuje príklad dokumentu, ktorý vytvorí prezentáciu so štyrmi snímkami. Predvolená úroveň snímky je 3, pretože hlavička úrovne 3 je prvá, za ktorou nasleduje obsah. Predvolenú úroveň snímky môžeme prepísať nastavením možnosti `slide_level` v hlavičke YAML. Ak nechceme svoje snímky štruktúrovať do sekcií, môžeme na vytvorenie všetkých snímok použiť iba hlavičky úrovne 1. [36]

3.11.5 Pandoc

Pandoc umožňuje konvertovať súbory z jedného formátu značkovacieho jazyka, ako je R Markdown, do iného. Týmto spôsobom môžeme vložiť HTML kód do súboru `rmarkdown`. Ak vytvoríme súbor HTML, Pandoc preloží HTML kód rovnako, ako to robí prehliadač pri navštívení webovej stránky. S pomocou Pandocu môžeme konvertovať súbor do formátov ako HTML, Word a ďalších. Keď klikneme na šípku vedľa tlačidla "Knit" a vyberie-

me možnosť vytvorenia PowerPoint prezentácie, RStudio vygeneruje dokument a pridá kód na vytvorenie súboru PowerPoint.



Obrázok 19: Ukážka tvorby súboru PowerPoint v RStudio, zdroj: vlastné spracovanie

RStudio má v sebe zabudovaný Pandoc, takže ak používame RStudio, nemusíme inštalovať Pandoc samostatne, inak je potrebné si nainštalovať Pandoc samostatne. V prípade, že máme samostatnú inštaláciu Pandoc, RStudio ho použije iba vtedy, ak je naša verzia vyššia ako verzia Pandoc v RStudio. [13]

3.12 R Shiny

Shiny je balík R, ktorý umožňuje používateľom vytvárať interaktívne webové aplikácie a panely s R kódom. Balík poskytuje rámec na vytváranie **používateľských rozhraní (UI)** pomocou funkcií R a umožňuje používateľom písať logiku na strane **servera**, aby reagovala na vstup používateľa a aktualizovala používateľské rozhranie v reálnom čase. Základná štruktúra aplikácie Shiny pozostáva z dvoch hlavných komponentov: používateľského rozhrania a logiky na strane servera.

Užívateľské rozhranie je vytvorené prostredníctvom funkcií v jazyku R, ktoré definujú štruktúru a usporiadanie webovej stránky, vrátane vstupných a výstupných prvkov. Logika na strane servera, napísaná v R kóde, sa vykonáva na serveri a reaguje na vstup od používateľa. Po interakcii používateľa so stránkou sa jeho vstup odosiela na server, kde sa spracuje a dynamicky aktualizuje užívateľské rozhranie v reálnom čase. [33]

Shiny prvky môžu byť integrované do dokumentov R Markdown spolu s ďalším textom a R kódom, čo nám umožňuje, aby naše analýzy boli interaktívne, ak je to potrebné.

V skutočnosti existuje veľa typov aplikácií Shiny a mnoho spôsobov, ako ich vytvoriť. Sú to nasledovné:

- Interaktívne R Markdown dokumenty so zabudovanými miniaplikáciami Shiny.

- Shiny aplikácie (predvolené CSS, napísané celé v jazyku R).
- Webové stránky (napríklad vlastné CSS, HTML a JavaScript).
- Shiny gadgety.
- Flex dashboardy. [14]

3.12.1 Štruktúra UI

Táto časť bude pokrývať rôzne časti rozhrania s príslušnými funkciami. V skratke sa pozrieme na štruktúru, ktorá bude rozdelená na: Pages, Layouts, Panels, Rows, Columns.

Shiny poskytuje niekoľko šablón pre webové aplikácie. Sú to všeobecné štruktúry na organizovanie rôznych prvkov v rámci rozhrania, ako je poloha vstupov, poloha výstupov atď. Mnohé z prvkov súvisiacich so štruktúrou sú rozdelené na tekuté a pevné. Hlavný rozdiel spočíva v tom, že tekuté štruktúry automaticky prispôbujú šírku svojich prvkov šírke prehliadača, zatiaľ čo pevné štruktúry majú šírku definovanú používateľom.

V zásade sa Shiny prvky usporadúvajú v rámci štruktúry po riadkoch, čo znamená, že sú umiestnené jeden pod druhým. Napríklad, funkcia `structure(elementA, elementB)` zobrazí prvok B pod prvkom A vnútri štruktúry. Hoci sa to vôbec neodporúča, počet prvkov, ktoré je možné umiestniť pod seba, je neobmedzený. Existujú však niektoré funkcie, ktoré usporiadajú svoje prvky podľa stĺpcov, teda jeden vedľa druhého (napríklad `sidebarLayout()`). [9]

Page (Stránka): tvorcovia tzv. builders stránky definujú typ stránky, na ktorej bude aplikácia postavená. V tomto zmysle sú prvými zahrnutými tvorcami štruktúr, pretože fungujú ako kontajnery pre zvyšok štruktúr a widgetov. Z technického hľadiska vytvárajú štruktúru HTML dokumentu, na ktorom bude aplikácia postavená. Nasleduje zoznam tvorcov stránok:

`fluidPage()`: toto je najjednoduchšia hlavná štruktúra, funkcia na vysokej úrovni, ktorá vytvára zobrazenie prispôbené šírke prehliadača. Spolu so `sidebarLayout()` to je najjednoduchší spôsob, ako vytvoriť Shiny používateľské rozhranie. V skutočnosti kombinácia `fluidPage()` a `sidebarLayout()` nahrádza `pageWithsidebar()`, ktorá je síce zastaraná, ale stále je prítomná v mnohých príkladoch na webe.

`fixedPage()`: na rozdiel od `fluidPage()`, vytvára aplikáciu s pevnou veľkosťou, nezávislú od šírky prehliadača. Ak nie je explicitne požadované, aby prvky v HTML mali konkrétnu veľkosť, vždy sa odporúča použiť `fluidPage()`.

`bootstrapPage()`: toto je najzákladnejší typ stránky, kde sa načíta iba JavaScript bootstrap knižnica.

`navbarPage()`: vytvorí stránku s navigačnou lištou umiestnenou na vrchu. Často sa kombinuje s funkciou `tabpanel()`.

Layouts (*Rozloženie*): tvorcovia rozloženia generujú prázdne štruktúry s určitými vlastnosťami, do ktorých budú prvky umiestnené neskôr. [16]

`sidebarLayout()`: vytvorí stránku s dvoma hlavnými sekciami, bočným a hlavným panelom. Štandardne majú šírku 4 a 8 pixelov. Každý z nich má svoje vlastné funkcie vytvárania: `sidebarPanel()` pre bočný panel a `mainPanel()` pre hlavný panel. Vstupy sa umiestňujú do bočného panela a výstupy do hlavného panela. Aj keď je táto štruktúra odporúčaná, je len vizuálnou konvenciou, takže neexistuje žiadne pevné pravidlo, ktoré by obmedzovalo umiestnenie vstupov alebo výstupov do jedného z nich.

Panels (*Panely*): panely sú ohraničené oblasti, ktoré slúžia na konkrétny účel. Normálne majú panely buď vstupné alebo výstupné prvky:

`inputPanel()`: tento panel slúži na zahrnutie vstupov do jedného bloku. Funkcia umiestni rôzne vstupy zľava doprava a zhora nadol.

`tabsetPanel()`: zobrazuje prvky v záložkách. Obzvlášť užitočné je zobrazit' rôzne výstupy a zoskupit' ich do záložiek podľa tém.

`absolutePanel()`: toto je panel so špecifikovanou veľkosťou a pozíciou. Parametre pozície sa budú brať tak, ako sa vezmú v absolútnej pozícii CSS. `FixedPanel()` je ekvivalentný `absolutePanel()` s jediným rozdielom, že používa pevnú pozíciu CSS.

`conditionalPanel()`: toto je panel, ktorý je viditeľný iba vtedy, ak je splnená určitá podmienka. Môže sa použiť na obsiahnutie vstupných alebo výstupných prvkov.

`tabpanel()`: toto je štruktúra záložiek, ktorú možno použiť vo vnútri `navbarPage()`, `tabsetPanel()` alebo `navlistPanel()` na vytvorenie záložiek v štruktúre, ktorá ich obsahuje.

Rows, Columns (*Riadky, stĺpce*): sú to štruktúry, ktoré obsahujú prvky riadok alebo stĺpec.

`fluidRow()` a `fixedRow()`: Tieto funkcie riadkov sú obaly, ktoré usporiadajú prvky tak, aby boli rovnakej výšky. V skutočnosti sa štruktúra používateľského rozhrania aplikácie Shiny väčšinou skladá z riadkov so stĺpcami vo vnútri, najmä pre najzákladnejšie vstavované funkcie. [16], [41]

3.12.2 Vstupy UI

Balík Shiny poskytuje širokú škálu vstupných miniaplikácií, ktoré je možné zahrnúť do aplikácie.

Free inputs (Voľné vstupy): Voľné vstupy možno rozdeliť hlavne do dvoch veľkých skupín: číselné vstupy a znakové vstupy. Sú nasledovné:

`numericInput()`: je voľné pole, do ktorého môže používateľ zadať číslo. Hoci Shiny umožňuje nastavenie maximálnych a minimálnych hodnôt tohto ovládača, je dôležité si uvedomiť, že v predvolenom nastavení miniaplikácia označí neplatné vstupy červenou farbou, ale aj tak sa spustí s poslednou zadanou hodnotou.

`textInput()`: je voľný text, do ktorého môže používateľ písať. Žiaľ, stále nie je žiadna zabudovaná funkcia na overenie formátu vstupného textu, ako je tomu napríklad u `numericInput()`, ktorý kontroluje, či je zadaná hodnota číselná. Na overenie špecifických formátov, ako napríklad e-mailová adresa, je potrebné implementovať dodatočnú validáciu.

`sliderInput()`: umožňuje používateľovi vybrať číselnú hodnotu posúvaním posuvníka. Rozdielom oproti voľnému textovému poľu je, že hodnota sa zvolí posúvaním škály, nie manuálnym písaním. Ďalšou výhodou `sliderInput()` je, že je obmedzený na určitý rozsah hodnôt, a preto používateľ nemôže zadať hodnoty mimo tohto rozsahu.

Lists (Zoznamy): Vo väčšine prípadov sa zoznamy používajú na filtrovanie možností. Konceptne sa dajú rozdeliť do dvoch veľkých skupín: jedinečné a viacnásobné selektory. Hoci pre niektoré widgety zoznamu je zmena z jedného typu na druhý taká jednoduchá ako nastavenie viacerých argumentov na TRUE alebo FALSE, je dôležité mať na pamäti, že objekt vráti zmeny zo znaku na znakový vektor.

`selectinput()`: je výber z rozbaľovacej ponuky. V rámci volieb argumentov možno zoznam volieb odovzdať ako znakový vektor, kde zobrazené hodnoty a hodnoty uložené v zodpovedajúcich vstupných objektoch budú rovnaké, alebo ako znakový vektor s priradenými názvami. V tomto prípade sa mená zobrazia v aplikácii, ale hodnota bude uložená v premennej.

`checkboxGroupInput()`: zobrazí zoznam začiarovacích políčok s jedným začiarovacím políčkom na každú prejdenú možnosť. Vytvorený objekt v zozname vstupov je ekvivalentný objektu vytvorenému pomocou `selectInput()` s nastavením `multiple=T`.

`checkboxInput()`: zobrazí jedno začiarkavacie políčko. Jeho výsledok je TRUE alebo FALSE.

Dates (Dátumy): Shiny poskytuje dve vstavané funkcie na výber dátumov. Obe majú veľmi intuitívne rozhranie na výber dátumu. Vrátene objekty sú dátumové objekty. V pokročilejších implementáciách je však možné, že v niektorých prípadoch sa objekty dátumov nespávajú podľa očakávania.

`dateInput()`: slúži na výber dátumu. Minimálnou a maximálnou hodnotou môže byť reťazec vo formáte yyyy-mm-dd alebo objekt dátumu.

`dateRangeInput()`: slúži na výber rozsahu dátumov. Minimálna a maximálna hodnota môže byť definovaná ako reťazec alebo dátum. Technicky je to ekvivalent dvojnásobku `dateInput()`, s jediným rozdielom, že funkcia nepovoľuje koncové hodnoty menšie ako počiatočné. Výsledný objekt je pole s dvomi prvkami, pričom prvým prvkom je začiatkový dátum a druhým je koncový dátum.

Files (Súbory): možnosť umožniť používateľom nahrávať vlastné súbory sa dá zahrnúť pomocou `fileInput()`. Tento prvok vytvára klasické rozhranie pre výber súborov, ktoré obsahuje tlačidlo, ktoré po kliknutí otvorí okno prehliadača. Ako s týmto objektom manipulovať, závisí od účelu aplikácie, či už ide o potrebu transformácie na dátový rámec, na zoznam riadkov a podobne.

Buttons (Tlačidlá):

`actionButton()`: predstavuje tlačidlo, ktoré vyvoláva určitú akciu. Často sa využíva na zabránenie automatického opakovaného spustenia kódu pri každej zmene vstupu. Táto funkcionálnosť je často využívaná v súvislosti s izoláciou.

`submitButton()`: oneskorí vykonanie akejkoľvek zmeny v každej vstupnej hodnote, kým sa nestlačí tlačidlo. Hoci to môže byť veľmi podobné `actionButton()`, existuje zásadný rozdiel: `submitButton()` mení všetky vstupné hodnoty súčasne, zatiaľ čo `actionButton()` umožňuje spúšťať akcie odlišne po stlačení tlačidla. [28], [35]

3.12.3 Serverová logika R Shiny

Súbor `server.R` obsahuje logiku na strane servera a spúšťa sa na serveri, ktorý hostí našu aplikáciu Shiny. Rovnako ako používateľské rozhranie, ktoré je pôvodne definované funkciou `shinyUI()`, je server pôvodne definovaný funkciou `shinyServer()`.

Vytvoríme prázdny súbor s názvom `server.R` a pridáme funkciu `shinyServer()`:

```
shinyServer(function(input, output, session){...})
```

Väčšinu logiky znázornenú elipsami v predchádzajúcom kóde v rámci tejto funkcie zakódujeme. Vstupné a výstupné parametre by mali byť vždy zahrnuté, pretože vstupný parameter sťahuje všetky vstupné akcie používateľa zo strany používateľského rozhrania a výstupný parameter posúva všetky údaje na strane servera a zobrazuje ich späť do používateľského rozhrania.

Tretí parameter, relácia, sa nevyžaduje, ale v určitých prípadoch sa odporúča, pretože nám umožňuje definovať špecifické premenné pre každú reláciu používateľa. Aplikáciu Shiny môže používať viacero osôb súčasne a vytvorením premenných špecifických pre reláciu môže byť táto skúsenosť jedinečná pre každého používateľa, pretože môže s aplikáciou súčasne interagovať. [33], [41]

4 Výsledky práce a diskusia

V tejto diplomovej práci sme sa v praktickej časti zaoberali využitím údajov pochádzajúcich zo Svetovej zdravotníckej organizácie (WHO), pričom sme sa špecificky venovali analýze údajov o úmrtiach v dôsledku dopravných nehôd. Ďalším zdrojom údajov, ktorý sme použili v praktickej časti, sú údaje pochádzajúce z Organizácie pre hospodársku spoluprácu a rozvoj (OECD) týkajúce sa počtu obetí dopravných nehôd. Posledným zdrojom údajov, ktorý sme sa rozhodli zahrnúť do praktickej časti, sú štatistické údaje z Eurostatu týkajúce sa úmrtnosti súvisiacej so srdcovými ochoreniami a úmrtnosti súvisiacej s rakovinou. Okrem týchto údajov sme tiež využili vlastné simulované údaje, čo nám umožnilo doplniť existujúce štatistiky a poskytnúť hlbší uhol pohľadu na problémy a možné riešenia.

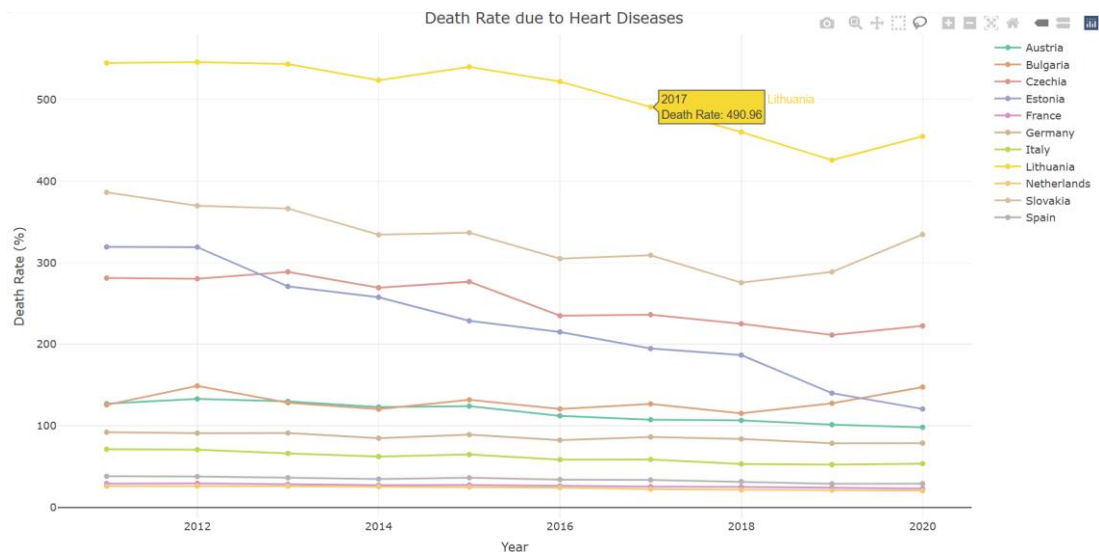
V nasledujúcich výsledkoch sme použili knižnice, ktoré nám otvorili unikátne možnosti ako prezentovať, spracovať a analyzovať naše údaje. Tu je krátky prehľad použitých knižníc:

- `library(plotly)` – knižnica pre tvorbu interaktívnych grafov;
- `library(dplyr)` – knižnica pre efektívnu prácu s dátami, ich transformáciu;
- `library(tidyverse)` – knižnica zameraná na tvorbu dátových analytických väzieb;
- `library(bayesplot)` – knižnica pre vizualizáciu bayesiánskych modelov;
- `library(rstanarm)` – knižnica pre pokročilé štatistické modelovanie;
- `library(webshot)` – knižnica pre programové vytváranie snímok zo Shiny aplikácií;
- `library(ggplot2)` – knižnica pre tvorbu štatistických grafov;
- `library(gganimate)` – knižnica pre vizualizovanie zmien v dátach alebo štatistikách v čase;
- `library(tweenr)` – knižnica poskytuje nástroje pre jednoduchú animáciu prechodov;
- `library(shiny)` – knižnica pre tvorbu interaktívnych webových aplikácií priamo v R;
- `library(shinydashboard)` – knižnica pre tvorbu dashboardov v Shiny;
- `library(DT)` – knižnica pre integráciu interaktívnych tabuliek;
- `library(forecast)` – poskytuje funkcie na predpovedanie údajov časových radov.

4.1 Praktická ukážka R *plotly*

4.1.1 Vizualizácia údajov úmrtnosti na srdcové choroby

Na obrázku č. 20 vidíme interaktívny graf, ktorý vizualizuje mieru úmrtnosti na srdcové choroby v rôznych krajinách v priebehu času. Jeho cieľom je ponúknuť pohľad na trendy, porovnania a špecifické dátové body prostredníctvom pútavého a informatívneho vizuálneho média.



Obrázok 20: Čiarový graf,
zdroj: vlastné spracovanie

Vykreslením miery úmrtnosti v dôsledku srdcových chorôb v rôznych rokoch (os x) nám umožnil graf sledovať trendy v čase v každej krajine. Uvedením miery úmrtnosti v percentuálnom formáte na osi y kvantitatívne informuje divákov o závažnosti a rozsahu úmrtnosti na srdcové choroby. v kontexte populácie. Použitie rôznych farieb pre každú krajinu uľahčuje priame porovnanie úmrtnosti na srdcové choroby medzi krajinami.

Symbol `%>%` je *pipe* operátor z `dplyr`, ktorý odovzdáva výstup jednej funkcie ďalšej, vďaka čomu je kód čitateľnejší. Funkcia `plot_ly()` sa používa na spustenie interaktívneho grafu `plotly` v dátovom rámci `ischemic_deaths`. Každý argument špecifikuje rôzne vlastnosti grafu, ako napríklad farbu podľa krajiny, text pri umiestnení kurzora myši atď.

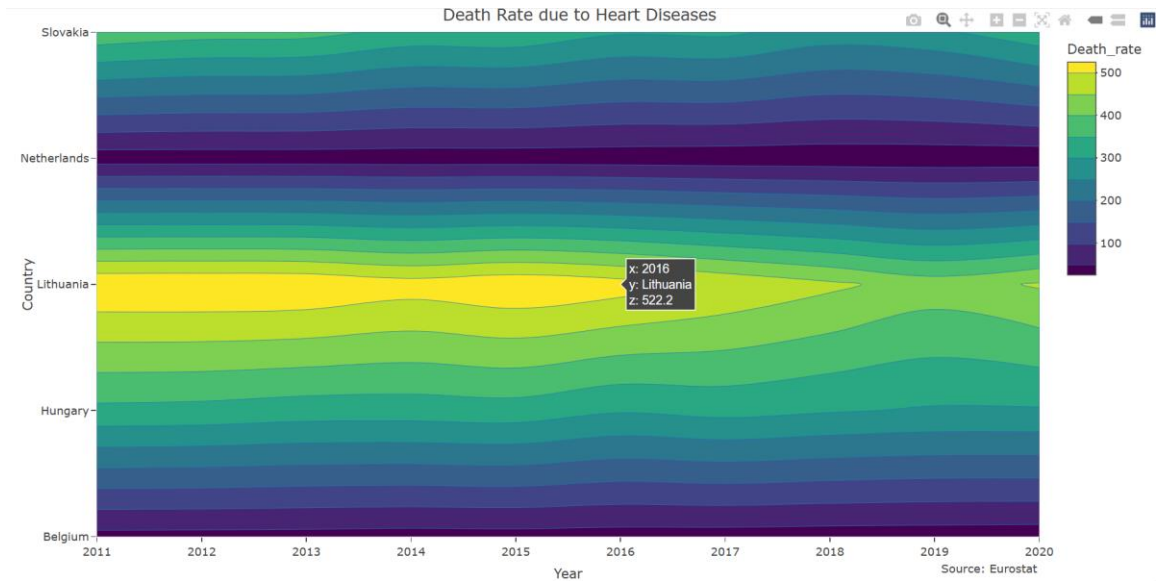
```

interactive_plot <- ischaemic_deaths %>%
  plot_ly(
    x = ~Year,
    y = ~Death_rate,
    color = ~Country,
    text = ~paste("Death Rate:", Death_rate),
    hoverinfo = "text + x + name",
    type = 'scatter',
    mode = 'lines + markers'
  ) %>%
  layout(
    title = "Death Rate due to Heart Diseases",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Death Rate (%)"),
    hovermode = 'closest'
  )

```

Prostredníctvom argumentov `text` a `hoverinfo`, keď používateľ umiestni kurzor myši v ľubovoľnom údajovom bode sa na grafe zobrazí popis s presnou úmrtnosťou, rokom a krajinou. Nakoniec, funkcia `layout()` upravuje vzhľad grafu s nadpismi osí, názvom a režimom vznášania, čím vytvára prehľadný a esteticky príjemný graf. Zreťazením týchto funkcií a použitím `%>%` sa vytvára interaktívny graf. Kompletný kód uvádza **príloha č. 1**.

V praktickej časti výskumu sme sa rozhodli použiť ako ďalší graf - obrysový graf na obrázku č. 21 na vizualizáciu a analýzu úmrtnosti na srdcové choroby v priebehu rokov v rôznych krajinách. Voľba obrysového grafu vo vizualizácii `plotly` je pre tento výskumný kontext obzvlášť výhodná. Obrysové grafy sú účinné pri zobrazovaní toho, ako sa miera úmrtnosti vyvíja v priebehu času (reprezentovaná rokmi) v rôznych krajinách. Tento typ grafu nám umožňuje jasnejšie pozorovať trendy, vzorce a potenciálne korelácie medzi rokom, krajinou a mierou úmrtnosti.

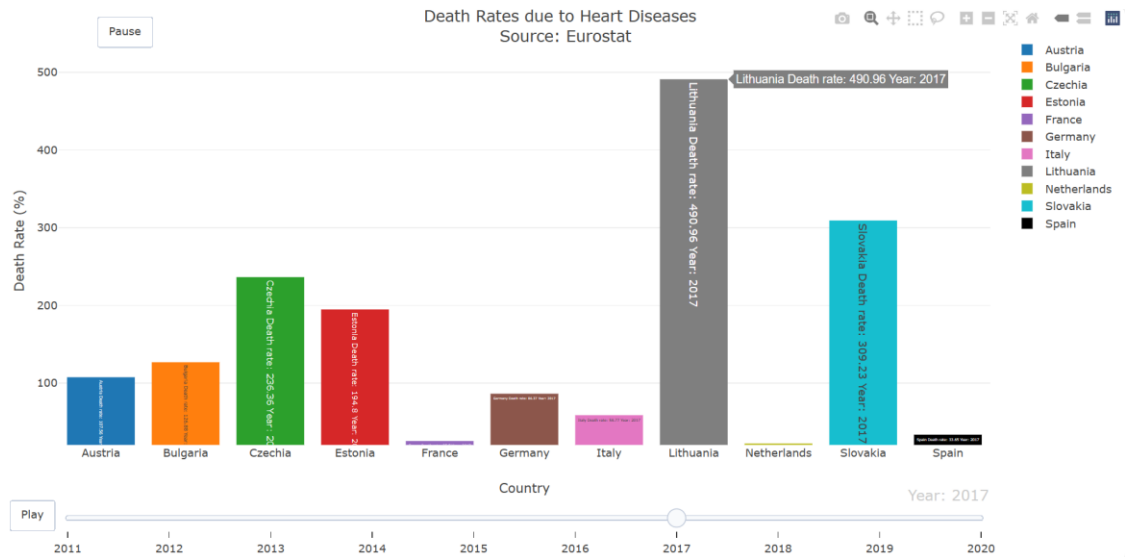


Obrázok 21: Obrysový graf,
zdroj: vlastné spracovanie

Os x predstavuje rok, os y predstavuje krajinu a os z predstavuje úmrtnosť. Graf obsahuje anotáciu, ktorá poskytuje dodatočné informácie. Anotácia špecifikuje zdroj údajov ako "Zdroj: Eurostat" a jeho umiestnenie na grafe je upravené pomocou atribútov x , y , $xref$, $yref$, $xanchor$ a $yanchor$. Kompletný kód uvádza **príloha č. 2**.

```
fig <- fig %>% layout(
  title = "Death Rate due to Heart Diseases",
  xaxis = list(title = 'Year'),
  yaxis = list(title = 'Country'),
  annotations = list(
    list(
      x = 1,
      y = -0.08,
      xref = "paper",
      yref = "paper",
      text = "Source: Eurostat",
      showarrow = FALSE,
      xanchor = "right",
      yanchor = "bottom"
    )
  )
)
```

Na obrázku č. 22 vidíme graf, ktorý obsahuje aj funkciu pre animáciu, ktorá používateľom umožňuje hrať sa v priebehu rokov a sledovať, ako sa dynamicky mení úmrtnosť. Táto interaktívna vizualizácia môže pomôcť pri identifikácii vzorov, odľahlých hodnôt a trendov v množine údajov efektívnejšie v porovnaní so statickými grafmi.



Obrázok 22: Stĺpcový graf s tlačidlami a posúvačom, zdroj: vlastné spracovanie

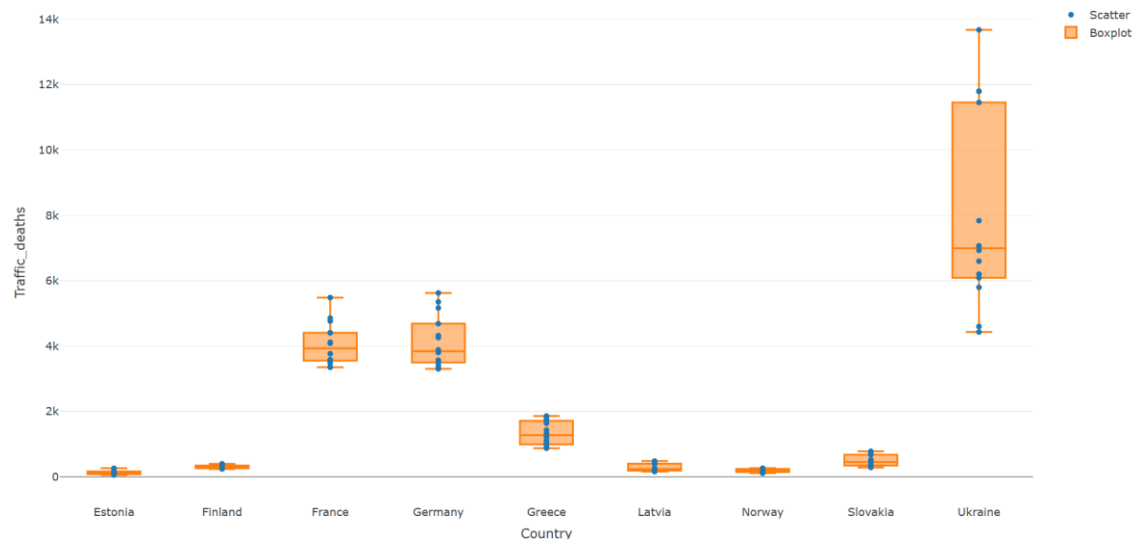
Veľkosť značiek v grafe je upravená na základe hodnôt úmrtnosti. Väčšie značky predstavujú vyššiu mieru úmrtnosti, zatiaľ čo menšie značky predstavujú nižšiu mieru úmrtnosti. Graf zobrazuje informatívny text pri umiestnení kurzora myši nad údajové body, zobrazujúci podrobnosti, ako je krajina, úmrtnosť a rok. Táto funkcia umožňuje skúmať konkrétne dátové body a získavať prehľad bez preplnenia vizualizácie.

Os x grafu predstavuje krajinu zo súboru údajov, os y predstavuje úmrtnosť. Údaje sú rozdelené podľa krajiny. Snímky animácie sú určené rokom. Veľkosť markerov je určená škálovaním hodnôt miery úmrtnosti vzhľadom na maximálnu mieru úmrtnosti v množine údajov. Súčasťou kódu je tlačidlo na ovládanie animácie. Tlačidlo sa nachádza na pozícii ($x = 0, 1, y = 1,05$) s kotvou nastavenou na „vpravo“. Možnosti animácie sú špecifikované pomocou funkcie `animation_opts()`. Animácia je nastavená tak, aby mala celkovo 600 snímok. Prekreslenie grafu pri každej zmene rámca zaisťuje parameter `redraw`. Kompletný kód uvádza **príloha č. 3**.

4.1.2 Vizualizácia údajov úmrtnosti pri dopravných nehodách

Bodový graf (*scatter plot*), ktorý sme vytvorili, ukazuje vzťah medzi dopravnými úmrtiami a rôznymi krajinami. Každý bod v bodovom grafe predstavuje krajinu, pričom na osi *x* sú krajiny a na osi *y* počet dopravných úmrtí.

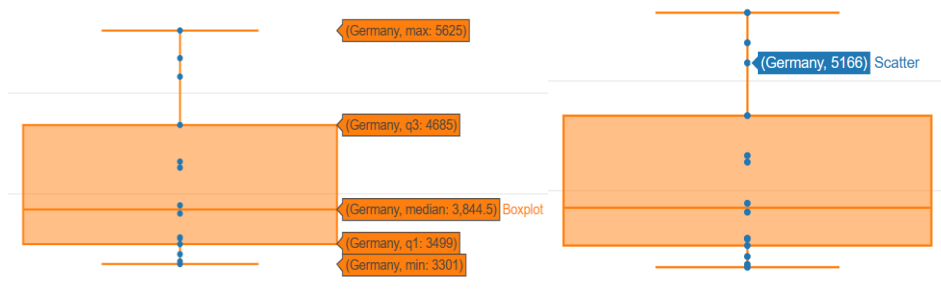
Vizualizáciou údajov v bodovom grafe môžeme rýchlo identifikovať akékoľvek vzory, trendy alebo odľahlé hodnoty v miere úmrtnosti na cestách v rôznych krajinách. Tento graf nám môže pomôcť pochopiť, ktoré krajiny majú vyšší alebo nižší počet dopravných úmrtí, identifikovať akékoľvek potenciálne korelácie medzi premennými a porovnať počet úmrtí pri dopravných nehodách medzi rôznymi krajinami.



Obrázok 23: Kombinovaný graf (box plot a scatter plot), zdroj: vlastné spracovanie

Okrem toho, krabicový graf (*boxplot*), ktorý ste vytvorili, dopĺňa bodový graf poskytnutím súhrnu distribúcie dopravných nehôd v rámci každej krajiny. Zobrazuje medián, kvartily a akékoľvek odľahlé hodnoty v údajoch, čo umožňuje komplexnejšie pochopenie variability v miere úmrtnosti v dopravných nehodách v jednotlivých krajinách.

Bodový graf zobrazuje jednotlivé dátové body, zatiaľ čo krabicový graf poskytuje súhrn distribúcie dopravných úmrtí v jednotlivých krajinách.



Obrázok 24: Box plot vs. Scatter plot,
zdroj: vlastné spracovanie

Vytvorenie bodového grafu

Kód vytvoril bodový graf pomocou funkcie `plot_ly()` z knižnice `plotly`. Špecifikovali sme údaje na osi x ako krajinu a údaje na osi y ako počet úmrtí v tisícoch. Nastavili sme parametre rozloženia pre graf, vrátane nadpisu a rozsahu osi x a osi y .

```
fig <- traffic_data %>%
  plot_ly(x = ~Country, y = ~Traffic_deaths) %>%
  layout(
    xaxis = list(
      title = "Country",
      range = c(10, 35)
    ),
    yaxis = list(
      title = "Traffic deaths",
      range = c(50, 500)
    )
  )
```

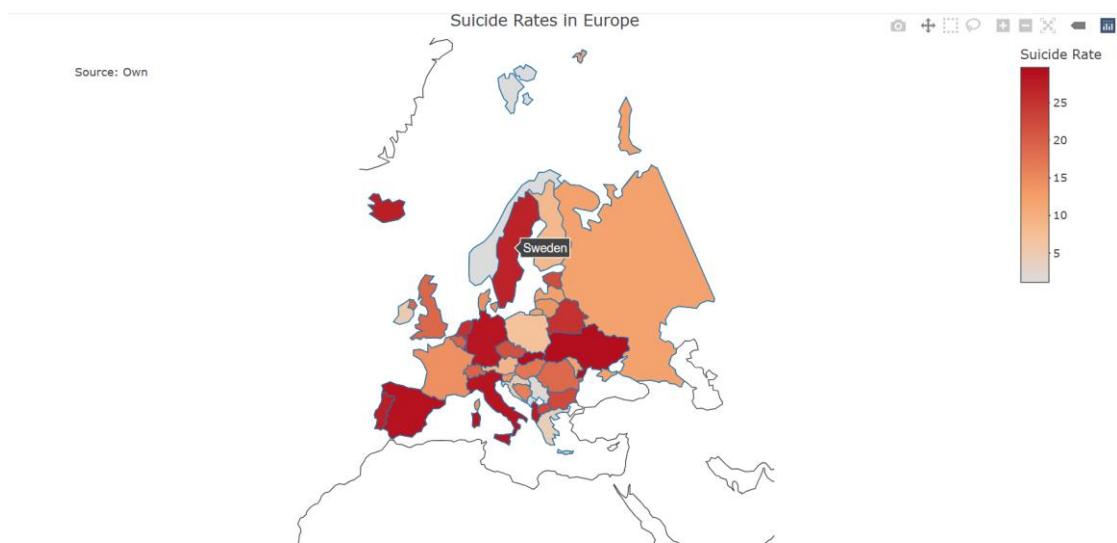
Vytvorenie krabicového grafu

Kód potom prepíše premennú `fig` vytvorením nového grafu pomocou funkcie `plot_ly()`. Tentoraz nastaví typ grafu ako bodový graf (`type = "scatter"`) na vytvorenie krabicového grafu. Pridá k bodovému grafu krabicový graf. Kompletný kód uvádza **príloha č. 4**.

```
fig <- traffic_data %>%
  plot_ly(x = ~Country, y = ~Traffic_deaths, type = "scatter",
          name = "Scatter") %>%
  add_boxplot(name = "Boxplot")
```

4.1.3 Vizualizácia nasimulovaných dát

Obrázok č. 25 vizualizuje mieru samovrážd v rôznych európskych krajinách. Každá krajina je na mape znázornená pomocou farebného gradientu, pričom tmavšie odtiene naznačujú vyššiu mieru samovrážd. Táto vizualizácia umožňuje ľahko porovnávať a identifikovať krajiny s relatívne vysokou alebo nízkou mierou samovrážd. Použitím choropletovej mapy je jasné, kde existujú koncentrácie vysokých alebo nízkych mier, čo umožňuje ďalšiu analýzu a interpretáciu faktorov, ktoré môžu prispieť k týmto mieram v rôznych krajinách. Pomáha porovnávať mieru samovrážd geograficky v rôznych regiónoch Európy a analyzovať potenciálne korelácie medzi mierou samovrážd a socioekonomickými faktormi.



Obrázok 25: Mapa Európy,
zdroj: vlastné spracovanie

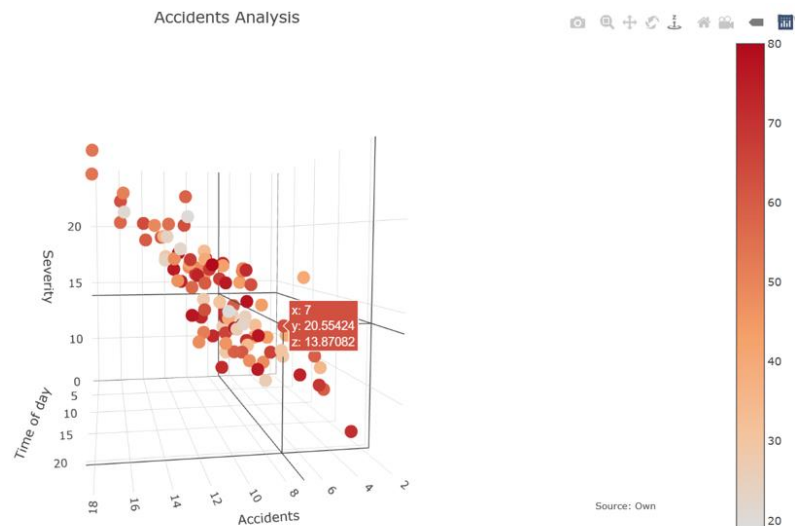
Najprv sme si nasimulovali údaje o miere samovrážd. Funkcia `set.seed()` sa používa na nastavenie zdroja pre reprodukovateľnosť pri generovaní náhodných čísel. Vytvorili sme si vektor krajiny obsahujúci názvy európskych krajín. Následne sme vygenerovali náhodné miery samovrážd. Funkcia `runif()` generuje náhodné čísla medzi 1 a 30 vrátane pre každú krajinu, čím simuluje mieru samovrážd.

```
suicide_rates <- runif(length(countries), min = 1, max = 30)
```

Vytvorili sme si dátový rámec s názvom „data“ s dvoma stĺpcami: „Country“ obsahujúci názvy krajín a „Suicide_Rate“ obsahujúci simulované sadzby. Na prispôbenie mapy

sme si definovali parametre `locations`, `z`, `colorscale`, `text`, `colorbar` a `hover-info`. Kompletný kód uvádza **príloha č. 5**.

Na obrázku č. 26 je 3D bodový graf integrujúci niekoľko premenných na vizualizáciu zložitých vzťahov v rámci údajov o dopravných nehodách. Graf zobrazuje údaje o simulovaných dopravných nehodách, pričom zahŕňa viaceré dimenzie informácií: počet nehôd, dennú dobu, kedy k nim došlo, ich závažnosť a vek zúčastnených vodičov.



Obrázok 26: 3D scatter graf,
zdroj: vlastné spracovanie

Os x zobrazuje počet nehôd, ktoré boli vygenerované z Poissonovho rozdelenia s priemerom („lambda“) 10. To naznačuje, že nehody sú jednotlivé udalosti, ktoré sa stanú konštantnou priemernou rýchlosťou nezávisle od času od poslednej udalosti.

```
accidents <- rpois(100, lambda = 10)
```

Os y predstavuje čas dňa, kedy došlo k nehode, v rozsahu od 0 do 24. Údaje sú v tomto rozsahu rovnomerne rozdelené, čo znamená, že k nehodám môže dôjsť s rovnakou pravdepodobnosťou kedykoľvek čas v rámci 24 hodín.

```
time_of_day <- runif(100, min = 0, max = 24)
```

Os z , na tejto osi je vynesená závažnosť nehôd. Odvodzuje sa pridaním Gaussovho hluku (s priemerom 5 a štandardnou odchýlkou 2) k počtu nehôd, čo naznačuje variabilitu závažnosti každej nehody, pravdepodobne zodpovedajúcu faktorom, ako sú rýchlosť, poveternostné podmienky alebo poškodenie vozidla.

```
severity <- accidents + rnorm(100, mean = 5, sd = 2)
```

Vek vodičov zapojených do nehôd zafarbí značky. Farebná škála sa pohybuje od svetlej farby po tmavú farbu, čo naznačuje, že mladší vodiči môžu byť zastúpení svetlejšími farbami a starší vodiči tmavšími farbami, hoci presné mapovanie závisí od veku údajov. Farebná škála je viditeľná na pozemku, čo nám umožnilo pochopiť vekové rozdelenie vodičov.

Argumenty *x*, *y* a *z* sa používajú na určenie, ktoré stĺpce dátového rámca sa majú vykresliť na ktorých osiach. Argument *marker* mení farbu bodov na základe veku vodiča so špecifikovanou farebnou škálou. Argument *showscale* spôsobí, že farebná stupnica bude viditeľná na grafe. Kompletný kód uvádza **príloha č. 6**.

```
fig <- plot_ly(data = accidents_data, x = ~accidents, y =  
~time_of_day, z = ~severity,  
              marker = list(color = ~driver_age, colorscale =  
c('#FFE1A1', '#683531'), showscale = TRUE))
```

3D bodový graf predstavuje viacero premenných a ponúka komplexný pohľad na dáta. Tento holistický prístup môže pomôcť identifikovať vzory alebo korelácie, ktoré nemusia byť zjavné pri izolovanej analýze premenných. Vykreslením nehôd v závislosti od dennej doby a ich závažnosti a následného farebného označenia podľa veku vodiča je ľahšie pozorovať potenciálne korelácie. Napríklad by sa dalo zistiť, že nehody vyššej závažnosti sa zvyčajne vyskytujú v určitých časoch dňa alebo sú viac spojené s určitými vekovými skupinami. Na základe nastavenia simulácie existuje navrhnutá pozitívna korelácia medzi počtom nehôd a ich závažnosťou.

Jasné anotácie a popisný názov pomáhajú divákovi pochopiť, vďaka čomu nie je zápletká len zbierkou údajov, ale aj súvislým príbehom o dopravných nehodách.

Stručne povedané, 3D bodový graf, ako je opísaný, je výkonným nástrojom na počiatočnú prieskumnú analýzu údajov, najmä pri práci s viacrozmernými súbormi údajov. Umožňuje lepšie porozumieť interakciám medzi rôznymi premennými, čo je kľúčové pre komplexnú analýzu údajov a rozhodovacie procesy.

Obrázok č. 27 poskytuje pohľad na Bayesovský Poissonov regresný model prispôbený simulovaným aktuárskym údajom. [46], [48] Vysvetlenie čo nám hovorí každý graf:

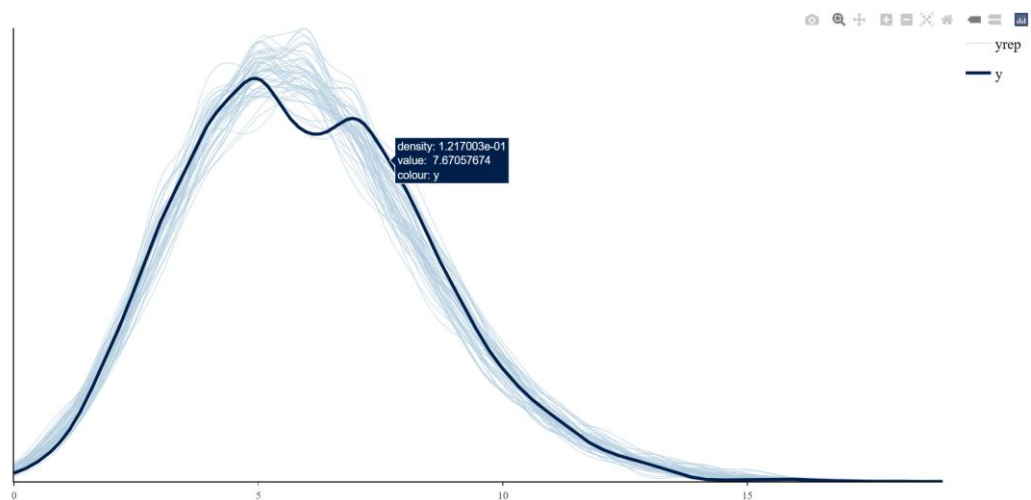
1. Graf hustoty aposteriórneho rozdelenia:

Tento graf zobrazuje aposteriórne rozdelenia parametrov modelu - v tomto prípade priesečník a koeficient pre vek. Vytieňované oblasti predstavujú 80 % dôveryhodných inter-

valov pre tieto parametre. Skúmaním týchto rozdelení je možné posúdiť neistotu v odhadoch parametrov modelu.

2. PPC graf (*Posterior Predictive Check Plot*):

Aposteriórny prediktívny kontrolný graf porovnáva pozorované údaje o škodách s predpokladanými hodnotami získanými z prispôsobeného modelu. Prekrytím týchto rozdelení sme mohli vizuálne skontrolovať, ako dobre model zachytáva pozorované údaje. V tomto prípade sa 50 žrebovaní prediktívnych hodnôt vykreslí oproti skutočným údajom o škodách. [48]



Obrázok 27: Bayesovský Poissonov regresný model, zdroj: vlastné spracovanie

V kontexte grafu PPC generovaného v kóde hrajú premenné y a y_{rep} odlišné úlohy:

1. Pozorované údaje y :

V grafe PPC sa y vzťahuje na pozorované údaje, konkrétne na skutočné hodnoty škôd zo simulovaných poisťno-matematických údajov. Toto sú údaje, ktoré sme zhromaždili alebo pozorovali v skutočnosti.

2. Replikované údaje y_{rep} :

Na druhej strane, y_{rep} predstavuje replikované alebo predpovedané údaje. V grafe PPC sa vzťahuje na simulované hodnoty škôd generovaných modelom na porovnanie s pozorovanými údajmi. Tieto replikované hodnoty sa generujú na základe posteriorných vzoriek z Bayesovského modelu, čo poskytuje spôsob, ako posúdiť, ako dobre model zodpovedá pozorovaným údajom.

Kódom, ktorý sme uviedli nižšie, sme si nainštalovali balík `bayesplot` priamo z GitHubu pomocou `devtools`. Voľba `dependencies = TRUE` zaisťuje, že sa nainštalujú aj všetky balíky, na ktorých závisí, a `build_vignettes = FALSE` preskočí vytváranie `vignets`², aby sa inštalácia urýchlila.

```
devtools::install_github("stan-dev/bayesplot", dependencies =
TRUE, build_vignettes = FALSE)
```

Nasimulovali sme si údaje, kde sa vek generuje ako Poissonova distribučná premenná s `lambda` 40, ku ktorej sme pridali 20, čím sa simuloval vek jednotlivcov v našom súbore údajov od 20 rokov vyššie.

Škody sme tiež nasimulovali ako Poissonovu distribučnú premennú, pričom miera (`lambda`) je funkciou veku/10, čo naznačuje, že škody rastú s vekom, ale nie lineárne.

```
age <- rpois(n, lambda = 40) + 20
claims <- rpois(n, lambda = age/10)
```

Pomocou funkcie `stan_glm()` z balíka `rstanarm` sme prispôbili údajom o škodách Poissonov regresný model s vekom ako prediktorom. Toto modeluje údaje o počte škôd vo vzťahu k veku za predpokladu, že údaje sledujú Poissonovo rozdelenie.

```
fit <- stan_glm(claims ~ age, data = actuarial_data, family =
"poisson")
```

Funkcia `as.matrix(fit)` konvertovala model prispôbeného objektu na maticu aposteriorných vzoriek. Funkcia `mcmc_areas()` z balíka `bayesplot` vytvorila plošné grafy aposteriorných rozdelení pre prienik a vekový koeficient, pričom ukázala hustotu aposteriorných vzoriek a zvýraznila mediány a 80% dôveryhodné intervaly. Účelom bolo vizuálne skontrolovať aposteriórne rozdelenie parametrov nášho modelu a poskytnúť pohľad na ich neistotu po pozorovaní údajov. [46]

```
posterior <- as.matrix(fit)
```

Funkciou `ppc_dens_overlay()` sme vytvorili graf prekrývajúci hustotu pozorovaných údajov (škôd) s hustotami z aposteriorného prediktívneho rozdelenia. Tento krok

² V kontexte inštalácie balíkov R sa `vignets` vzťahuje na inštruktážny dokument alebo príručku, ktorá poskytuje podrobné informácie o funkciách balíka.

je rozhodujúci pre kontrolu toho, ako dobré sú predpovede modelu v súlade so skutočnými údajmi. Výraz `posterior_predict(fit, draws = 50)` generuje predpovede z a-posteriórneho rozdelenia parametrov na základe 50 ťahov. Toto sa porovnáva so skutočnými údajmi o škodách, aby sa vyhodnotila vhodnosť modelu. Kompletný kód uvádza **príloha č. 7**.

```
plot_ppc <- ppc_dens_overlay(y = actuarial_data$claims,
                             yrep = posterior_predict(fit, draws =
50))
```

Porovnaním pozorovaných údajov y s replikovanými dátami y_{rep} v PPC grafe je možné vizuálne skontrolovať, do akej miery sú predpovede modelu v súlade so skutočnými dátami. Toto porovnanie pomáha pri hodnotení prediktívneho výkonu modelu a pri hodnotení, či model primerane zachytáva vzorce a variabilitu, ktoré sú prítomné v pozorovaných údajoch.

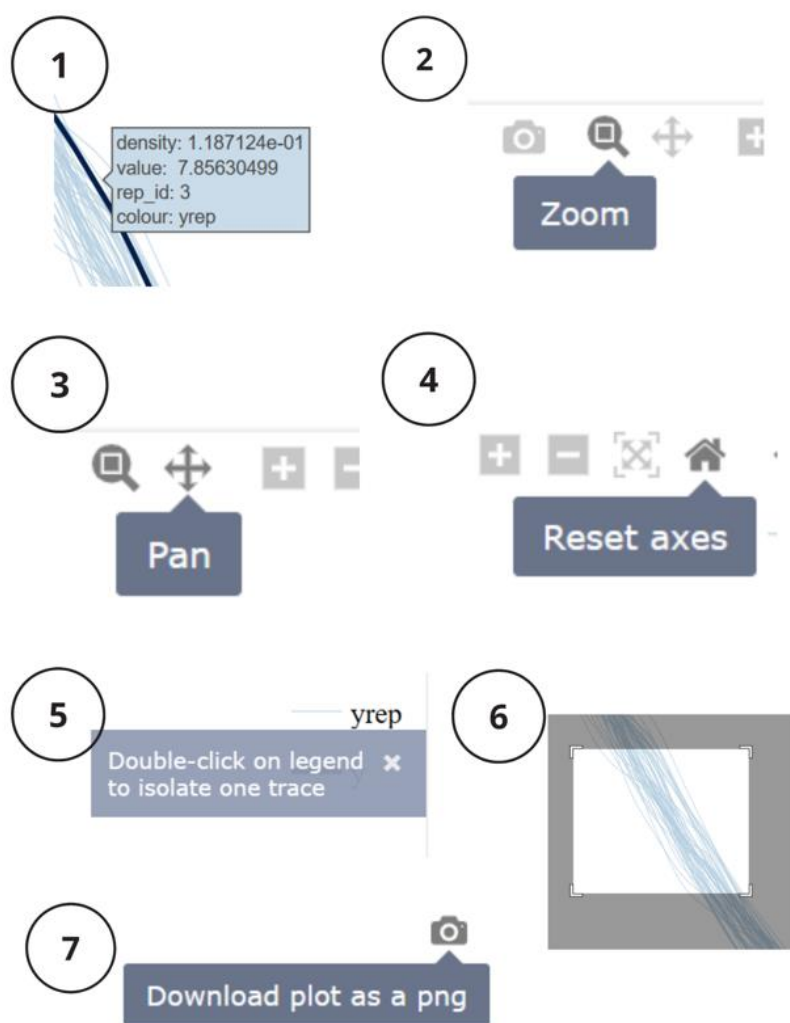
Prekrytie a-posteriórnych prediktívnych hustôt sa v našom prípade zhoduje s hustotou údajov o skutočných škodách a to naznačuje, že predpovede modelu sú v súlade s pozorovanými údajmi.

4.1.4 Interaktívna prezentácia

Interakcia s grafom vytvoreným pomocou knižnice `plotly` v R ponúka všestranný a dynamický zážitok. Po vytvorení interaktívneho grafu v prostredí R a jeho vykreslení (zvyčajne v paneli RStudio Viewer, webovom prehliadači alebo vložené do webovej aplikácie), sme s ním mohli interagovať niekoľkými spôsobmi:

1. **Vznášanie sa:** Pri presune kurzoru na rôzne časti grafu sme videli podrobné informácie o každom údajovom bode.
2. **Priblíženie:** Kliknutím a potiahnutím sme mohli vybrať konkrétnu oblasť grafu, ktorú sme chceli priblížiť, aby sme si ju pozreli lepšie.
3. **Posúvanie:** Po priblížení sme sa mohli kliknutím a potiahnutím pohybovať v rámci priblíženého zobrazenia a skúmať rôzne časti našich údajov.
4. **Resetovanie zobrazenia:** Zahŕňa tlačidlo „Home“ alebo „Reset Axes“, ktoré resetuje úroveň priblíženia a polohu posúvania na pôvodné zobrazenie.

5. **Prepnutie stôp:** V prípade, že graf zobrazuje viacero sérií (napríklad rôzne krajiny na našom ako na našom grafe), mohli sme kliknúť na položky legendy a prepnúť (zobrazit'/skryť) konkrétne série.
6. **Výber dátových bodov** (*Box Select a Lasso Select*): Niektoré konfigurácie nám umožnili vybrať dátové body v rámečku (*Box Select*) alebo nakresliť okolo nich tvar voľného tvaru (*Lasso Select*), čo bolo užitočné na zvýraznenie alebo interaktívne analyzovanie podmnožiny našich údajov.
7. **Stiahnutie grafu:** V interaktívnych ovládacích prvkoch je možnosť stiahnuť si aktuálny pohľad na graf ako statický obrázok (napr. vo formáte PNG).



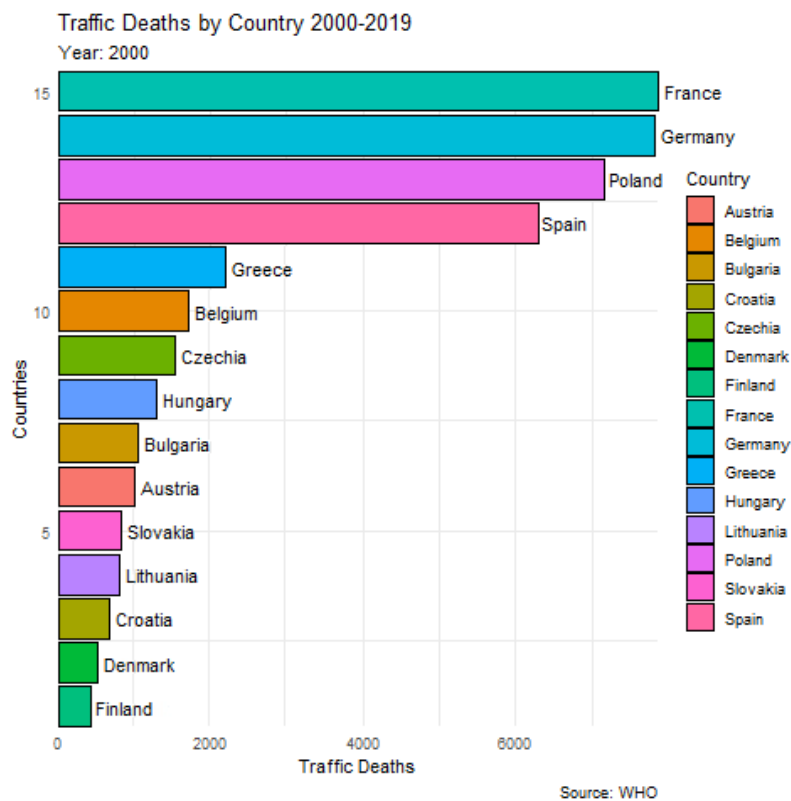
Obrázok 28: Panel nástrojov balíka *plotly* vo webovom prehliadači, zdroj: vlastné spracovanie

4.2 Praktická ukážka R *gganimate*

4.2.1 Vizualizácia údajov úmrtnosti pri dopravných nehodách

Animácia č. 1 predstavuje dynamickú reprezentáciu toho, ako sa za 20 rokov vyvíjalo poradie krajín z hľadiska počtu úmrtí pri dopravných nehodách, pričom sme poukázali na krajiny, ktoré zaznamenali významné posuny vo svojich pozíciách.

Animovaný stĺpcový graf zobrazuje kolísavé poradie krajín na základe počtu dopravných úmrtí, ktoré zažili v priebehu rokov. Každý stĺpec predstavuje krajinu a dĺžka stĺpca zodpovedá počtu dopravných úmrtí v danom roku. Ako animácia postupuje, pruhy sa pohybujú nahor a nadol, čo naznačuje zmeny v poradí krajín na základe počtu dopravných úmrtí. Podnadpis sa dynamicky mení, aby zobrazoval rok, ktorý je zastúpený v jednotlivých snímkach.



Animácia 1: Stĺpcový graf,
zdroj: vlastné spracovanie

Po načítaní potrebných knižníc sme usporiadali dátový rámec `traffic_data` v zostupnom poradí na základe počtu úmrtí pri dopravných nehodách. Zoskupili sme údaje podľa roku a priradili každej krajine poradie na základe počtu dopravných úmrtí pomocou

metódy nerozhodného výsledku. Zrušili sme zoskupenie údajov a skonvertovali stĺpec rok na celé čísla.

```
traffic_data <- traffic_data %>%
  arrange(desc(Traffic_deaths))
traffic_data <- traffic_data %>%
  group_by(Year) %>%
  mutate(ordering = rank(Traffic_deaths, ties.method = "first"))
%>%
  ungroup()
traffic_data$Year <- as.integer(traffic_data$Year)
```

Nastavili sme konštanty ako titulok, počet snímok grafu (`fpgw`), okno konca grafu (`end_gw`), obmedzenie, počet snímok za sekundu (`fps`) a časovanie ukončenia pauzy (`end_pause`).

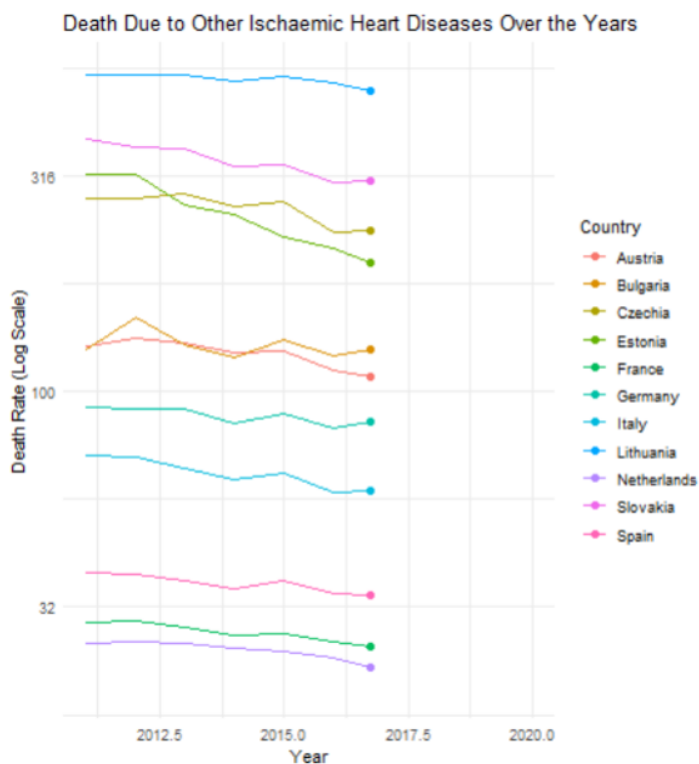
Definovali sme funkciu s názvom `make_barchart_race`, ktorá vytvára animované preteky v pruhovom grafe. Následne sme vytvorili objekt `ggplot` so špecifikovanou estetickou a vrstvami na vytvorenie stĺpcového grafu. Nakonfigurovali sme prechody a vzhľad grafu v priebehu času podľa rokov. Aplikovali sme efekty animácie, ako je vstup a výstup, ako aj nastavenie animácie tak, aby sledovala pevnú polohu osi x .

Zavolali sme funkciu `make_barchart_race` s názvom grafu a zadali počet snímok za sekundu (`fps`) a ukončíte trvanie pauzy (`end_pause`). Výslednú animovanú vizualizáciu sme uložili ako súbor GIF. Kompletný kód uvádza **príloha č. 8**.

Táto vizuálna reprezentácia môže pomôcť pri identifikácii trendov, vzorcov a anomálií v štatistikách dopravných úmrtí v rôznych krajinách v priebehu rokov.

4.2.2 Vizualizácia údajov úmrtnosti na srdcové choroby

Animácia č. 2 vizualizuje údaje pomocou čiarového grafu, ktorý ukazuje trend v úmrtnosti pre každú krajinu v priebehu času. Obsahuje body na označenie konkrétnych údajových bodov, zatiaľ čo os y je zobrazená na logaritmickú stupnici pre lepšiu vizualizáciu údajov. Vytvorili sme prechody medzi jednotlivými rokmi a poskytli dynamický pohľad na vývoj úmrtnosti na srdcové choroby v rôznych krajinách v priebehu času.



Animácia 2: Čiarový graf,
zdroj: vlastné spracovanie

Pomocou funkcie `ggplot()` sme vytvorili čiarový graf úmrtnosti proti roku, zoskupenie a zafarbenie podľa krajiny podľa krajiny. Pridali sme body, ktoré reprezentujú jednotlivé dátové body. Graf je nastavený na prechod a odhaľuje dátové body na základe roku.

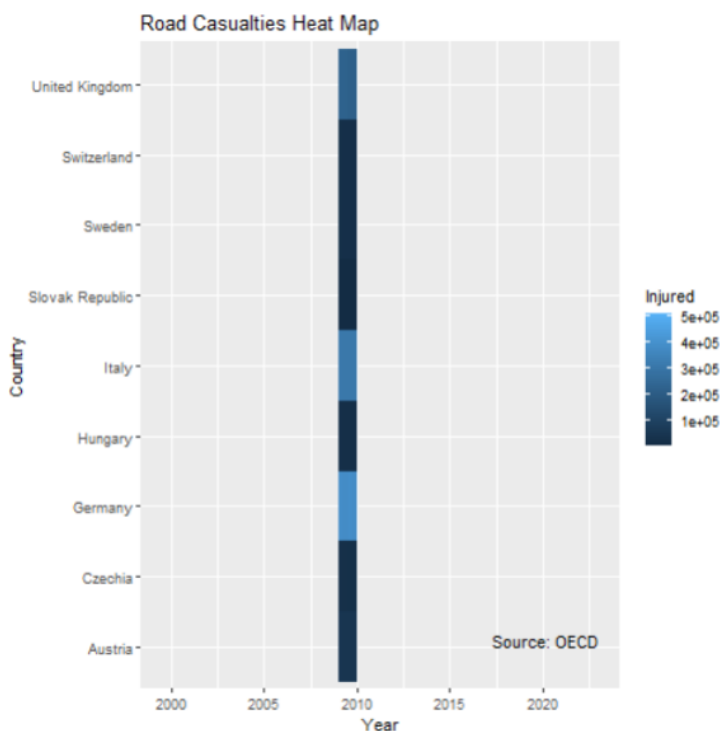
```
ggplot(ischaemic_deaths, aes(x = Year, y = Death_rate, group =
Country, color = Country)) +
  geom_line() +
  geom_point(size = 2) +
  transition_reveal(Year) +
  scale_y_log10(labels = scales::comma, breaks =
scales::trans_breaks("log10", function(x) 10^x)) +
  coord_cartesian(clip = 'off') +
  labs(title = 'Death Due to Other Ischaemic Heart Diseases Over
the Years',
        y = 'Death Rate (Log Scale)', color = 'Country') +
  theme_minimal() +
  theme(plot.margin = margin(5.5, 40, 5.5, 5.5),
        legend.position = "right")
```

Súradnicový systém je nastavený tak, aby zahŕňal všetky dátové body a neorezával žiadne body za hranicami grafu. Téma zápletky je prispôsobená minimalistickému štýlu so špecifikovanými okrajmi a umiestnením legendy. Kompletný kód uvádza **príloha č. 9**.

Táto vizualizácia je užitočná, pretože nám umožňuje vizuálne porovnať, ako sa miera úmrtnosti na srdcové choroby menila v priebehu času v rôznych krajinách. Mohli by sme opísať trendy, porovnať mieru medzi krajinami a prípadne vyvodit' závery o vplyve choroby srdca na rôzne populácie.

4.2.3 Vizualizácia údajov obetí na cestách

Táto animácia zobrazuje tepelnú mapu zobrazujúcu nehody na cestách v rôznych krajinách v priebehu času. Os x predstavuje roky, os y krajiny a intenzita farby (výplň) predstavuje počet obetí na cestách (zranených a mŕtvych osôb).



Animácia 3: Tepelná mapa,
zdroj: vlastné spracovanie

Pri tvorbe tepelnej mapy bolo potrebné vytvoriť objekt `ggplot` s názvom `heat_map`, ktorý mapuje rok na os x , krajinu na os y a počet obetí na výplň dlaždíc v tepelnej mape.

Vizualizácia sa v priebehu času mení vďaka funkcii `transition_time(Year)`, ktorá animuje zmeny v nehodách na cestách z roka na rok. Do grafu sme pridali anotáciu označujúcu zdroj údajov. Táto anotácia je umiestnená na základe maximálneho roku a minimálnej krajiny v súbore údajov. Kompletný kód uvádza **príloha č. 10**.

```
heat_map <- ggplot(your_data, aes(x = Year, y = Country, fill =
Injured)) +
  geom_tile() +
  labs(title = "Road Casualties Heat Map", x = "Year", y =
"Country", fill = "Injured") +
  transition_time(Year) +
  annotate("text", x = max(your_data$Year) + 1, y =
min(match(your_data$Country, your_data$Country)),
        label = "Source: OECD", hjust = 1, vjust = 0)
```

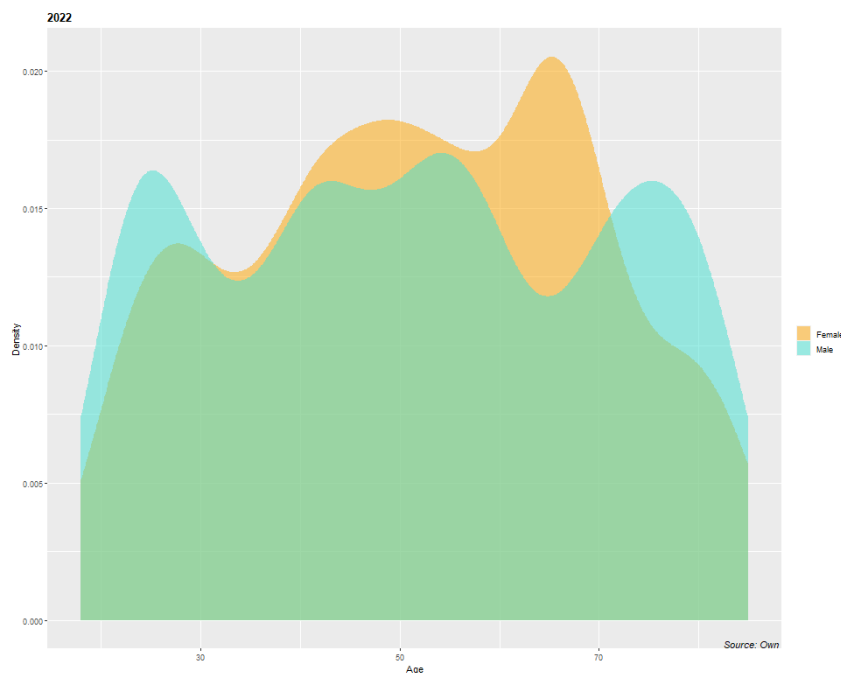
Animácia poskytuje dynamické znázornenie toho, ako sa nehody na cestách v jednotlivých krajinách v priebehu rokov líšia. Pomáha pri sledovaní trendov a vzorcov bezpečnosti na cestách alebo jej nedostatku.

4.2.4 Vizualizácia nasimulovaných dát

Vytvorili sme animovaný graf hustoty, ktorý zobrazuje zmeny v priebehu času v rozdelení veku v rámci simulovaného súboru údajov, rozdelený podľa pohlavia. Naša vizualizácia spája vek, pohlavie a čas, aby nám poskytla prehľad o tom, ako sa vekové rozdelenie v rámci mužských a ženských skupín vyvíja počas sledovaného obdobia.

Počiatkový segment kódu simuluje súbor údajov, ktorý zahŕňa rozsah veku (od 18 do 85 rokov), priraduje pohlavie ku každému záznamu (s pravdepodobnosťou 40 % pre ženy a 60 % pre mužov) a distribuuje ich. záznamy v sekvencii dátumov počnúc 1. januárom 2022.

```
df <- data.frame(
  age_years = sample(18:85, 1000, replace = TRUE),
  GenderTypeID = sample(c("female", "male"), 1000, replace = TRUE,
prob = c(0.4, 0.6)),
  SubmissionDateTime = seq(as.Date("2022-01-01"), by = "1 day",
length.out = 1000))
```



Animácia 4: Graf hustoty,
zdroj: vlastné spracovanie

Pred vizualizáciou sme údaje vyfiltrovali, aby sa vylúčili chýbajúce hodnoty veku, a potom sme manipulovali s cieľom extrahovať rok zo `SubmissionDateTime`. Funkcia `filter(!is.na(vek_roky))` odstráni všetky záznamy s chýbajúcimi údajmi o veku. Nakoniec je zoskupený podľa roku, veku a pohlavia, vďaka čomu je vhodný pre graf hustoty.

```
MSPs_age_gender <- df %>%
  filter(!is.na(age_years)) %>%
  mutate(year = year(SubmissionDateTime)) %>%
  select(year, age_years, GenderTypeID) %>%
  group_by(year, age_years, GenderTypeID)
```

Základný krok vizualizácie používa `ggplot2` na vytvorenie grafu hustoty. Na osi x sme zobrazili vek a na osi y hustotu (funkcia počtu záznamov). Záznamy sú rozdelené podľa pohlavia, reprezentované rôznymi farbami (oranžová pre ženy a tyrkysová pre mužov). Funkcia `geom_density(bw = 4, alfa = 0,5, typ čiary = 0)` pridá vrstvu grafu hustoty so špecifikovanou šírkou pásma, priehľadnosťou (`alfa`) a typom čiary. Grafy hustoty ukazujú distribúciu premennej (v našom kontexte veku). Názvy a menovky sme definovali pomocou funkcie `labs()`. Funkcie `theme()` a `scale_fill_manual()` sme použili na doladenie vzhľadu vrátane manipulácií s legendami a priradenia farieb pre kategórie pohlavia.

Použitím `gganimate` sme rozšírili `ggplot2` a to tak, že sme pridali časovú dynamiku do grafu. Komponent `transition_time(year)` zaisťuje, že graf sa vyvíja počas simulovaného časového obdobia a zobrazuje zmeny vo vekovej distribúcii podľa pohlavia v rôznych rokoch. Funkcie `enter_fade()` a `exit_shrink()` ovládajú ako prvky vstupujú a opúšťajú snímku počas animácie, čím sa zvyšuje vizuálna plynulosť. Graf tiež obsahuje vstupné a výstupné animácie pre plynulejšie prechody a používa funkciu `easing()`, aby animácia plynula prirodzenejšie. Kompletný kód uvádza **príloha č. 11**.

Grafy hustoty sú vynikajúce na zobrazenie distribúcie spojitej premennej a jej rozloženia v rôznych kategóriách (v našom prípade pohlavie). To môže pomôcť identifikovať šikmosť, alebo iné javy, ktoré nemusia byť zrejmé zo samotného priemeru alebo mediánu.

Vyplnením hustôt na základe pohlavia graf ľahko umožňuje porovnanie medzi mužským a ženským vekovým rozložením. Toto porovnanie môže zdôrazniť rozdiely v centrálnej tendencii, variabilite a tvare distribúcie medzi týmito dvoma pohlaviami.

Animovaný graf môže sprístupniť komplexné údaje rôznorodému publiku tým, že zjednoduší pochopenie zmien v priebehu času.

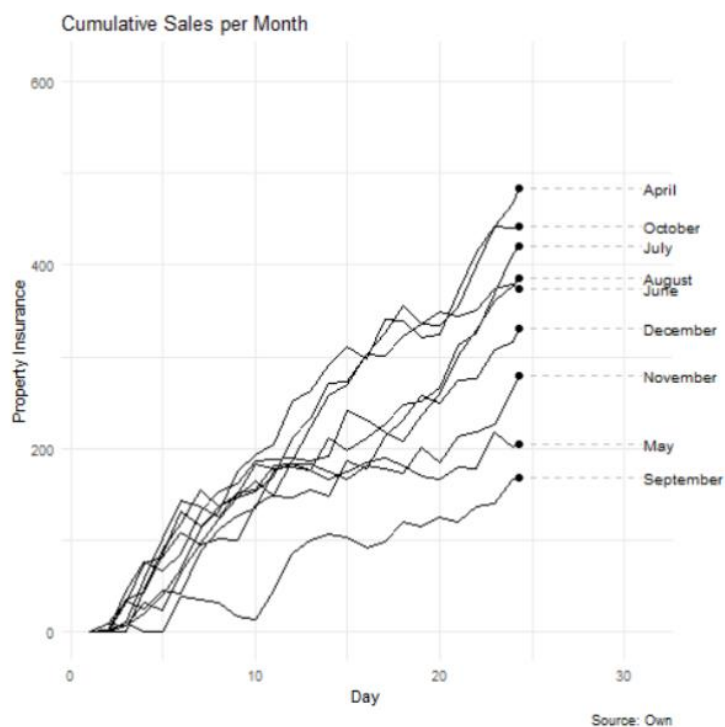
Animácia č. 5 vizuálne reprezentuje kumulatívne údaje o predaji počas niekoľkých mesiacov, konkrétne od apríla do decembra. Simulované údaje v našom kóde sú navrhnuté tak, aby predstavovali denné odchýlky v kumulatívnom predaji produktu „Poistenie majetku“ v rôznych mesiacoch (od apríla do decembra).

Po načítaní knižníc sme si ako prvé nastavili základ pre generovanie náhodných čísel na zabezpečenie reprodukovateľnosti, definovali mesiace (od apríla do decembra) a dni v mesiaci (1 až 30). Pomocou `expand.grid` sme vytvorili dátový rámec s názvom `data`, ktorý obsahuje každú kombináciu definovaných mesiacov a dní. To zaisťuje, že každý deň každého mesiaca je reprezentovaný ako jedinečný riadok v dátovom rámci.

```
set.seed(123)
months <- 4:12
days <- 1:30
data <- expand.grid(Month = months, Day = days)
```

Do dátového rámca sme pridali nový stĺpec s názvom `Kumulatif`, inicializovaný na 0. Tento stĺpec je určený na sledovanie kumulatívnych predajov za každý mesiac.

```
data$Kumulatif <- 0
```



Animácia 5: Čiarový graf,
zdroj: vlastné spracovanie

Cyklosom `for`-loop sme nasimulovali denné variácie predaja so všeobecným vzostupným trendom. Zabezpečuje, že kumulatívne tržby neklesnú pod 0 a budú sa akumulovať za každý deň a na začiatku každého nového mesiaca sa vynulujú.

```
for (i in 1:length(months)) {
  month_rows <- which(data$Month == months[i])
  daily_variation <- runif(length(month_rows) - 2, min = -20, max
= 50)
  data$Kumulatif[month_rows[1]] <- 0
  for (j in 2:length(month_rows)) {
    change <- daily_variation[j - 1]
    data$Kumulatif[month_rows[j]] <- max(0,
data$Kumulatif[month_rows[j-1]] + change)
  }
}
```

Číselnú reprezentáciu mesiacov sme previedli na reťazce ich názvov pre lepšiu čitateľnosť v grafe.

```
data$Month <- factor(data$Month, levels = months, labels =  
month.name[months])
```

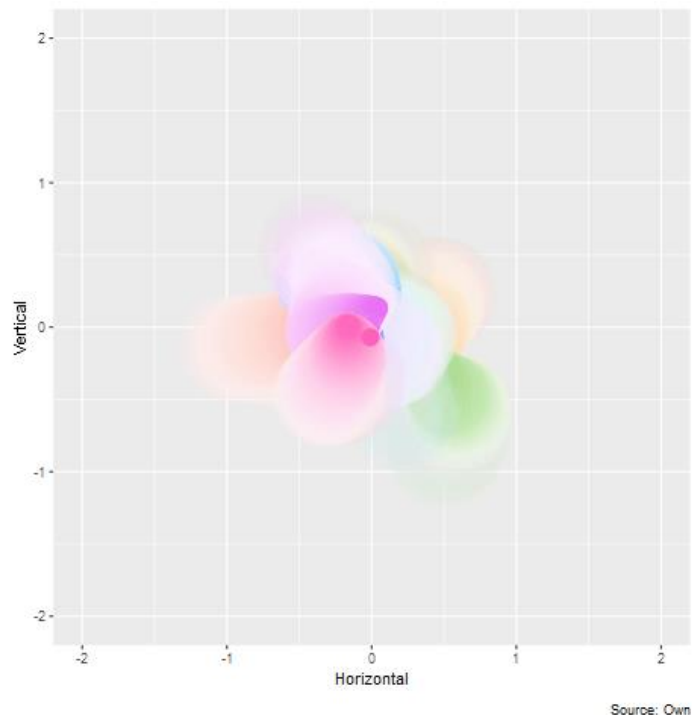
Funkcia `geom_line()` zabezpečuje, aby všetky čiary, ktoré zobrazujú kumulatívny predaj za každý deň v mesiaci boli čierne. Funkciou `geom_segment()` sme zaviedli sivé prerušované čiary, ktoré označujú koniec mesačného údajového bodu pre vizuálnu orientáciu. Dátové body `geom_point()` sme označili krúžkami, pretože zlepšujú vizuálnu reprezentáciu kumulatívnej hodnoty predaja každého dňa. Štítky mesiaca `geom_text()` sú umiestnené na konci údajov každého mesiaca, aby sa identifikoval mesiac. Funkciu `transition_reveal()` sme použili na animáciu grafu a odhalenie údajov zo dňa na deň. Osi a názvy grafu sú jasne označené, aby indikovali, že graf zobrazuje kumulatívne údaje o predaji za každý mesiac, s osobitným dôrazom na produkt „Poistenie majetku“. Kompletný kód uvádza **príloha č. 12**.

Hlavným účelom tejto animácie je ukázať, ako sa predaje môžu akumulovať v priebehu každého mesiaca. Vzostupný trend, napriek denným výkyvom (s náhodnými variáciami vrátane možných malých poklesov), naznačuje, že tržby celkovo majú tendenciu rásť v priebehu mesiaca. To by mohlo odrážať obchodný model, v ktorom bez ohľadu na denné vzostupy a pády je všeobecná trajektória predaja v priebehu mesiacov pozitívna.

Resetovaním kumulatívnych predajov na 0 na začiatku každého mesiaca a ich oddeleným sledovaním umožňuje animácia porovnanie medzi mesiacmi. To môže pomôcť identifikovať vzorce, ako napríklad, ktoré mesiace môžu mať silnejší rast predaja, ako sa dynamika predaja ku koncu roka zvyšuje alebo či niektoré mesiace vykazujú väčšiu volatilitu ako iné.

Stručne povedané, animácia rozpráva príbeh o raste, variabilite a porovnateľnom výkone v rôznych obdobiach a ponúka zjednodušený, ale ilustratívny model toho, ako sa môžu správať údaje o predaji v reálnom svete a ako ich možno analyzovať pre strategické poznatky.

Animácia č. 6 zobrazuje vizualizáciu predstavujúcu Brownove pohyby (*Brownian motion*) s náhodnými fluktuáciami v priebehu času. Graf zobrazuje sériu bodov, ktoré sa pohybujú pozdĺž cesty a prechádzajú cez rôzne časové body. Vertikálne a horizontálne pohyby bodov naznačujú kolísanie v čase, pričom každá séria má svoju vlastnú trajektóriu. [26] Simulované údaje, ktoré sme si vygenerovali poskytujú prehľad o tom, ako sa skupina poistnomatematických údajov správa v priebehu času s pridanými náhodnými výkyvmi. Efekt animácie zlepšuje pochopenie toho, ako sa tieto trajektórie menia v priebehu časových bodov.



Animácia 6: Brownove pohyby a fluktuácia časových sérií,
zdroj: vlastné spracovanie, podľa [26]

Spočiatku sme si nastavili parametre `ntimes` a `nseries` na určenie počtu časových bodov a časových radov, ktoré sa majú vygenerovať. Funkcia `make_bridges()` je definovaná na vytváranie Brownovho pohybu na základe poskytnutých parametrov časových bodov a radov.

```
ntimes <- 20
nseries <- 10
make_bridges <- function(ntimes, nseries) {
  replicate(nseries, c(0, rbridge(frequency = ntimes - 1))) %>%
  as.vector() }
```

Nižšie uvedený úryvok kódu zobrazuje ako sme nasimulovali aktuárske údaje s náhodnými fluktuáciami kombináciou Brownových pohybov so šumom pridaným cez `rnorm`.

```
tbl <- tibble(
  Time = rep(1:ntimes, nseries),
  Horizontal = make_bridges(ntimes, nseries) + rnorm(ntimes *
nseries, mean = 0, sd = 0.1),
  Vertical = make_bridges(ntimes, nseries) + rnorm(ntimes *
nseries, mean = 0, sd = 0.1),
  Series = gl(nseries, ntimes))
```

Funkciu `ggplot()` sme použili na vytvorenie základného grafu, kde sú dátové body vykreslené v 2D priestore s osou x ako horizontálne a s osou y ako vertikálne a zafarbené podľa série. Základný graf je rozdelený podľa času, čo ukazuje, ako sa údajové body vyvíjajú v každom časovom bode.

Animáciu sme vytvorili pomocou `gganimate` prechodom bodov v priebehu času, čím sa vytvorila dynamická vizualizácia toho, ako sa údaje menia v rôznych časových bodoch.

```
base_anim <- base_pic + transition_time(time = Time)
animate(base_anim)
```

K animácii sme pomocou funkcie `shadow_wake()` pridali tieňový efekt, aby sa vytvoril vizuálny efekt, ktorý vylepšuje pohyb údajových bodov. Kompletný kód uvádza **príloha č. 13**.

```
wake8 <- base_anim +
  shadow_wake(
    wake_length = 0.3,
    size = 15,
    colour = "white",
    falloff = "quintic-in"
  )
animate(wake8, detail = 5, type = "cairo")
```

Animácia nám umožňuje sledovať trendy a variabilitu poistno-matematických údajov. Generovaním viacerých sérií s Brownovými pohybmi a pridaním náhodného šumu môžeme vidieť, ako sa dátové body vyvíjajú v priebehu času a vykazujú výkyvy.

Pridanie náhodných fluktuácií k Brownovým pohybom simuluje stochastický proces. Pomáha to pri štúdiu toho, ako neisté faktory ovplyvňujú poistno-matematické údaje a ako sa tieto neistoty prejavujú v trajektóriách údajov. [26]

Animovaná vizualizácia vytvorená zo simulovaných údajov poskytuje dynamickú reprezentáciu vývoja údajov v priebehu času. Umožňuje nám vizuálne sledovať pohyb údajových bodov, vidieť vznikajúce vzory a pozorovať, ako sa údaje menia v každom časovom bode.

4.3 Praktická ukážka R Markdown

Na vytvorenie tejto prezentácie v PowerPointe pomocou R Markdown sme použili špecifický formát s názvom R Markdown (Rmd), ktorý umožňuje integráciu R kódu, naratívneho textu a výstupu (napríklad grafov) do jedného dokumentu.

Dokument začína metadátami špecifikujúcimi názov, autora, dátum, výstupný formát (prezentácia v PowerPointe) a ďalšie nastavenia (napríklad vykresľovanie HTML).

```
---
title: "***Analýza úmrtnosti na rakovinu**"
author: "Simona Buchancová"
date: "24-02-2024"
output:
  powerpoint_presentation:
    df_print: paged
always_allow_html: true
---
```

Každá snímka je reprezentovaná sekciou v dokumente R Markdown, ktorá je ohraničená dvojitými krížovými symbolmi (##) a tvorí nadpis. Podnadpis sme zahrnuli ako hlavičku sekcie, pred ktorou boli tri krížové symboly (###).

```
## Miera úmrtnosti na rakovinu
### Prečo na tom záleží?
```

V každej časti snímky je napísaný text, ktorý poskytuje kontext, vysvetlenie alebo hlavičky pre rôzne časti prezentácie.

Vďaka syntaxi R Markdown bolo možné jednoducho a efektívne integrovať obrázky do našej prezentácie:

1. Obrázok zo súboru

Túto syntax sme použili na priame vloženie obrázka uloženého lokálne v počítači do prezentácie. Na prvej snímke sme zahrnuli obrázok znázorňujúci rakovinové bunky na vizuálne vylepšenie obsahu.

```
![] (C:/Users/PC/Downloads/cancer-cells-mm.jpg)
```

2. Prepojený obrázok

Túto syntax sme použili na prepojenie obrázkov s externými adresami URL. To nám umožnilo kliknúť na obrázky v rámci prezentácie a byť presmerovaní na určené webové stránky čo zvýšilo interaktivitu a poskytlo ďalšie informácie súvisiace s obsahom prezentácie.

```
[![]](https://1.bp.blogspot.com/-EFP9V7TRoms/YT19gBeshoI/AAAAAAAAAU5U/1pz-mr0Kye8AnmhOVFhcNCw_3ZjZJfiNXgCLcBGAsYSYHQ/s1600)/lafler0/www/Heading.0jpg /complicated-calcs.html)
```

V poskytnutom kóde R Markdown sme formátovanie, ako je bold text a kurzíva, dosiahli pomocou syntaxe R Markdown.

1. **Bold text:** Bol text sme dosiahli tak, že sme želaný text obklopili dvojitými hviezdikami (**).

```
title: "***Analýza úmrtnosti na rakovinu**"
```

2. **Kurzíva:** Kurzívu sme nastavili tak, že sme text obklopili jednou hviezdikou (*).

```
date: "*24-02-2024*"
```

Odrážky sme vytvárali pomocou spojovníka (-), za ktorým nasleduje medzera:

```
- **Cena poistných produktov**  
- **Nastavenie rezervy**  
- **Vývoj produktov**
```

V PowerPoint prezentácii sme zobrazili 3 grafy: Miera úmrtnosti jednotlivých krajín, Výskyt rakoviny v jednotlivých krajinách a Analýza úmrtnosti krajín v priebehu času. Časti kódu R sú zahrnuté v dokumente pomocou trojitého spätného začiaroknutia, za ktorým nasleduje {r} na označenie R kódu a uzatváracieho trojitého spätného začiaroknutia.

V kóde R Markdown sme použili niekoľko balíkov na uľahčenie rôznych úloh, ako je manipulácia s údajmi, vizualizácia a vykresľovanie grafov. Balík `plotly` sme použili na generovanie interaktívnych grafov, ako sú čiarové grafy zobrazujúce trendy v úmrtnosti na rakovinu v priebehu času v rôznych krajinách. Ďalší balík `ggplot2` sme použili na vytvorenie grafov hustoty ilustrujúcich distribúciu výskytu rakoviny v rôznych krajinách. Na gene-

rovanie animovaného grafu zobrazujúceho trendy v úmrtnosti na rakovinu v priebehu rokov sme použili balík `gganimate`. Pre zachytenie snímok obrazovky interaktívnych grafov, ktoré sa majú zahrnúť do prezentácie programu PowerPoint sme použili balík `webshot2`.

Celkovo kombinácia týchto balíkov poskytuje komplexnú súpravu nástrojov na analýzu, vizualizáciu a prezentáciu údajov, ktorá umožňuje vytváranie informatívneho a vizuálne pútavého obsahu pre prezentáciu v PowerPointe o analýze úmrtnosti na rakovinu.

4.3.1 Tvorba grafu: Miera úmrtnosti jednotlivých krajín

Načítali sme knižnice a prešli na načítanie dát, ktoré obsahujú miery úmrtnosti na rakovinu pre vybrané krajiny. Dáta sme získali z portálu Eurostat. Funkcia `plotly()` inicializuje objekt grafu `plotly`. Ako hlavný argument berie dátový rámec `cancer_data` a špecifikuje premenné, ktoré sa majú vykresliť („Year“ na osi x a „Death Rate“ na osi y). Okrem toho nastavuje farbu značiek grafu na základe premennej „Country“. Tento operátor `%>%` umožňuje reťazenie viacerých operácií. Odovzdáva výstup jedného volania funkcie ako vstup pre ďalšie volanie funkcie.

```
p <- plot_ly(cancer_data, x = ~Year, y = ~Death_Rate, color =
~Country, colors = colors, type = 'scatter', mode = 'lines +
markers') %>%
  layout(title = 'Cancer Death Rate Trends',
         xaxis = list(title = 'Year'),
         yaxis = list(title = 'Death Rate'))
```

Pri vkladaní grafu do prezentácie v PowerPointe pomocou výstupného formátu prezentácie v R Markdown sa však stratí interaktivita grafu `plotly`, čo má za následok vloženie statického obrazu grafu do prezentácie. Dôvodom je, že PowerPoint priamo nepodporuje interaktívne grafy `plotly`. Pri vykresľovaní dokumentu R Markdown do programu PowerPoint sa graf `plotly` pred vloženíím do prezentačných snímok skonvertuje na formát statického obrázka (napr. PNG alebo JPEG). Tento proces prevodu zachytí snímku grafu v konkrétnom okamihu, pričom zachová jeho vzhľad, ale odstráni všetky interaktívne funkcie, ako sú efekty vznášania, približovanie alebo posúvanie.

Kvôli obmedzeniu PowerPointu sme na priame zobrazovanie interaktívnych grafov `plotly` zvolili prístup vloženia interaktívneho grafu ako súboru HTML a poskytnutia odkazu, na ktorý sa dá kliknúť na snímke programu PowerPoint, aby sme k nemu mali prístup. To nám umožnilo interagovať s grafom `plotly` mimo prostredia PowerPointu.

```
[Click here] (file://C:/Users/PC/Downloads/cancer_plotly.html) for  
more information.
```

4.3.2 Tvorba grafu: Výskyt rakoviny v jednotlivých krajinách

V prvom kroku sme si nasimulovali dáta ako štyri sady náhodných údajov (`spain_data`, `slovakia_data`, `poland_data`, `france_data`), ktoré sú generované pomocou funkcie `rnorm()`, ktorá generuje náhodné čísla z normálneho rozdelenia. Každý súbor pozostáva z 200 údajových bodov s rôznymi strednými hodnotami a hodnotami štandardnej odchýlky, ktoré predstavujú hypotetickú mieru výskytu rakoviny v každej krajine.

```
spain_data <- rnorm(200, mean = 100, sd = 20)  
slovakia_data <- rnorm(200, mean = 120, sd = 25)  
poland_data <- rnorm(200, mean = 90, sd = 18)  
france_data <- rnorm(200, mean = 110, sd = 22)
```

Následne sme vytvorili dátový rámec `df` s dvoma stĺpcami: „Country“ a „Cancer Occurrence“. Potom sme vytvorili graf hustoty pomocou balíka `ggplot2`.

```
gg <- ggplot(df, aes(x = Cancer_Occurrence, color = Country)) +  
  geom_density() +  
  geom_rug() +  
  ylab("") +  
  xlab("Cancer Occurrence")
```

V poslednom kroku sme objekt `ggplot` `gg` skonvertovali na objekt `plotly` pomocou funkcie `ggplotly()` z balíka `plotly`.

```

ggplotly(gg) %>%
  layout(
    plot_bgcolor = '#e5ecf6',
    xaxis = list(
      title = 'Cancer Occurrence',
      zerolinecolor = '#ffff',
      zerolinewidth = 2,
      gridcolor = '#ffff'),
    yaxis = list(
      title = 'Density',
      zerolinecolor = '#ffff',
      zerolinewidth = 2,
      gridcolor = '#ffff')
  )

```

4.3.3 Tvorba grafu: Analýza úmrtnosti krajín v priebehu času

Pri tomto grafe sme postupovali podobne ako pri grafe Výskyt rakoviny v jednotlivých krajinách. Načítali sme si knižnice a dáta, ktoré sme získali z Eurostatu. Vytvorili sme nový objekt ggplot s dátovým rámcom „cancer_data“ ako zdrojom údajov.

```

p <- ggplot(cancer_data, aes(x = Year, y = Death_Rate, group =
Country, color = Country)) +
  geom_line() +
  labs(title = 'Death Due to Heart Diseases Over the Years') +
  scale_y_log10() +
  scale_x_continuous(breaks = seq(2000, 2020, by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 11),
        panel.background = element_rect(fill = "black"),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        axis.line = element_blank(),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 10))

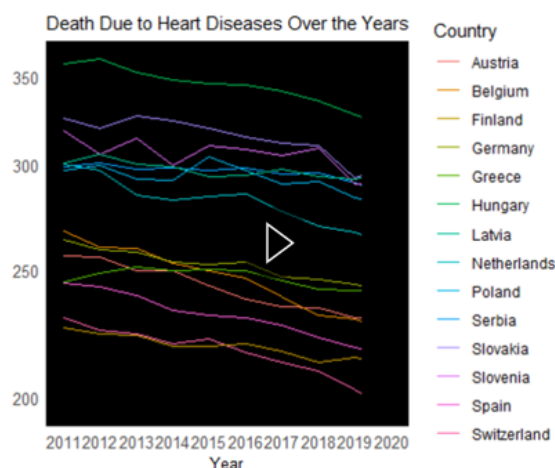
```

Estetické zobrazenia sme definovali funkciou `aes()`, kde $x = \text{„Year“}$ a $y = \text{„Death Rate“}$ a $color = \text{„Country“}$ zoskupuje čiary podľa krajiny. Do grafu sme pridali vrstvu čiar, ktorá spája dátové body pomocou funkcie `geom_line()`. Názov grafu sme nastavili cez funkciu `labs()`. Za pomoci funkcie `scale_y_log10()` sme naškálovali os y logaritmicke, aby sme lepšie vizualizovali údaje. Nastavili sme zlomy na osi x od roku 2 000 do roku 2020 s krokom 1 pomocou funkcie `scale_x_continuous()`. Aplikovali sme funkciu `theme_minimal()`, čím sme poskytli čistejší vizuál. Ďalšie funkcie `theme()` upravujú prvky grafu, ako je veľkosť nadpisu, farba pozadia, čiary mriežky a vlastnosti osi. Kompletný kód uvádza **príloha č. 14**.

V dôsledku nezobrazenia animácie, keď sme v prezentácii programu PowerPoint vygenerovanej pomocou R Markdown vybrali možnosť „Zobraziť snímky ako prezentáciu“ prikladám animáciu samostatne.

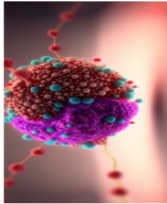

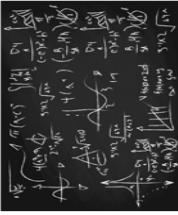

Analýza úmrtnosti krajín v priebehu času

Táto vizuálna reprezentácia zobrazuje trendy v úmrtnosti na rakovinu v rôznych krajinách v priebehu rokov.



Obrázok 29: Ukážka animácie v PowerPoint prezentácii ako výstupu z R Markdown, zdroj: vlastné spracovanie

Pri práci s programom R Markdown na generovanie prezentácií programu PowerPoint sa prispôsobenie vzhľadu našich snímok, ako je napríklad zmena farby textu alebo farby pozadia, riadi pomocou šablón programu PowerPoint a nastavení hlavičky YAML v dokumente R Markdown. Na rozdiel od HTML alebo iných webových formátov, kde by sme mali nad týmito aspektmi podrobnú kontrolu prostredníctvom CSS alebo vloženého štýlu, sú prezentácie v PowerPointe generované z R Markdown o niečo prísnejšie, pokiaľ ide o prispôsobenie. Farbu textu alebo pozadia, rovnako tak aj štýl textu môžeme zmeniť priamo v PowerPoint prezentácii až po jej vygenerovaní.

<p style="text-align: center;">Analýza úmrtnosti na rakovinu</p> <p style="text-align: right;">Simona Buchanová</p> <p style="text-align: right;"><small>14.03.2024</small></p>	<p style="text-align: center;">Miera úmrtnosti na rakovinu</p>  <p>Rakovina, jedna z hlavných príčin úmrtí na celom svete, predstavuje významné výzvy nielen pre verejné zdravie, ale aj pre odvetvie poisťovníctva. Aktuárstvo vo svojom jadre posudzuje riziko a neistotu, pričom sa často spolieha na mieru úmrtnosti, napríklad na rakovinu, pri vývoji finančných modelov, ktoré sú základom poisťných produktov.</p>	<p style="text-align: center;">Úmrtnosť na rakovinu: Ústredný bod v aktuárskej analýze</p> <p>Prečo na tom záleží?</p> <p>Analýza miery úmrtnosti na rakovinu poskytuje poisťným matematikom podrobné informácie o rizikách úmrtnosti spojených s rôznymi typmi rakoviny v rôznych demografických skupinách. Tieto informácie sú najdôležitejšie pre:</p> <ul style="list-style-type: none"> • Cena poisťných produktov • Nastavenie rezervy • Vývoj produktov 	1	2	3
<p style="text-align: center;">Cena poisťných produktov</p>  <p>Presná cena je základom udržateľných ponúk poistenia. Analýza miery úmrtnosti na rakovinu môžu poisťní matematici lepšie odhadnúť pravdepodobnosť výplat, pričom podľa toho upravujú poisťné, aby pokryli tieto riziká a zároveň zostali konkurenčne schopní.</p>	<p style="text-align: center;">Nastavenie rezervy</p>  <p>Poisťovne musia vyčleniť určitú sumu peňazí na krytie budúcich škôd. Podrobná analýza miery úmrtnosti, vrátane miery úmrtnosti na rakovinu, je nevyhnutná na určenie týchto rezerv, aby sa zabezpečilo, že poisťovateľ dokáže splniť svoje budúce záväzky.</p>	<p style="text-align: center;">Vývoj produktov</p>  <p>Pochopenie rizík spojených s rakovinou umožňuje poisťovníkam navrhovať produkty, ktoré spĺňajú potreby ich klientov, ako je poistenie kritických chorôb, a zároveň zabezpečiť finančnú životaschopnosť týchto produktov.</p>	4	5	6

Obrázok 30: Ukážka PowerPoint prezentácie ako výstupu z R Markdown, zdroj: vlastné spracovanie

Úloha vizualizácie údajov

Vizualizácia údajov hrá kľúčovú úlohu v poistnej matematike tým, že premenia zložité súbory údajov na zrozumiteľné a použiteľné informácie. **Ploty** umožňujú aktívne vyvierať interaktívne a pútavé vizualizácie, ktoré môžu zvýrazniť trendy a vzorce v miere úmrtnosti na rakovinu v priebehu času alebo v rôznych regiónoch. **Gganimate**, umožňuje aktívom vyhľadávať animácie, ktoré ilustrujú zmeny v priebehu času alebo geografických oblastí v miere úmrtnosti na rakovinu.

Miera úmrtnosti jednotlivých krajín

Interaktívny plot ilustruje zmeny v úmrtnosti v priebehu rokov v rôznych krajinách.

[Click here for more information.](#)

Výskyt rakoviny v jednotlivých krajinách

Analýza distribúcie údajov pomocou grafov hustoty môže odhaliť základné vzorce alebo trendy v priebehu času alebo medzi regiónmi, čo pomáha pochopiť epidemiológiu rakoviny.

[Click here for more information.](#)

7

8

9

Analýza úmrtnosti krajín v priebehu času

Táto vizuálna reprezentácia zobrazuje trendy v úmrtnosti na rakovinu v rôznych krajinách v priebehu rokov.

Ďakujem za pozornosť!

10

11

Obrázok 31: Ukážka PowerPoint prezentácie ako výstupu z R Markdown, zdroj: vlastné spracovanie

4.4 Praktická ukážka R Shiny

Náš kód vytvára pokročilý webový informačný panel na vizualizáciu a analýzu údajov o dopravných úmrtiach pomocou shiny rámca spolu s niekoľkými ďalšími balíkmi v R. Tento informačný panel umožňuje používateľom interagovať s údajmi rôznymi spôsobmi, vrátane zobrazenia prehľadu, analýzy trendov a porovnávania údajov medzi krajinami, prezzeranie ročných porovnaní a vytváranie predpovedí o budúcich trendoch dopravnej úmrtnosti na základe historických údajov. V tejto časti sme využívali balík `plotly` v rámci aplikácie Shiny dashboard pri tvorbe interaktívnych a vizuálne podmanivých grafov v rôznych častiach analýzy.

Používateľské rozhranie (UI)

V našom informačnom paneli je používateľské rozhranie (UI) vytvorené pomocou balíka `shinydashboard` pre štruktúru a rozloženie, spolu s balíkmi `plotly` a `ggplot2` pre interaktívne grafy.

Používateľské rozhranie obsahuje prispôbené CSS (Cascading Style Sheets)³ na úpravu štýlu, zmenu farby bočného panela, formát písom, veľkosť a farbu textu pre estetiku a čitateľnosť.

Používateľské rozhranie zahŕňa rozbaľovacie ponuky (`selectInput`), posúvače (`sliderInput`) a tlačidlá akcií (`actionButton`). Tieto prvky umožňujú používateľom interagovať s informačným panelom výberom krajín, rozsahov rokov a spúšťaním akcií, ako sú prognózy. Využíva dynamické a interaktívne grafy, na ktoré môžu používatelia umiestniť kurzor myši, približovať ich a filtrovať.

Štruktúra ovládacieho panela je definovaná pomocou `dashboardPage`, `dashboardHeader`, `dashboardSidebar` a `dashboardBody`. Bočný panel obsahuje položky ponuky pre rôzne karty: Prehľad údajov (Data Overview), Analýza trendov (Trend Analysis), Analýza údajov (Data Analysis), Ročné porovnanie (Yearly Comparison) a Predikcia (Prediction).

Serverová funkcia

Jadrom serverovej logiky je načítanie a príprava dát o dopravných úmrtiach pomocou balíka `readxl`. Tieto údaje slúžia ako základ pre všetky analýzy vykonávané na dashboarde.

³ CSS je jazyk šablóny so štýlmi, ktorý sa používa na popis prezentácie dokumentu napísaného v HTML alebo XML.

Serverová funkcia dynamicky aktualizuje dostupné voľby v rozbaľovacích ponukách na základe množiny údajov (napr. aktualizácia dostupných krajín) a reaguje na používateľské výbery a vstupy. Tu vstupuje do hry programovací model Shiny, ktorý umožňuje UI dynamicky reagovať na zmeny v používateľskom vstupe. Zahŕňa filtrovanie množiny údajov na základe používateľských výberov (napr. vybrané krajiny, ročné obdobie), vykonávanie výpočtov (napr. sumarizácia celkových úmrtí za rok) a používanie štatistických modelov na predpovede (balíček prognóz).

Generuje a aktualizuje výstupy (tabuľky s DT, grafy s plotly) na zobrazenie výsledkov analýzy alebo predikcie na základe vstupov a interakcií používateľa. Zahŕňa to vykresľovanie vizualizácií a tabuliek na strane servera, ktoré sa potom odošlú do používateľského rozhrania na zobrazenie.

Karty na bočnom paneli

1. Prehľad údajov (*Data Overview*)

Zobrazuje tabuľku údajov o dopravných úmrtiach. Primárnym cieľom je poskytnúť prístupný, informatívny a interaktívny súhrn množiny údajov o dopravných úmrtiach. Táto karta je kritická, pretože umožňuje používateľom oboznámiť sa so štruktúrou, obsahom a možno aj akýmkoľvek významnými trendmi alebo odľahlými hodnotami na prvý pohľad.

Pomocou balíka DT, ktorý poskytuje rozhranie pre knižnicu JavaScript 'DataTables', sme urobili tabuľku údajov interaktívnou. Používatelia môžu rýchlo nájsť konkrétne položky v množine údajov zadaním kľúčových slov do vyhľadávacieho poľa. Kliknutím na hlavičky stĺpcov môžu používatelia zoradiť tabuľku podľa hodnôt daného stĺpca, vzostupne alebo zostupne. Údaje sú rozdelené do stránok, cez ktoré môžu používatelia prechádzať pomocou ovládacích prvkov stránkovania. To zabraňuje tomu, aby bol používateľ zahltený veľkými množinami údajov.

```
dashboardBody(  
  tabItems(  
    tabItem(tabName = "overview_tab",  
            fluidRow(  
              box(DTOutput("data_overview"), width = 12)  
            )),  
  ),  
)
```

2. Analýza trendov (*Trend Analysis*)

Zameriava sa na analýzu a vizualizáciu trendu dopravných úmrtí v priebehu času pre vybrané krajiny. Táto funkcia umožňuje používateľom dynamicky skúmať vývoj smrteľných dopravných nehôd a ponúka prehľad o vzorcoch alebo zmenách v špecifikovanom súbore údajov.

Vstup výberu `selectInput` umožňuje používateľom vybrať si jednu alebo viaceru krajín z dynamicky generovaného zoznamu možností. Tento zoznam je vyplnený na základe jedinečných krajín prítomných v súbore údajov o dopravných úmrtiach.

```
tabItem(tabName = "trend_analysis_tab",
        fluidRow(
          box(title = "Select Countries",
              selectInput(inputId = "selected_countries",
                          label = "Countries",
                          choices = NULL,
                          multiple = TRUE,
                          selected = NULL),
              width = 4),
          box(plotlyOutput("trend_plot"), width = 8)
        )),
```

Keď používateľia umiestnia kurzor myši na dátový bod, zobrazia sa podrobné informácie (napr. presný počet úmrtí). Umožnili sme používateľom priblížiť konkrétne časové obdobia ako aj rozsahy údajov a posúvať sa po časovej osi.

3. Analýza údajov (*Data Analysis*)

Táto karta je navrhnutá tak, aby uľahčila porovnávaciu analýzu dopravných nehôd medzi dvoma vybranými krajinami počas špecifikovaného rozsahu rokov. Umožňuje používateľom do hĺbky preskúmať a porovnať rôzne trajektórie alebo vzorce smrteľných úrazov súvisiacich s premávkou medzi dvoma národmi, pričom odhalí poznatky, ktoré môžu byť kľúčové pre pochopenie faktorov vedúcich k rozdielom vo výsledkoch alebo pre posúdenie účinnosti opatrení bezpečnosti premávky implementovaných v rôznych regiónoch.

Dva widgety `selectInput` umožňujú používateľom vybrať dve krajiny označené ako `Country 1` a `Country 2`. Tieto rozbaľovacie ponuky sú vyplnené zoznamom krajín extrahovaných zo súboru údajov a umožňujú porovnávaciu analýzu medzi dvoma vybranými krajinami.

```

tabItem(tabName = "data_comparison_tab",
  fluidRow(
    box(title = "Country Selection",
      selectInput(inputId = "country_1",
        label = "Country 1",
        choices = NULL,
        selected = NULL),
      selectInput(inputId = "country_2",
        label = "Country 2",
        choices = NULL,
        selected = NULL),
      sliderInput("comparison_years_range",
        "Select Year Range:",
        min = 2000, max = 2020,
        value = c(2000, 2020),
        step = 1),
      width = 4),
    box(plotlyOutput("comparison_plot"), width = 8)
  )),

```

Miniaplikácia `sliderInput` umožňuje používateľom vybrať si rozsah rokov, pre ktoré chcú analyzovať úmrtnosť na cestách. Táto funkcia umožňuje skúmanie trendov v konkrétnych časových obdobiach a umožňuje používateľom vykonávať krátkodobé aj dlhodobé analýzy.

Po výbere krajín a ročného rozsahu aplikácia filtruje množinu údajov, aby získala relevantné údaje, teda záznamy týkajúce sa vybraných krajín v určenom časovom rámci. Filtrovaný súbor údajov sa potom použije na vytvorenie stĺpcového grafu, ktorý porovnáva ročné úmrtia v doprave medzi dvoma vybranými krajinami.

4. Ročné porovnanie (*Yearly Comparison*)

Komponent našej aplikácie Shiny je navrhnutý tak, aby umožnil makroskopický pohľad na trendy smrteľných dopravných nehôd v priebehu času, na ročnej báze, naprieč všetkými dostupnými údajmi. Táto funkcia je užitočná najmä pre zainteresované strany, ktoré chcú posúdiť dlhodobé trendy, identifikovať konkrétne roky s výrazne vysokým alebo nízkym

kým počtom úmrtí v doprave a pochopiť celkovú trajektóriu smrteľných nehôd súvisiacich s dopravou.

```
tabItem(tabName = "yearly_comparison_tab",
        fluidRow(
          box(title = "Year Selection",
              sliderInput("selected_years_range",
                          "Select Year Range:",
                          min = 2000, max = 2020,
                          value = c(2000, 2020),
                          step = 1,
                          post = ""
                        ),
              width = 4),
          box(plotlyOutput("yearly_comparison_plot"), width
=      8)
        )),
```

Používatelia majú možnosť interagovať s miniaplikáciou `sliderInput` a vybrať rozsah rokov na analýzu. Tento prvok umožňuje používateľom definovať časový rozsah ich dopytu, pričom ponúka flexibilitu zameranú na užšie časové rámce alebo rozšíriť svoj pohľad na širšie historické rozsahy.

5. Predikcia (Prediction)

Komponent Predikcia sme navrhli tak, aby využíval historické údaje o úmrtnosti v doprave na predpovedanie budúcich trendov pre vybranú krajinu. Táto funkcia je obzvlášť cenná pre tvorcov politik, výskumníkov a odborníkov, ktorí chcú premietnuť trendy dopravnej smrti do budúcnosti pre lepšie plánovanie a intervenčné stratégie.

Miniaplikácia `selectInput` umožňuje používateľom vybrať si jednu krajinu z dostupnej množiny údajov. Tento výber určuje rozsah údajov, ktoré sa majú použiť v predikčnom modeli.

Miniaplikácia `numericInput` umožňuje používateľom určiť počet rokov, ktoré chcú premietnuť do budúcnosti. Tento vstup priamo ovplyvňuje horizont prognózy generovanej aplikáciou.

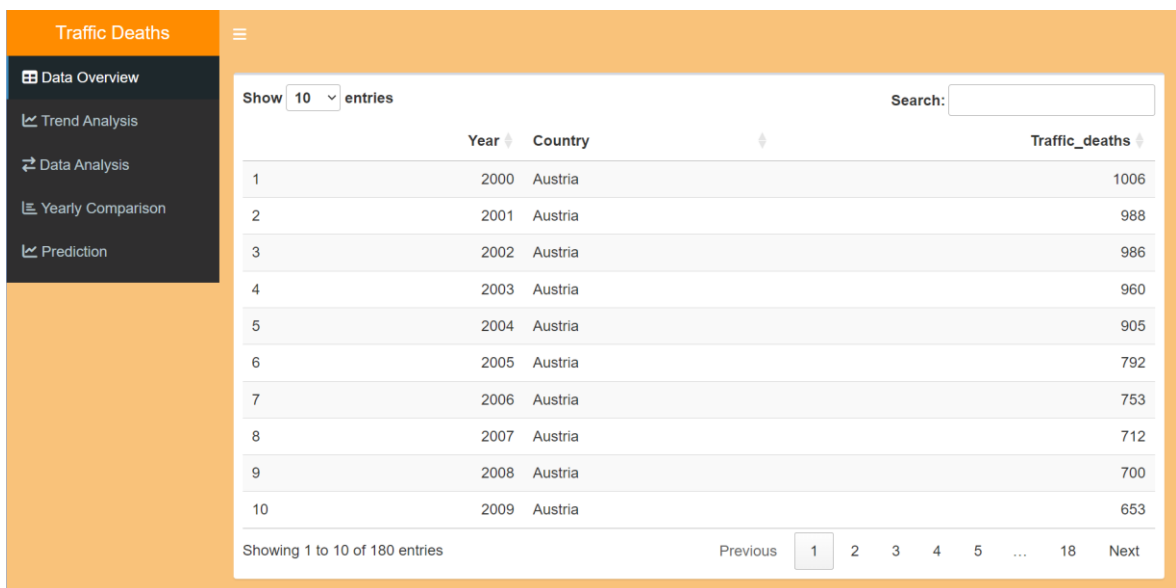
Tlačidlo akcie označené ako „Predpovedať“ slúži ako spúšťač na spustenie procesu predpovedania. Tento dizajn zaisťuje, že server vykonáva výpočtovo náročný proces prognózovania len vtedy, keď je používateľ pripravený, čím sa zvyšuje efektívnosť aplikácie.

```
tabItem(tabName = "prediction_tab",
        fluidRow(
          box(title = "Predictive Model Input",
              selectInput(inputId = "predict_country",
                          label = "Select Country",
                          choices = NULL,
                          selected = NULL),
              numericInput(inputId = "num_years",
                          label = "Number of Years to
Forecast",
                          value = 5),
              actionButton("predict_btn", "Predict"),
              width = 4),
          box(title = "Prediction Result with Linear Trend",
              plotlyOutput("prediction_plot"),
              width = 8)
        ))
```

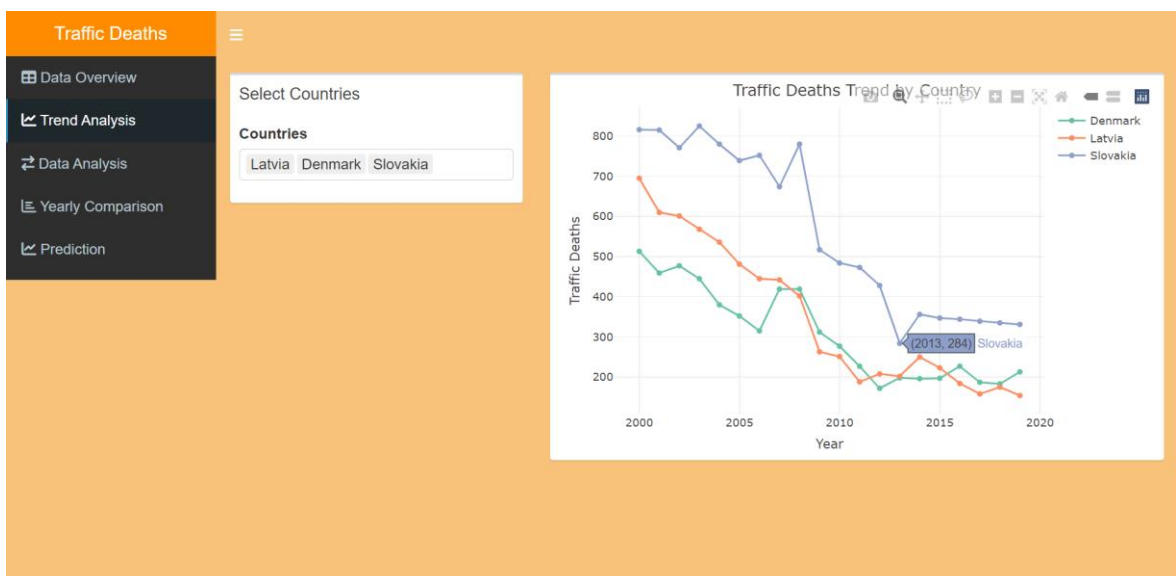
Po výbere krajiny a špecifikácii horizontu prognózy (počet rokov) a po spustení procesu predikcie aplikácia filtruje súbor údajov tak, aby obsahoval iba údaje týkajúce sa zvolenej krajiny.

Filtrovaný súbor údajov sa potom podrobí analýze časových radov, pričom sme použili lineárny regresný model, ktorý zohľadňuje trend v čase. Model používa historické údaje o dopravných úmrtiach na predpovedanie budúcich hodnôt na špecifikovaný počet rokov po poslednom dostupnom údajovom bode.

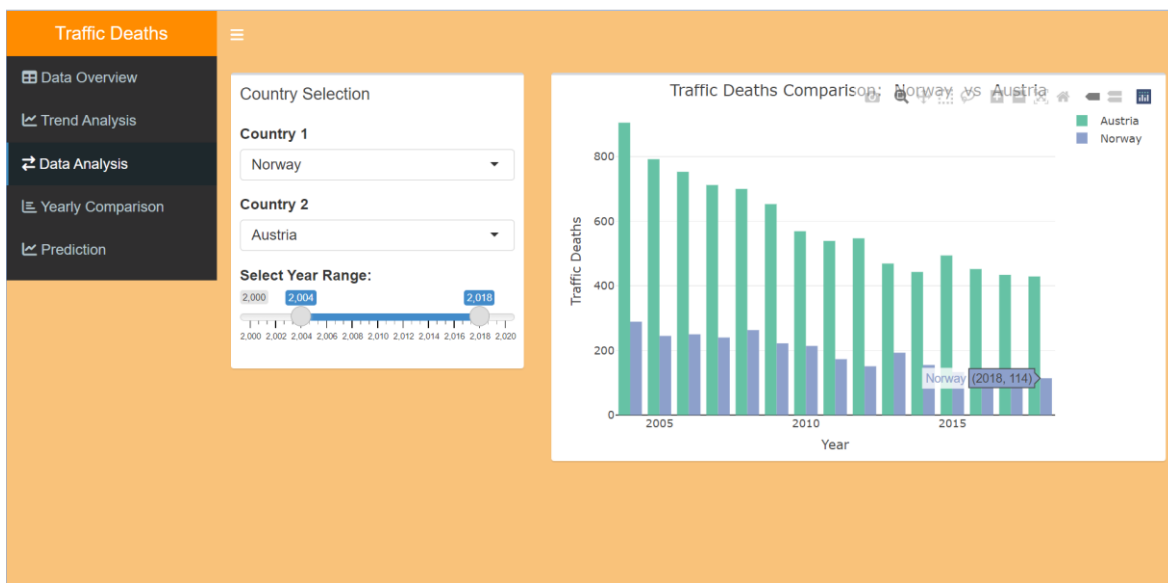
Pozoruhodnou vlastnosťou vizualizácie je zahrnutie intervalov spoľahlivosti okolo prognózovaných hodnôt, ktoré sú znázornené ako tieňované oblasti okolo predikčnej čiary. Tieto intervaly poskytujú pohľad na neistotu spojenú s predpoveďami a poskytujú používateľom predstavu o potenciálnych rozdieloch v budúcich výsledkoch. [20] Kompletný kód uvádza **príloha č. 15**.



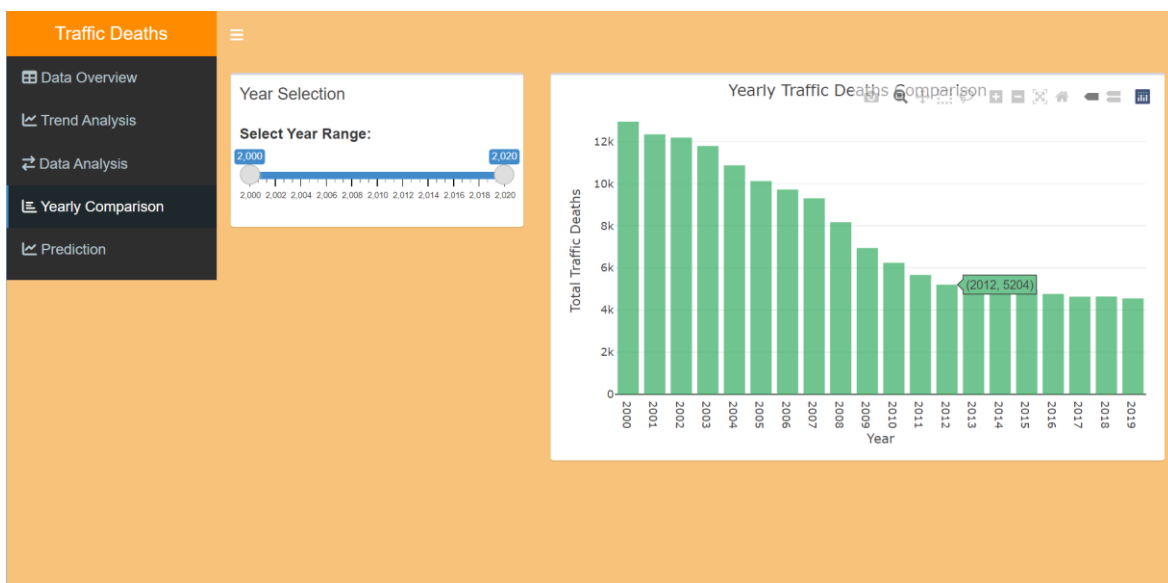
Obrázok 32: Vizuál karty *Data Overview*, zdroj: vlastné spracovanie



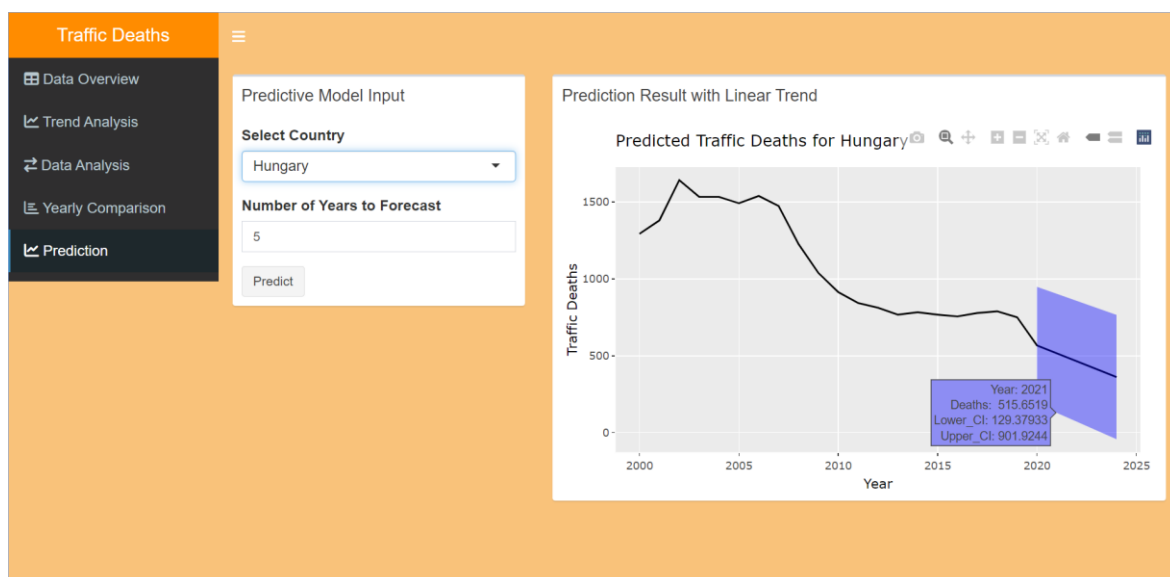
Obrázok 33: Vizuál karty *Trend Analysis*, zdroj: vlastné spracovanie



Obrázok 34: Vizuál karty *Data Analysis*, zdroj: vlastné spracovanie



Obrázok 35: Vizuál karty *Yearly Comparison*, zdroj: vlastné spracovanie



Obrázok 36: Vizuál karty *Prediction*, zdroj: vlastné spracovanie

ZÁVER

Skúmanie pokročilých metód a techník vizualizácie v kontexte aktuárskych údajov pomocou programovacieho jazyka R odhalilo silu a všestrannosť vizuálnej reprezentácie pri analýze údajov. Táto práca podrobne preskúmala vývoj, aplikácie a potenciál vizualizačných nástrojov a techník v R, zdôrazňujúc ich význam v porozumení dát a komunikácii záverov získaných z analýzy. Vďaka komplexnej sade špecializovaných nástrojov, ako sú knižnice `ggplot2`, `plotly`, `gganimate`, R Markdown a Shiny, s ktorými sme pracovali, môžeme potvrdiť, že je jazyk R efektívnym nástrojom pre výskumníkov a analytikov údajov, ktorí sa zameriavajú na štatistickú analýzu a vizualizáciu nielen v aktuárstve.

Evolúcia vizualizácie v jazyku R, od základných grafických systémov po integráciu nástrojov sofistikovaného ekosystému, výrazne zlepšila možnosti manipulácie a prezentácie údajov. Bezproblémový prechod od prípravy údajov k analýze a prezentácii v rámci jednotného pracovného postupu poukázal efektívnosť a prispôsobivosť R pri zvládaní rôznych úloh vizualizácie.

Integrácia interaktívnych prvkov, animačných techník a možnosti prispôsobenia dostupné v R nám poskytli komplexnú súpravu nástrojov na vytváranie presvedčivých a relevantných vizualizácií. Schopnosť prispôsobiť každý aspekt grafu, kombinovať rôzne typy grafov a začleniť interaktívne funkcie nám umožnilo efektívne sprostredkovať komplexné informácie o údajoch.

Pri práci s R Markdown na tvorbu prezentácií vo formáte PowerPoint sa ukázalo, že balík `plotly` má tendenciu nezobrazovať interaktívnu časť grafu priamo. Riešením tohto problému bolo vloženie HTML odkazu pod obrázok, aby sa interaktivita prejavila až po otvorení prezentácie. Naopak, animácie boli integrované bez problémov. Aj napriek tomu, že balík `gganimate` môže pri dlhších kódach generovať výstup pomalšie, jeho používanie bolo veľmi intuitívne.

Analýza pokročilých vizualizačných metód a nástrojov v našej práci nielenže potvrdila ich silu a všestrannosť, ale aj ich nevyhnutnosť v kontexte aktuárskeho vzdelávania. V kontexte požiadaviek syllabov aktuárskeho vzdelávania, ktoré kladú dôraz na efektívnu komunikáciu výsledkov, sme identifikovali, že vizualizácia hrá kľúčovú úlohu pri prezentovaní výsledkov aktuárskej práce rôznym cieľovým skupinám. Okrem toho nám vizualizácia umožňuje analyzovať a porozumieť dátam z hľadiska ich historických trendov a vývoja, čo je nevyhnutné pri riadení finančných rizík. Vzhľadom na neustály vývoj technológií a potre-

bu pre zrozumiteľnú prezentáciu výsledkov analýzy v kontexte narastajúcej komplexity dát, je jasné, že vizualizácia zostane nedeliteľnou súčasťou aktuárskej práce aj do budúcnosti.

Napriek rozsiahlym možnostiam R si však aj Python zachováva svoje miesto ako výkonný jazyk v oblasti analýzy údajov, strojového učenia a umelej inteligencie, vďaka svojej flexibilita a všestrannosti. Voľba medzi R a Pythonom potom závisí od špecifických cieľov projektu a preferencií používateľa, pričom oba jazyky ponúkajú jedinečné výhody pre rôzne aspekty analýzy a vizualizácie údajov.

Je dôležité poznamenať, že prezentácia určitých aspektov analýzy a vizualizácie údajov môže byť v tejto práci pre čitateľa náročná vzhľadom na požiadavku interaktivity. Dynamická povaha interaktívnych prvkov, ako sú animovaná grafika, komplexné vizualizácie alebo interaktívne nástroje na skúmanie údajov, sa nemusí efektívne prenášať do statických formátov. Niektoré časti práce si teda pre pochopenie vyžadujú interaktívnu prácu používateľa s uvedeným softvérom.

S nástupom dynamických a interaktívnych elementov v analýze a vizualizácii údajov, vznikajú nové výzvy spojené s efektívnou prezentáciou týchto prvkov v tradičných alebo statických médiách. Zdôrazňuje to potrebu po adaptabilných a inovatívnych platformách pre dátovú komunikáciu a prezentáciu.

Analýza pokročilých vizualizačných metód a nástrojov v našej práci odhalila hlboký potenciál týchto techník v rozširovaní možností pochopenia a interpretácie dát. V kombinácii s pokračujúcim vývojom a integráciou nových technológií predstavujú tieto nástroje „silnú zbraň“ v arzenáli data scientistov, ponúkajúc nové pohľady a podporujúc informované rozhodovanie v celej škále disciplín. Súčasný dynamický a interaktívny aspekt vizualizácie naznačujú, že cesta k vnímavejšej a zrozumiteľnejšej prezentácii údajov je evolučným procesom, ktorý bude pokračovať aj v nadchádzajúcich rokoch.

Zoznam použitej literatúry

- [1] BAUMER, S. Benjamin – KAPLAN, T. Daniel – HORTON, J. Nicholas. *Modern Data Science with R*. New York : CRC Press, 2021. 650 s. ISBN 978-0429577505.
- [2] BERENGUERES, Jose – SANDELL, Maribeth. *Introduction to Data Visualization & Storytelling*. New York : Independently Published, 2019. 188 s. ISBN 978-1085827836.
- [3] BISWAL, Avijeet. 2024. *What is Data Analytics?* In simplilearn.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.simplilearn.com/tutorials/data-analytics-tutorial/what-is-data-analytics>.
- [4] BLACK, Alison – LUNA, Paul – LUND, Ole – WALKER, Sue. *Information Design Research and Practice*. New York : Taylor & Francis, 2017. 766 s. ISBN 978-1317125280.
- [5] BRUSH, Kate – BURNS, Ed. 2022. *data visualization* In techtarget.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization>.
- [6] CAIRO, Alberto. *The Functional Art An Introduction to Information Graphics and Visualization*. California : Pearson Education, 2012. 384 s. ISBN 978-0133041361.
- [7] CHOY, Esther. 2023. *Say what? The art of storytelling* In theactuary.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.theactuary.com/2023/07/06/say-what-art-storytelling>.
- [8] CRAN. 2024. *Package 'plotly'* In cran.biotoools.fr [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://cran.biotoools.fr/web/packages/plotly/plotly.pdf>.
- [9] DABBAS, Elias. *Interactive Dashboards and Data Apps with Plotly and Dash*. Birmingham : Packt Publishing, 2021. 364 s. ISBN 978-1800560352.
- [10] DATASCIENCEPURSUIT. 2023. *The Data Science Process* In datasciencepursuit.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.datasciencepursuit.com/learn/courses/intro-to-data-science/the-data-science-process.html>.
- [11] DATASCIENCEPURSUIT. 2023. *The Data Science Process* In datasciencepursuit.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.datasciencepursuit.com/learn/courses/intro-to-data-science/understanding-data-science.html>.
- [12] DATASCIENCEPURSUIT. 2023. *What is Data Science?* In datasciencepursuit.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.datasciencepursuit.com/learn/courses/intro-to-data-science/what-is-data-science.html>.
- [13] GANDRUD, Christopher. *Reproducible Research with R and R Studio*. New York : CRC Press, 2018. 323 s. ISBN 978-1315360720.

- [14] GENDRON, Jay. *Introduction to R for Business Intelligence*. Birmingham : Packt Publishing, 2016. 228 s. ISBN 978-1785286513.
- [15] GEORGE-PALILONIS, Jennifer. *A Practical Guide to Graphics Reporting Information Graphics for Print, Web & Broadcast*. New York : Taylor & Francis, 2016. 232 s. ISBN 978-1317486350.
- [16] GRANJON, David. *Outstanding User Interfaces with Shiny*. New York : CRC Press, 2022. 510 s. ISBN 978-1000587081.
- [17] HASSAN, G. Hesham. 2021. *Designing Infographics to support teaching complex science subject: A comparison between static and animated Infographics* In proquest.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.proquest.com/openview/1e6e82ec6960555d4a5601a9babfe578/1?pqorigsite=gscholar&cbl=18750>.
- [18] HILLEBRAND, Julian – NIERHOF, Maximilian. *Mastering RStudio - Develop, Communicate, and Collaborate With R*. Birmingham : Packt Publishing, 2015. 322 s. ISBN 978-1783982547.
- [19] JURETIG, Francisco. *R Statistics Cookbook Over 100 Recipes for Performing Complex Statistical Operations with R 3.5*. Birmingham : Packt Publishing, 2019. 448 s. ISBN 978-1789802924.
- [20] KABACOFF, Robert. *R in Action, Third Edition Data Analysis and Graphics with R and Tidyverse*. New York : Manning, 2022. 656 s. ISBN 978-1617296055.
- [21] KABACOFF, Robert. *The Seven Pillars of Storytelling*. New York : Manning Publications Co., 2022. 656 s. ISBN 978-1617296055.
- [22] KASSAMBARA, Alboukadel. *R Graphics Essentials for Great Data Visualization*. South Caroline : CreateSpace Independent Publishing Platform, 2017. 145 s. ISBN 978-1979748100.
- [23] LANZETTA, B. Vitor. *R Data Visualization Recipes*. Birmingham : Packt Publishing, 2017. 334 s. ISBN 978-1788394079.
- [24] LI, Qi. *Embodying Data Chinese Aesthetics, Interactive Visualization and Gaming Technologies*. Singapur : Springer Nature Singapore, 2020. 182 s. ISBN 978-9811550690.
- [25] LINDSAY, Ffion. *The Seven Pillars of Storytelling*. Bristol : Sparkol Books, 2015. 86 s. ISBN 978-1853264696.
- [26] MAZO, M. Robert. *Brownian Motion Fluctuations, Dynamics, and Applications*. Oxford : Clarendon Press, 2008. 304 s. ISBN 978-0191565083.
- [27] MOULIK, Tania. *Applied Data Visualization with R and Ggplot2*. Birmingham : Packt Publishing, 2018. 140 s. ISBN 978-1789612158.

- [28] MURRAY, Scott. *Interactive Data Visualization for the Web*. Sebastopol : O'Reilly, 2017. 458s. ISBN 978-1449339739.
- [29] NORDMANN, Emily – MCALEER, Phil. 2022. *Data Visualization Using R for Researchers Who Do Not Use R* In journals.sagepub.com [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://journals.sagepub.com/doi/10.1177/25152459221074654>.
- [30] O'BRIEN, T. Daniel. *Urban Informatics Using Big Data to Understand and Serve Communities*. Oxon : CRC Press, 2022. 234 s. ISBN 978-1000781304.
- [31] PLOTLY. *R Figure Reference: layout* In plotly.com [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://plotly.com/r/reference/layout/#layout-autosize>.
- [32] POAGE, Jim – POAGE, Jennifer. 2016. *Five Reasons Why Storytelling is Such an Essential Tool for Actuaries* IN soa.org [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://www.soa.org/news-and-publications/newsletters/the-independent-consultant/2016/november/ind-2016-iss56/five-reasons-why-storytelling-is-such-an-essential-tool-for-actuaries/>.
- [33] RESNIZKY, G. Hernan. *Learning Shiny*. Birmingham : Packt Publishing, 2015. 246 s. ISBN 978-1785281990.
- [34] SCHMIDT, Johanna. 2020. *Usage of Visualization Techniques in Data Science Workflows* In scitepress.org [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://www.scitepress.org/PublishedPapers/2020/91819/91819.pdf>.
- [35] SIEVERT, Carson. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. New York : CRC Press, 2020. 470 s. ISBN 978-0429824203.
- [36] SUPPORT.POSIT. 2024. *Rendering PowerPoint Presentations with the RStudio IDE* In support.posit.co [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://support.posit.co/hc/en-us/articles/360004672913-Rendering-PowerPoint-Presentations-with-the-RStudio-IDE>.
- [37] TEUTONICO, Donato. *Ggplot2 Essentials*. Birmingham : Packt Publishing, 2015. 234 s. ISBN 978-1785283529.
- [38] TONIDANDEL, Scott – KING, Eden – CORTINA, Jose. *Big Data at Work: The Data Science Revolution and Organizational Psychology*. New York : Routledge, 2016. 382 s. ISBN 978-1848725812.
- [39] TREISCHL, J. Edgar. *Practice R*. Berlin : De Gruyter, 2023. 397 s. ISBN 978-3110704976.

- [40] WICKHAM, Hadlei – GROLEMUND, Garrett. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol : O'Reilly, 2016. 492 s. ISBN 978-1491910368.
- [41] WICKHAM, Hadlei. *Mastering Shiny*. Sebastopol : O'Reilly, 2021. 372 s. ISBN 978-1492047339.
- [42] WICKHAM, Hadley. *ggplot2: Elegant Graphics for Data Analysis*. New York : Springer, 2009. 211 s. ISBN 978-0387981406.
- [43] WILKE, Claus. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. Sebastopol : O'Reilly, 2019. 390 s. ISBN 978-1492031031.
- [44] WONG, M., Dona. *The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures*. China : Norton paperback, 2013. 160 s. ISBN 978-0393072952.
- [45] XIE, Yihui. *Dynamic Documents with R and Knitr*. New York : CRC Press, 2017. 294 s. ISBN 978-1315360706.
- [46] YU, Han – MOHARIL, Janhavi –BLAIR , Rachael. 2020. *BayesNetBP: An R Package for Probabilistic Reasoning in Bayesian Networks* In [jstatsoft.org](https://www.jstatsoft.org) [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://www.jstatsoft.org/article/view/v094i03>.
- [47] YUK, Mico. 2022. *Storytelling Through Data Analytics* In [theactuarmagazine.org](https://www.theactuarmagazine.org/storytellingthrough-data-analytics/) [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://www.theactuarmagazine.org/storytellingthrough-data-analytics/>.
- [48] ZELAZNY, Gene. 2001. *Say it with charts* In data.vk.edu.ee [online]. [citované dňa 23.03.2024]. Dostupné na internete: <https://data.vk.edu.ee/PowerBI/Opikud/Zelazny%20Gene%20Say%20It-%20With%20Charts.pdf>.
- [49] ZHANG, Zhongheng. 2016. *Naïve Bayes classification in R* In atm.amegroups.org [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://atm.amegroups.org/article/view/10175/11433>.
- [50] ZHOU, Jiawei – MIN, H. Seung. 2021. *An R Package for Easy and Elegant Data Visualization* In [frontiersin.org](https://www.frontiersin.org) [online]. [citované dňa 24.03.2024]. Dostupné na internete: <https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2021.802894/full>.

Prílohy

Príloha 1

Táto príloha obsahuje kompletný kód použitý na tvorbu *interaktívneho čiarového grafu*.

```
library(plotly)
library(dplyr)
ischaemic_deaths <-
read_csv("C:/Users/PC/Downloads/death_rate.csv",
         col_types = cols(
           Year = col_double(),
           Country = col_character(),
           Death_rate = col_double()
         ))

ischaemic_deaths$Year <-
as.numeric(as.character(ischaemic_deaths$Year))
interactive_plot <- ischaemic_deaths %>%
  plot_ly(
    x = ~Year,
    y = ~Death_rate,
    color = ~Country,
    text = ~paste("Death Rate:", Death_rate),
    hoverinfo = "text+x+name",
    type = 'scatter',
    mode = 'lines+markers'
  ) %>%
  layout(
    title = "Death Rate due to Heart Diseases",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Death Rate (%)"),
    hovermode = 'closest'
  )
interactive_plot
```

Príloha 2

Táto príloha obsahuje kompletný kód použitý na tvorbu *interaktívneho obrysového grafu*.

```
library(plotly)
library(readr)

df <- read_csv("C:/Users/PC/Downloads/death_rat.csv",
              col_types = cols(
                Year = col_double(),
                Country = col_character(),
                Death_rate = col_double()
              ))

fig <- plot_ly(data = df, x = ~Year, y = ~Country, z =
~Death_rate, type = "contour")
fig <- fig %>% layout(
  title = "Death Rate due to Heart Diseases",
  xaxis = list(title = 'Year'),
  yaxis = list(title = 'Country'),
  annotations = list(
    list(
      x = 1,
      y = -0.08,
      xref = "paper",
      yref = "paper",
      text = "Source: Eurostat",
      showarrow = FALSE,
      xanchor = "right",
      yanchor = "bottom"
    )
  )
)
fig
```

Príloha 3

Táto príloha obsahuje kompletný kód použitý na tvorbu interaktívneho stĺpcového grafu s tlačidlami a posúvačom.

```
library(plotly)
library(reader)

ischaemic_deaths <- read_csv("C:/Users/Simona
Buchancová/Downloads/death_rate.csv",
                             col_types = cols(Year = col_double(),
                                                Country =
col_character(),
                                                Death_rate =
col_double())
)

max_death_rate <- max(ischaemic_deaths$Death_rate)
ischaemic_deaths$Marker_Size <- 100 * ischaemic_deaths$Death_rate
/ max_death_rate

p <- plot_ly(ischaemic_deaths,
             x = ~Country,
             y = ~Death_rate,
             type = 'bar',
             text = ~paste("Country: ", Country, "<br>",
                           "Death Rate: ", Death_rate, "<br>",
                           "Year: ", Year),
             marker = list(color = ~Country),
             frame = ~Year) %>%
  layout(title = "Death Rates due to Heart Diseases",
         annotations = list(
           list(x = 1, y = 0,
                xref = "paper", yref = "paper",
                text = "Year: 2000",
                showarrow = FALSE,
                xanchor = "right",
                yanchor = "bottom"
           ),
           list(x = 1, y = 0,
                xref = "paper", yref = "paper",
                text = "Source: Your Source Attribution",
                showarrow = FALSE,
                xanchor = "right",
                yanchor = "bottom"
           )
         ),
         updatemenus = list(
           list(type = "buttons",
                buttons = list(
                  list(method = "animate",
                       args = list(NULL, animation_opts = 600,
redraw = TRUE),
                       label = "Play"),
```

```

        list(method = "animate",
              args = list(list(NULL), animation_opts = 0,
                           label = "Pause")
            ),
        direction = "left",
        x = 0.1,
        y = 1.05,
        xanchor = "right"
    )
    ),
    xaxis = list(title = "Country"),
    yaxis = list(title = "Death Rate (%)")
)

update_annotation <- function(frame){
  p <- p %>% layout(annotations = list(
    list(x = 1, y = 0,
         xref = "paper", yref = "paper",
         text = paste("Year: ", frame),
         showarrow = FALSE,
         xanchor = "right",
         yanchor = "bottom"
    )
  ))
}
p

```

Príloha 4

Táto príloha obsahuje kompletný kód použitý na tvorbu *interaktívneho boxplotu a scatter plotu*.

```
library(plotly)
library(dplyr)

traffic_data <-
read_csv("C:/Users/PC/Downloads/traffic_deaths_EU.csv",
         col_types = cols(
           Year = col_double(),
           Country = col_character(),
           Traffic_deaths = col_double()
         )
)

fig <- traffic_data %>%
  plot_ly(x = ~Country, y = ~Traffic_deaths) %>%
  layout(
    xaxis = list(
      title = "Country",
      range = c(10, 35)
    ),
    yaxis = list(
      title = "Traffic deaths",
      range = c(50, 500)
    )
  )
fig <- traffic_data %>%
  plot_ly(x = ~Country, y = ~Traffic_deaths, type = "scatter",
         name = "Scatter") %>%
  add_boxplot(name = "Boxplot")
fig
```

Príloha 5

Táto príloha obsahuje kompletný kód použitý na tvorbu interaktívnej mapy.

```
library(plotly)
set.seed(123)
countries <- c("Albania", "Andorra", "Austria", "Belgium", "Bosnia
and Herzegovina", "Bulgaria",
               "Croatia", "Cyprus", "Czech Republic", "Denmark",
               "Estonia", "Finland",
               "France", "Germany", "Greece", "Hungary",
               "Iceland", "Ireland", "Italy",
               "Latvia", "Liechtenstein", "Lithuania",
               "Luxembourg", "Malta", "Moldova",
               "Monaco", "Montenegro", "Netherlands", "North
Macedonia", "Norway", "Poland",
               "Portugal", "Romania", "San Marino", "Serbia",
               "Slovakia", "Slovenia",
               "Spain", "Sweden", "Switzerland", "Turkey",
               "Ukraine", "United Kingdom", "Vatican City")
suicide_rates <- runif(length(countries), min = 1, max = 30)

data <- data.frame(country = countries, suicide_rate =
suicide_rates)

map <- plot_ly(z = ~data$suicide_rate,
               locations = ~data$country,
               locationmode = "country names",
               type = "choropleth",
               colorscale = c('#FFE1A1', '#683531'),
               text = ~data$country,
               colorbar = list(title = "Suicide Rate"),
               hoverinfo = "text")

map <- map %>% layout(title = "Suicide Rates in Europe",
                    geo = list(scope = "europe"),
                    annotations = list(
                      text = "Source: Own",
                      showarrow = FALSE,
                      x = 0,
                      y = 1,
                      xref = "paper",
                      yref = "paper",
                      xanchor = "left",
                      yanchor = "top",
                      font = list(size = 12, color = "black")
                    ))

map
```

Príloha 6

Táto príloha obsahuje kompletný kód použitý na tvorbu *interaktívneho 3D scatter grafu*.

```
library(plotly)
accidents <- rpois(100, lambda = 10)
severity <- accidents + rnorm(100, mean = 5, sd = 2)
time_of_day <- runif(100, min = 0, max = 24)
driver_age <- sample(18:80, 100, replace = TRUE)
accidents_data <- data.frame(accidents, severity, time_of_day,
age)

midpoint_x <- mean(range(accidents_data$accidents))
midpoint_y <- mean(range(accidents_data$severity))
midpoint_z <- mean(range(accidents_data$time_of_day))

fig <- plot_ly(data = accidents_data, x = ~accidents, y =
~severity, z = ~time_of_day,
marker = list(color = ~driver_age, colorscale =
c('#FFE1A1', '#683531'), showscale = TRUE)) %>%
add_markers() %>%
layout(scene = list(xaxis = list(title = 'Accidents'),
yaxis = list(title = 'Severity'),
zaxis = list(title = 'Time of Day'),
annotations = list(
list(
x = 1.1,
y = -0.1,
xref = 'paper',
yref = 'paper',
text = 'Source: Own',
showarrow = FALSE
)
)),
title = list(text = 'Accidents Analysis', x = 0.5)
)

fig
```

Príloha 7

Táto príloha obsahuje kompletný kód použitý na tvorbu *interaktívneho Bayesovského Poissonovho regresného modelu*.

```
devtools::install_github("stan-dev/bayesplot", dependencies =
TRUE, build_vignettes = FALSE)
library(plotly)
library(bayesplot)
library(rstanarm)

set.seed(123)
n <- 1000
age <- rpois(n, lambda = 40) + 20
claims <- rpois(n, lambda = age/10)

actuarial_data <- data.frame(age = age, claims = claims)

fit <- stan_glm(claims ~ age, data = actuarial_data, family =
"poisson")

posterior <- as.matrix(fit)

plot_title <- ggtitle("Posterior distributions",
                      "with medians and 80% intervals")
mcmc_areas(posterior,
           pars = c("(Intercept)", "age"),
           prob = 0.8) + plot_title

plot_ppc <- ppc_dens_overlay(y = actuarial_data$claims,
                           yrep = posterior_predict(fit, draws =
50))

ppc_plotly <- ggplotly(plot_ppc)
ppc_plotly
```

Príloha 8

Táto príloha obsahuje kompletný kód použitý na tvorbu *stĺpcového grafu* ako animácie.

```
library(ggplot2)
library(gganimate)
library(tweenr)
library(dplyr)
library(readr)

traffic_death <-
read_csv("C:/Users/PC/Downloads/traffic_deaths_EU.csv")

traffic_data <-
read_csv("C:/Users/PC/Downloads/traffic_deaths_EU.csv",
         col_types = cols(
           Year = col_double(),
           Country = col_character(),
           Traffic_deaths = col_double()
         )
)

traffic_data <- traffic_data %>%
  arrange(desc(Traffic_deaths))

traffic_data <- traffic_data %>%
  group_by(Year) %>%
  mutate(ordering = rank(Traffic_deaths, ties.method = "first"))
%>%
  ungroup()

traffic_data$Year <- as.integer(traffic_data$Year)

caption <- "Source: WHO"

fpgw <- 10
end_gw <- 38
limit <- 25
fps <- 15
end_pause <- 150
xlab <- "Countries"
ylab <- "Traffic Deaths"

make_barchart_race <- function(title = "Title", fps, end_pause) {

  p <- ggplot(data = traffic_data, aes(x = ordering, y =
Traffic_deaths,
                                     fill = Country, label =
Country, text = paste("Year: ", Year))) +
    geom_bar(stat = "identity", color = "black") +
    coord_flip(clip = "off", expand = FALSE) +
    geom_text(aes(label = Country), hjust = -0.1, vjust = 0.5,
show.legend = FALSE) +
```

```

labs(title = title,
      subtitle = 'Year: {frame_time}',
      x = xlab, y = ylab,
      caption = caption) +
theme_minimal() +
transition_time(Year) +
enter_fade() +
exit_fade() +
ease_aes('linear') +
view_follow(fixed_x = TRUE)

animate(p, nframes = fpgw * end_gw + end_pause, fps = fps,
end_pause = end_pause)
}

make_barchart_race(title = "Traffic Deaths by Country 2000-2019",
fps = fps, end_pause = end_pause)

anim_save("traffic_deaths_animation1.gif")

```

Príloha 9

Táto príloha obsahuje kompletný kód použitý na tvorbu *čiarového grafu* ako animácie.

```
library(readr)
library(ggplot2)
library(gganimate)

ischaemic_deaths <-
read_csv("C:/Users/PC/Downloads/death_rate.csv",
         col_types = cols(
           Year = col_double(),
           Country = col_character(),
           Death_rate = col_double()
         ))

ggplot(ischaemic_deaths, aes(x = Year, y = Death_rate, group =
Country, color = Country)) +
  geom_line() +
  geom_point(size = 2) +
  transition_reveal(Year) +
  scale_y_log10(labels = scales::comma, breaks =
scales::trans_breaks("log10", function(x) 10^x)) +
  coord_cartesian(clip = 'off') +
  labs(title = 'Death Due to Other Ischaemic Heart Diseases Over
the Years',
       y = 'Death Rate (Log Scale)', color = 'Country') +
  theme_minimal() +
  theme(plot.margin = margin(5.5, 40, 5.5, 5.5),
        legend.position = "right")

anim_save("ischaemic_deaths_animation1.gif")
```

Príloha 10

Táto príloha obsahuje kompletný kód použitý na tvorbu *tepelnej mapy* ako animácie.

```
library(gganimate)
library(ggplot2)

traffic_data <-
read_csv("C:/Users/PC/Downloads/traffic_deaths_EU.csv",
         col_types = cols(
           Year = col_double(),
           Country = col_character(),
           Traffic_deaths = col_double()
         )
)

heat_map <- ggplot(your_data, aes(x = Year, y = Country, fill =
Injured)) +
  geom_tile() +
  labs(title = "Road Casualties Heat Map", x = "Year", y =
"Country", fill = "Injured") +
  transition_time(Year) +
  annotate("text", x = max(your_data$Year) + 1, y =
min(match(your_data$Country, your_data$Country)),
         label = "Source: OECD", hjust = 1, vjust = 0)

anim_save("traffic_deaths_heatmap.gif")
```

Príloha 11

Táto príloha obsahuje kompletný kód použitý na tvorbu *grafu hustoty* ako animácie.

```
library(tidyverse)
library(ggthemes)
library(gganimate)
library(lubridate)

df <- data.frame(
  age_years = sample(18:85, 1000, replace = TRUE),
  GenderTypeID = sample(c("female", "male"), 1000, replace = TRUE,
prob = c(0.4, 0.6)),
  SubmissionDateTime = seq(as.Date("2022-01-01"), by = "1 day",
length.out = 1000)
)

MSPs_PQs_age_gender <- df %>%
  filter(!is.na(age_years)) %>%
  mutate(year = year(SubmissionDateTime)) %>%
  select(year, age_years, GenderTypeID) %>%
  group_by(year, age_years, GenderTypeID)

p3 <- ggplot(MSPs_PQs_age_gender, aes(x = age_years, fill =
factor(GenderTypeID))) +
  geom_density(bw = 4, alpha = 0.5, linetype = 0) +
  labs(title = "{round(frame_time, 0)}", x = "Age", y = "Density")
+
  theme(legend.title = element_blank(), plot.title =
element_text(face = "bold")) +
  scale_fill_manual(values = c("orange", "turquoise"), labels =
c("female", "male")) +
  transition_time(year) +
  enter_fade() +
  exit_shrink() +
  ease_aes('sine-in-out') +
  geom_text(aes(x = 85, y = 0, label = "Source: Own"), hjust = 1,
vjust = 0, size = 3, color = "black")

animate(p3, width = 1000, height = 800)
anim_save("age_gender_density.gif", animation = p3, renderer =
gifski_renderer(), duration = 10, fps = 10)
```

Príloha 12

Táto príloha obsahuje kompletný kód použitý na tvorbu *čiarového grafu* s mesiacmi ako animácie.

```
set.seed(123)
library(ggplot2)
library(gganimate)
months <- 4:12
days <- 1:30
data <- expand.grid(Month = months, Day = days)
data$Kumulatif <- 0
for (i in 1:length(months)) {
  month_rows <- which(data$Month == months[i])
  daily_variation <- runif(length(month_rows), min = -20, max =
50)
  data$Kumulatif[month_rows[1]] <- 0
  for (j in 2:length(month_rows)) {
    change <- daily_variation[j - 1]
    data$Kumulatif[month_rows[j]] <- max(0,
data$Kumulatif[month_rows[j - 1]] + change)
  }
}

data$Month <- factor(data$Month, levels = months, labels =
month.name[months])
ggplot(data, aes(Day, Kumulatif, group = Month)) +
  geom_line() +
  geom_segment(aes(xend = 31, yend = Kumulatif), linetype = 2,
colour = 'grey') +
  geom_point(size = 2) +
  geom_text(aes(x = 31.1, label = Month), hjust = 0) +
  transition_reveal(Day) +
  coord_cartesian(clip = 'off') +
  labs(title = 'Cumulative Sales per Month', y = 'Property
Insurance') +
  theme_minimal() +
  theme(plot.margin =margin(5.5, 40, 5.5, 5.5))+
  annotate("text", x = 31, y = -20, label = "Source: Own", hjust =
1, size = 3, color = "black")
```

Príloha 13

Táto príloha obsahuje kompletný kód použitý na tvorbu *Brownovho pohybu* ako animácie.

```
library(tidyverse)
library(gganimate)
library(e1071)

ntimes <- 20
nseries <- 10

make_bridges <- function(ntimes, nseries) {
  replicate(nseries, c(0, rbridge(frequency = ntimes - 1))) %>%
  as.vector()
}

tbl <- tibble(
  Time = rep(1:ntimes, nseries),
  Horizontal = make_bridges(ntimes, nseries) + rnorm(ntimes *
nseries, mean = 0, sd = 0.1),
  Vertical = make_bridges(ntimes, nseries) + rnorm(ntimes *
nseries, mean = 0, sd = 0.1),
  Series = gl(nseries, ntimes)
)

base_pic <- tbl %>%
  ggplot(aes(x = Horizontal, y = Vertical, colour = Series)) +
  geom_point(show.legend = FALSE, size = 5) +
  coord_equal() +
  xlim(-2, 2) +
  ylim(-2, 2) +
  theme(
    plot.caption = element_text(hjust = 1, margin = margin(t =
10)),
    plot.caption.position = "plot"
  ) +
  labs(caption = "Source: Own")

base_pic + facet_wrap(~ Time)

base_anim <- base_pic + transition_time(time = Time)
animate(base_anim)

wake8 <- base_anim +
  shadow_wake(
    wake_length = 0.3,
    size = 15,
    colour = "white",
    falloff = "quintic-in"
  )
animate(wake8, detail = 5, type = "cairo")

anim_save("shadow.gif")
```

Príloha 14

Táto príloha obsahuje kompletný kód použitý na tvorbu *PowerPoint prezentácie* ako výstupu z R Markdown.

```
---
title: "***Analýza úmrtnosti na rakovinu**"
author: "Simona Buchancová"
date: "24-02-2024"
output:
  powerpoint_presentation:
    df_print: paged
always_allow_html: true
---
```

Miera úmrtnosti na rakovinu

Rakovina, jedna z hlavných príčin úmrtí na celom svete, predstavuje významné výzvy nielen pre verejné zdravie, ale aj pre odvetvie poisťovníctva. Aktuárstvo vo svojom jadre posudzuje riziko a neistotu, pričom sa často spolieha na mieru úmrtnosti, napríklad na rakovinu, pri vývoji finančných modelov, ktoré sú základom poisťných produktov.

Úmrtnosť na rakovinu: Ústredný bod v aktuárskej analýze

Prečo na tom záleží?

Analýza miery úmrtnosti na rakovinu poskytuje poisťným matematikom podrobné informácie o rizikách úmrtnosti spojených s rôznymi typmi rakoviny v rôznych demografických skupinách. Tieto informácie sú najdôležitejšie pre:

- **Cena poisťných produktov**
- **Nastavenie rezervy**
- **Vývoj produktov**

Cena poisťných produktov

Presná cena je základom udržateľných ponúk poistenia. Analýzou miery úmrtnosti na rakovinu môžu poisťní matematici lepšie odhadnúť pravdepodobnosť výplat, pričom podľa toho upravujú poisťné, aby pokryli tieto riziká a zároveň zostali konkurencie schopní.

Nastavenie rezervy

Poisťovne musia vyčleniť určitú sumu peňazí na krytie budúcich škôd. Podrobná analýza miery úmrtnosti, vrátane miery úmrtnosti na

rakovinu, je nevyhnutná na určenie týchto rezerv, aby sa zabezpečilo, že poisťovateľ dokáže splniť svoje budúce záväzky.

```
[![] (https://1.bp.blogspot.com/-EFP9V7TRoms/YT19gBeshoI/AAAAAAAAU5U/1pzmr0Kye8AnmhOVFhcNCw_3ZjZJfiNXgCLcBGAsYHQ/s1600/Heading.jpg)] (https://www.flerlagetwins.com/2021/09/complicated-calcs.html)
```

****Vývoj produktov****

Pochopenie rizík spojených s rakovinou umožňuje poisťovníam navrhovať produkty, ktoré spĺňajú potreby ich klientely, ako je poistenie kritických chorôb, a zároveň zabezpečiť finančnú životaschopnosť týchto produktov.

```
[![] (https://assets-global.website-files.com/603f10760571345ee07ddf0b/60c7e64efb2d0e0846a0348a_how_to_cross_sell_more_insurance_products.jpg)] (https://www.gonectar.com/blog/how-to-cross-sell-more-insurance-products)
```

****Úloha vizualizácie údajov****

****Vizualizácia údajov**** hrá kľúčovú úlohu v poistnej matematike tým, že premieňa zložité súbory údajov na zrozumiteľné a použiteľné informácie. ****Plotly**** umožňuje aktuárom vytvárať interaktívne a pútavé vizualizácie, ktoré môžu zvýrazniť trendy a vzorce v miere úmrtnosti na rakovinu v priebehu času alebo v rôznych regiónoch. ****Gganimate****, umožňuje aktuárom vytvárať animácie, ktoré ilustrujú zmeny v priebehu času alebo geografických oblastí v miere úmrtnosti na rakovinu.

****Miera úmrtnosti jednotlivých krajín****

Interaktívny plot ilustruje zmeny v úmrtnosti v priebehu rokov v rôznych krajinách.

```
```{r echo=FALSE, message=FALSE}
library(plotly)
library(webshot2)
library(readr)

cancer_data <- read_csv("C:/Users/PC/Downloads/cancer_data.csv",
 col_types = cols(
 Year = col_double(),
 Country = col_character(),
 `Death Rate` = col_double()
))

colors <- c('#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
'#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf')

p <- plot_ly(cancer_data, x = ~Year, y = ~`Death Rate`, color =
~Country, colors = colors, type = 'scatter', mode =
'lines+markers') %>%
 layout(title = 'Cancer Death Rate Trends',
 xaxis = list(title = 'Year'),
 yaxis = list(title = 'Death Rate'))
```

```

p
...
[Click here] (file:///C:/Users/PC/Downloads/plotlyPlot.html) for
more information.

Výskyt rakoviny v jednotlivých krajinách

Analýza distribúcie údajov pomocou grafov hustoty môže odhaliť
základné vzorce alebo trendy v priebehu času alebo medzi regiónmi,
čo pomáha pochopiť epidemiológiu rakoviny.

```{r}

set.seed(1)

countries <- c('Spain', 'Slovakia', 'Poland', 'France')

spain_data <- rnorm(200, mean = 100, sd = 20)
slovakia_data <- rnorm(200, mean = 120, sd = 25)
poland_data <- rnorm(200, mean = 90, sd = 18)
france_data <- rnorm(200, mean = 110, sd = 22)

df <- data.frame(
  Country = factor(rep(countries, each = 200)),
  Cancer_Occurrence = c(spain_data, slovakia_data, poland_data,
france_data)
)

gg <- ggplot(df, aes(x = Cancer_Occurrence, color = Country)) +
  geom_density() +
  geom_rug() +
  ylab("") +
  xlab("Cancer Occurrence")

ggplotly(gg) %>%
  layout(
    plot_bgcolor = '#e5ecf6',
    xaxis = list(
      title = 'Cancer Occurrence',
      zerolinecolor = '#ffff',
      zerolinewidth = 2,
      gridcolor = '#ffff'),
    yaxis = list(
      title = 'Density',
      zerolinecolor = '#ffff',
      zerolinewidth = 2,
      gridcolor = '#ffff')
  )
...
[Click here] (file:///C:/Users/PC/Downloads/cancer_plotly.html) for
more information.

## Analýza úmrtnosti krajín v priebehu času

```

Táto vizuálna reprezentácia zobrazuje trendy v úmrtnosti na rakovinu v rôznych krajinách v priebehu rokov.

```
```{r death_rate_plot, echo=FALSE}

library(ggplot2)
library(gganimate)
library(readr)
cancer_data <- read_csv("C:/Users/PC/Downloads/cancer_data.csv",
 col_types = cols(
 Year = col_double(),
 Country = col_character(),
 `Death Rate` = col_double()
))

p <- ggplot(cancer_data, aes(x = Year, y = `Death Rate`, group =
Country, color = Country)) +
 geom_line() +
 labs(title = 'Death Due to Heart Diseases Over the Years') +
 scale_y_log10() +
 scale_x_continuous(breaks = seq(2000, 2020, by = 1)) +
 theme_minimal() +
 theme(plot.title = element_text(size = 11),
 panel.background = element_rect(fill = "black"),
 panel.grid = element_blank(),
 panel.border = element_blank(),
 axis.line = element_blank(),
 axis.title = element_text(size = 10),
 axis.text = element_text(size = 10))
animated_plot <- p + transition_reveal(Year)
animate(animated_plot)
```

---

## **Ďakujem za pozornosť!**

---
```

Príloha 15

Táto príloha obsahuje kompletný kód použitý na tvorbu *Shiny dashboardu*.

```
library(shiny)
library(tidyverse)
library(plotly)
library(shinydashboard)
library(readr)
library(DT)
library(forecast)
library(readxl)
library(ggplot2)

ui <- dashboardPage(
  dashboardHeader(title = "Traffic Deaths Dashboard"),
  dashboardSidebar(
    tags$head(
      tags$style(
        HTML("
          /* Change sidebar color to black */
          .skin-blue .sidebar {
            background-color: black;
          }
          .skin-blue .main-sidebar {
            background-color: black;
          }

          /* Change font family */
          body, h1, h2, h3, h4, h5, h6 {
            font-family: Arial, sans-serif;
          }

          /* Change font size */
          body {
            font-size: 16px;
          }

          /* Change font weight */
          h1 {
            font-weight: bold;
          }

          /* Change text color */
          h2 {
            color: black;
          }
          .skin-blue .sidebar {
            background-color: #2f2e2f;
          }
          .skin-blue .main-header .logo {
            background-color: #FF8C00;
          }
          .skin-blue .main-header .navbar {
```

```

        background-color: #f9c27a;
    }
    .skin-blue .main-sidebar {
        background-color: #f9c27a;
    }
    .content-wrapper, .right-side, .main-footer {
        background-color: #f9c27a;
    }
    ")
    )
),
sidebarMenu(
    menuItem("Data Overview", tabName = "overview_tab", icon =
icon("table")),
    menuItem("Trend Analysis", tabName = "trend_analysis_tab",
icon = icon("line-chart")),
    menuItem("Data Analysis", tabName = "data_comparison_tab",
icon = icon("exchange")),
    menuItem("Yearly Comparison", tabName =
"yearly_comparison_tab", icon = icon("bar-chart")),
    menuItem("Prediction", tabName = "prediction_tab", icon =
icon("chart-line"))
)
),
dashboardBody(
    tabItems(
        tabItem(tabName = "overview_tab",
            fluidRow(
                box(DTOutput("data_overview"), width = 12)
            )),
        tabItem(tabName = "trend_analysis_tab",
            fluidRow(
                box(title = "Select Countries",
                    selectInput(inputId = "selected_countries",
                        label = "Countries",
                        choices = NULL,
                        multiple = TRUE,
                        selected = NULL),
                    width = 4),
                box(plotlyOutput("trend_plot"), width = 8)
            )),
        tabItem(tabName = "data_comparison_tab",
            fluidRow(
                box(title = "Country Selection",
                    selectInput(inputId = "country_1",
                        label = "Country 1",
                        choices = NULL,
                        selected = NULL),
                    selectInput(inputId = "country_2",
                        label = "Country 2",
                        choices = NULL,
                        selected = NULL),
                    sliderInput("comparison_years_range",
                        "Select Year Range:",

```

```

        min = 2000, max = 2020,
        value = c(2000, 2020),
        step = 1),
      width = 4),
    box(plotlyOutput("comparison_plot"), width = 8)
  )),

  tabItem(tabName = "yearly_comparison_tab",
    fluidRow(
      box(title = "Year Selection",
        sliderInput("selected_years_range",
          "Select Year Range:",
          min = 2000, max = 2020,
          value = c(2000, 2020),
          step = 1,
          post = ""
        ),
        width = 4),
      box(plotlyOutput("yearly_comparison_plot"), width
= 8)
    )),

  tabItem(tabName = "prediction_tab",
    fluidRow(
      box(title = "Predictive Model Input",
        selectInput(inputId = "predict_country",
          label = "Select Country",
          choices = NULL,
          selected = NULL),
        numericInput(inputId = "num_years",
          label = "Number of Years to
Forecast",
          value = 5),
        actionButton("predict_btn", "Predict"),
        width = 4),
      box(title = "Prediction Result with Linear Trend",
        plotlyOutput("prediction_plot"),
        width = 8)
    ))
  )
)
)
server <- function(input, output, session) {

  traffic_deaths_data <- read_excel("C:/Users/Simona
Buchancová/Downloads/traffic_deat.xlsx")

  observe({
    updateSelectInput(session, "selected_countries",
      choices =
sort(unique(traffic_deaths_data$Country)))
    updateSelectInput(session, "country_1",
      choices =
sort(unique(traffic_deaths_data$Country)))
    updateSelectInput(session, "country_2",

```

```

        choices =
sort(unique(traffic_deaths_data$Country))
      updateSelectInput(session, "predict_country",
        choices =
sort(unique(traffic_deaths_data$Country))
      })

output$data_overview <- renderDT({
  datatable(traffic_deaths_data, options = list(pageLength =
10))
})

output$trend_plot <- renderPlotly({
  req(input$selected_countries)

  filtered_data <- traffic_deaths_data %>% filter(Country %in%
input$selected_countries)

  plot_ly(data = filtered_data, x = ~Year, y = ~Traffic_deaths,
color = ~Country,
        type = 'scatter', mode = 'lines+markers') %>%
  layout(title = "Traffic Deaths Trend by Country",
        xaxis = list(title = "Year"),
        yaxis = list(title = "Traffic Deaths"))
})

output$comparison_plot <- renderPlotly({
  req(input$country_1, input$country_2,
input$comparison_years_range)

  filtered_data <- traffic_deaths_data %>%
  filter(Year >= input$comparison_years_range[1], Year <=
input$comparison_years_range[2],
        Country %in% c(input$country_1, input$country_2))

  plot_ly(data = filtered_data, x = ~Year, y = ~Traffic_deaths,
color = ~Country,
        type = 'bar', barmode = 'group') %>%
  layout(title = paste("Traffic Deaths Comparison: ",
input$country_1, " vs ", input$country_2),
        xaxis = list(title = "Year"),
        yaxis = list(title = "Traffic Deaths"))
})

output$yearly_comparison_plot <- renderPlotly({
  req(input$selected_years_range)

  filtered_data <- traffic_deaths_data %>%
  filter(Year >= input$selected_years_range[1], Year <=
input$selected_years_range[2]) %>%
  group_by(Year) %>%
  summarize(Total_Deaths = sum(Traffic_deaths))

  plot_ly(data = filtered_data, x = ~Year, y = ~Total_Deaths,
        type = 'bar', marker = list(color =
'rgba(50,171,96,0.7)')) %>%

```

```

        layout(title = "Yearly Traffic Deaths Comparison",
              xaxis = list(title = "Year", tickformat = "",
                           tickvals = unique(filtered_data$Year)),
              yaxis = list(title = "Total Traffic Deaths"))
    })
    output$prediction_plot <- renderPlotly({
      req(input$predict_country, input$num_years)

      country_data <- filter(traffic_deaths_data, Country ==
input$predict_country)

      if (nrow(country_data) >= 1) {
        y <- country_data$Traffic_deaths

        if (length(y) > 1) {
          start_year <- min(country_data$Year)

          skoda <- ts(y, frequency = 1, start = c(start_year, 1))
          fit <- tslm(skoda ~ trend)
          forecasted_values <- forecast(fit, h = input$num_years)

          forecast_data <- data.frame(Year = seq(from = start_year,
by = 1, length.out = length(y) + input$num_years),
                                     Deaths = c(y,
forecasted_values$mean),
                                     Lower_CI = c(rep(NA,
length(y)), forecasted_values$lower[,2]),
                                     Upper_CI = c(rep(NA,
length(y)), forecasted_values$upper[,2]))

          p <- ggplot(data = forecast_data, aes(x = Year, y =
Deaths)) +
            geom_line(color = "black") +
            geom_ribbon(aes(ymin = Lower_CI, ymax = Upper_CI), fill
= "blue", alpha = 0.4) +
            labs(title = paste("Predicted Traffic Deaths for",
input$predict_country),
                 x = "Year", y = "Traffic Deaths")

          ggplotly(p)

        } else {
          showNotification("Insufficient data points for
predictions.", type = "warning")
        }
      } else {
        showNotification("No data available for the selected
country.", type = "error")
      }
    })
  }

shinyApp(ui, server)

```