

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Vojtěch Burian



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## VIZUALIZACE VÍCEROZMĚRNÝCH DAT POMOCÍ WEBOVÝCH TECHNOLOGIÍ

VISUALIZATION OF MULTIDIMENSIONAL DATA USING WEB TECHNOLOGIES

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Vojtěch Burian**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Zdeněk Mžourek**

**BRNO 2016**

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Vojtěch Burian

**ID:** 145978

**Ročník:** 2

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Vizualizace vícerozměrných dat pomocí webových technologií

**POKYNY PRO VYPRACOVÁNÍ:**

Při analýze jakýchkoliv dat a následném vyhodnocení výsledků je vizuální prezentace výsledků jako jsou například grafy velmi důležitou pomůckou. Ve většině případů je k vytváření těchto analýz potřeba speciální software, které je potřeba odpovídajícím způsobem ovládat. To stejné platí i jen pro úpravy jako je přiblížení či změna jemnosti mřížky grafu.

Cílem této práce je využít možnosti moderních webových prohlížečů pro sestavení interaktivních vizualizací poskytnutých výsledků analýz biomedicínských dat.

Použit bude buď jazyk Python nebo R za využití knihoven jako je například Bokeh nebo Shiny.

**DOPORUČENÁ LITERATURA:**

[1] TUFTE, Edward R. The Visual Display of Quantitative information. 2nd edition. Cheshire, CT: Graphics Press, 2001. ISBN 978-0961392147.

[2] ROJAS G., Sergio J. a Francisco J. BLANCO-SILVA. Learning SciPy for Numerical and Scientific Computing. 2nd edition. Packt Publishing, 2015. ISBN 1783987707.

**Termín zadání:** 1.2.2016

**Termín odevzdání:** 25.5.2016

**Vedoucí práce:** Ing. Zdeněk Mžourek

**Konzultant diplomové práce:**

**doc. Ing. Jiří Mišurec, CSc., předseda oborové rady**

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Práce se věnuje problematice vizualizace dat. Vizualizace dat je užitečným nástrojem, jak prezentovat a získat nové informace, a tak dojít k lepším rozhodnutím. V teoretické části je rozebírána problematika analýzy dat. Dále jsou uvedeny významy a použití jednotlivých typů grafů a v další části je poukázáno na problematiku tvorby grafů. Jsou zmíněny základy grafického designu a tvorby webových stránek. Praktická část se věnuje vizualizaci dat ze zpracovaných výsledků analýzy řeči u pacientů s Parkinsonovou nemocí. Protože většina lidí v odborně zaměřených odvětvích, jako je například lékařenský průmysl, nemá nebo neumí pracovat se specializovaným softwarem (například program Matlab), byly vytvořeny, pomocí programovacího jazyku Python, výstupy ve formátu HTML tabulka a SVG. Obě tyto části jsou zapracovány do webové stránky, kterou lze jednoduše otevřít ve webovém prohlížeči, který je již nainstalován v drtivé většině počítačů, a to bez závislosti na používané platformě operačního systému.

## **KLÍČOVÁ SLOVA**

Analýza dat, vizualizace dat, Python, grafy, SVG, HTML, internetové prohlížeče, grafický design, typografie.

## **ABSTRACT**

Scope of this work is the problematic of data visualization. Data visualization is a useful tool to present and gather new information and thus get to better decisions. In theoretical part, data analysis topics are dealt with. Then specific types of graphs are listed and explained, and in next part problems with graph creation are pinpointed. Basics of graphic and web design are also mentioned. Practical part is focused on visualization of data of processed results of speech analysis, gathered from patients with Parkinson disease. Because majority of people in medical industry do not have, or is not able to work with specialized software (such as Matlab), outputs in HTML table and SVG format were created based on Python programming language. Both these parts are accommodated into webpage, which can be easily opened in web browser installed in most of computers regardless of operation system used.

## **KEYWORDS**

Data analysis, Data Visualization, Python, Graphs, SVG, HTML, Internet Browser, Graphic Design, Typography.

BURIAN, V. *Vizualizace vícerozměrných dat pomocí webových technologií*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, , 2016. 57 s. Vedoucí diplomové práce Ing. Zdeňek Mžourek.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vizualizace vícerozměrných dat pomocí webových technologií“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Mé poděkování patří Ing. Zdeňkovi Mžourkovi, vedoucímu mé diplomové práce, za jeho čas, trpělivost a rady. Také bych rád poděkoval přátelům a také rodině za jejich podporu.

Brno .....

.....

(podpis autora)



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Technická 12, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených projektem Centrum sensorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

(podpis autora)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

<b>Úvod</b>	<b>12</b>
<b>1 Vícerozměrná analýza dat</b>	<b>13</b>
1.1 Využití vícerozměrné analýzy dat . . . . .	13
1.2 Parametrická a neparametrická vícerozměrná statistika . . . . .	14
1.3 Možné problémy dat a jejich řešení . . . . .	14
1.3.1 Standardizace dat . . . . .	14
1.4 Mnohorozměrné normální rozdělení . . . . .	15
1.5 Asociační koeficienty . . . . .	15
1.6 Shluková analýza . . . . .	15
1.7 Ordinační analýza . . . . .	16
<b>2 Vizualizace dat</b>	<b>17</b>
2.1 Průzkum . . . . .	17
2.2 Vysvětlení . . . . .	17
2.3 Tři hlavní aspekty . . . . .	18
2.4 Programování . . . . .	18
2.5 Interaktivita . . . . .	18
2.6 Možnosti vizualizace dat . . . . .	19
2.6.1 Legenda, popisky . . . . .	21
<b>3 Grafický design</b>	<b>22</b>
3.1 Barva . . . . .	22
3.1.1 Vnímání barev . . . . .	22
3.2 Typografie . . . . .	23
3.2.1 Písmo . . . . .	23
3.3 Inovativní přístup . . . . .	23
3.3.1 Názvy, popisky . . . . .	23
3.3.2 Barva . . . . .	24
3.3.3 Symbolika . . . . .	24
3.3.4 Dodržení formy . . . . .	24
3.3.5 Na co si dát pozor . . . . .	24
3.4 Přehlednost grafů . . . . .	24
3.4.1 Porovnání hodnot . . . . .	25
3.4.2 Spojnicový graf . . . . .	26
3.4.3 Grafika u grafů . . . . .	26
3.4.4 Logaritmické měřítko . . . . .	27



3.4.5	Nepopsané osy . . . . .	27
3.4.6	Zdroj dat . . . . .	27
<b>4</b>	<b>Vývoj webových stránek</b>	<b>31</b>
4.1	EasyPHP . . . . .	31
4.2	Webové prohlížeče . . . . .	32
4.2.1	Google Chrome . . . . .	32
4.2.2	Internet Explorer . . . . .	32
4.2.3	Mozilla Firefox . . . . .	32
4.2.4	Safari . . . . .	32
4.2.5	EDGE . . . . .	33
4.3	HTML . . . . .	33
4.3.1	JavaScript . . . . .	33
4.3.2	CSS . . . . .	33
<b>5</b>	<b>SVG</b>	<b>34</b>
<b>6</b>	<b>Python</b>	<b>35</b>
6.1	Continuum Anaconda . . . . .	35
6.2	PyCharm . . . . .	36
<b>7</b>	<b>Aplikovaná část</b>	<b>37</b>
7.1	Vstupní nastavení . . . . .	37
7.1.1	Pygal . . . . .	38
7.1.2	DataTables . . . . .	38
7.2	Popis aplikace v Pythonu . . . . .	38
7.2.1	Blok načtení . . . . .	38
7.2.2	Blok zpracování . . . . .	39
7.2.3	Blok výstupu . . . . .	40
7.3	Popis kódu HTML . . . . .	40
7.3.1	Import CSS . . . . .	42
7.3.2	Import JavaScriptu . . . . .	42
7.3.3	Provázání SVG s tabulkou . . . . .	42
7.4	Blok problematiky . . . . .	47
<b>8</b>	<b>Závěr</b>	<b>50</b>
	<b>Literatura</b>	<b>51</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>55</b>

Seznam příloh	56
A Obsah přiloženého CD	57

## SEZNAM OBRÁZKŮ

2.1	Vizuální ukázka radarového grafu . . . . .	20
2.2	Vizuální ukázka Treemaps . . . . .	21
3.1	Sloupcový graf: čemu se vyhnout, prezentace problému v rozdílu mezi jednotlivými hodnotami . . . . .	25
3.2	Sloupcový graf: čemu se vyhnout, prezentace problému v rozdílu mezi jednotlivými hodnotami . . . . .	26
3.3	Sloupcový graf: jak to udělat správně, prezentace problému v rozdílu mezi jednotlivými hodnotami . . . . .	28
3.4	Spojnicový graf: čemu se vyhnout, prezentace problému velkého a zbytečného místa pod křivkou . . . . .	28
3.5	Spojnicový graf: jak to udělat správně, prezentace problému velkého a zbytečného místa pod křivkou . . . . .	29
3.6	Kruhový graf: čemu se vyhnout, prezentace problému špatně zvoleného kontrastu . . . . .	29
3.7	Spojnicový graf: jak to udělat správně, prezentace problému velkého a zbytečného místa pod křivkou . . . . .	30
3.8	Kruhový graf: čemu se vyhnout, prezentace problému 3D grafů . . . . .	30
5.1	Vizuální ukázka interaktivního výstupu SVG souboru zobrazeného pomocí webového prohlížeče . . . . .	34
7.1	Vizuální ukázka dat v otevřeném CSV souboru v MS Excel . . . . .	38
7.2	Vizuální ukázka box-plot grafu . . . . .	40
7.3	Zjednodušený diagram aplikace vytvořené v Pythonu . . . . .	41
7.4	Zjednodušený diagram aplikace vytvořené JS . . . . .	43
7.5	Radarový graf: zobrazení všech 70 grafů najednou . . . . .	44
7.6	Vnitřní zobrazení SVG struktury pro definování popisků v legendě . . . . .	45
7.7	Radarový graf se skrytými grafy i popisky . . . . .	46
7.8	Celkový pohled na box-plot graf, radarový graf a tabulku . . . . .	48
7.9	IE detekuje na lokálním spuštění JS. Hláška pro upozornění uživatele. . . . .	49

# SEZNAM TABULEK

2.1	Názorná ukázka tabulky . . . . .	20
-----	----------------------------------	----

# ÚVOD

Žijeme v době, kdy máme snadný přístup k informacím různého druhu a je až alarmující, jak nepřehledné množství informací je okolo nás. Vizualizace dat je dnes oborem rychle rostoucím. Souvisí to právě s rozvojem informačních technologií. Jsme zahlceni daty, a proto je potřeba s daty nejen umět pracovat a analyzovat je, ale dokázat je i prezentovat. Vizualizace dat je nejrychlejším způsobem, jak sdílet informace s ostatními. Vizualizace je proces mapování informací do vizuálů. Tvoříme pravidla, která interpretují data a vyjadřují hodnoty jako vizuální vlastnosti.

Užitečným nástrojem pro uchopení, zjednodušení a vizualizaci velmi složitých dat jsou vícerozměrné statistické metody. Při nesprávném užití těchto metod mohou v případě neznalosti vznikat chybné výsledky, které kvůli složité struktuře dat a komplikovaným výpočtům nejsou ihned zřejmé. Příklady užití vícerozměrných metod je možné najít i v oblasti biologie, ekologie, techniky, kybernetiky, sociologie, ekonomie či marketingu.

U pacientů s Parkinsonovou nemocí se objevují až v 70 % případů problémy s řečí. Při neinvazivní analýze u těchto pacientů na základě řečových signálů je výstupem mnoho různých parametrů popisující vlastnosti pacientovy řeči. Cílem této diplomové práce je popsání základů vizualizace a vytvoření aplikace, která z těchto vstupních dat vytvoří interaktivní výstup, který lze jednoduše otevřít ve webovém prohlížeči, jež je již nainstalován v drtivé většině počítačů bez závislosti na používané platformě operačního systému. Pro vytvoření interaktivních výstupů je použit programovací jazyk Python, který se vyznačuje jednoduchou syntaxí a modulárností. Tento výstup je vytvořen tak, aby byl co nejjednodušší, dobře čitelný a zároveň velmi uživatelsky přívětivý [6].

# 1 VÍCEROZMĚRNÁ ANALÝZA DAT

Každý den máme před sebou různá rozhodnutí, která provádíme tak nějak sami, aniž bychom o tom věděli. Intuitivně tak sbíráme data a analyzujeme je pro naše rozhodnutí tak, abychom se rozhodli co nejlépe. Každá věc kolem nás se dá popsat mnoha způsoby a informacemi a při složitějších úlohách již není tak jednoduché zvládnout toto rozhodování tak jednoznačně.

Bude tedy popsáno využití vícerozměrné analýzy dat a budou uvedeny i příklady. Dále budou popsány možné problémy, které se mohou ve vícerozměrných datech vyskytovat. Vícerozměrné metody nám umožňují znázornit a popsat vícerozměrná data, zjišťovat vztahy mezi jednotlivými proměnnými a mezi danými objekty [12, 13, 26].

## 1.1 Využití vícerozměrné analýzy dat

Každý objekt ve světě kolem nás je vícerozměrný, můžeme vnímat nejen jeho tvar, ale také další charakteristické prvky jako například jeho barvu, chuť atd. Přestože všechny tyto vjemy přijímáme každý den a tvoříme z nich jednotlivé objekty, nedokážeme si je představit zapsané v datové tabulce nebo je popsat jinému člověku. Právě z těchto důvodů využíváme vícerozměrnou analýzu dat, jejíž metody jsou užitečné pro explorativní analýzu složitých dat.

Přestože díky klasické statistice můžeme různými způsoby popsat jednotlivě měřené či pozorované proměnné, při hodnocení velkého množství proměnných je pro nás obtížné vytvořit z jednotlivých výstupů jednoduší obraz, díky němuž pochopíme podstatu. Nástrojem, jež nám pomáhá tento proces usnadnit, je vícerozměrná analýza dat. Společnými body klasických statistických analýz a vícerozměrné analýzy dat je hledání souvislostí a jejich výklad.

Vícerozměrná analýza dat má však i své nevýhody. Pouze v případě existující vazby mezi naměřenými proměnnými lze zjednodušit vícerozměrný problém. Pokud mezi proměnnými vazba neexistuje nebo je velmi slabá, nemá smysl používat vícerozměrné metody [12, 13, 26].

## 1.2 Parametrická a neparametrická vícerozměrná statistika

Podobně jako klasická jednorozměrná analýza se i vícerozměrná statistická analýza řídí stejnými zákonitostmi. Její metody jsou citlivé na předpoklady o rozložení a přítomnosti odlehlých hodnot apod. Analýza hlavních komponent s parametrickou kovariancí či korelací je klasickým příkladem provázanosti. Přestože zbývající data nevykazují žádný vztah, přítomnost odlehlé hodnoty vede k vysoké hodnotě korelace a její významnosti. Při analýze hlavních komponent však v této situaci nejdůležitější faktorová osa ukazuje informaci o přítomnosti odlehlé hodnoty v datech a nepřispívá k pochopení zdrojů variability dat. Vícerozměrné metody však lze považovat za analogické k neparametrickým přístupům klasické statistiky (např. některé shlukovací algoritmy). Při výpočtu vícerozměrných analýz proto musíme věnovat pozornost ověření předpokladů [12, 13, 26].

## 1.3 Možné problémy dat a jejich řešení

Na vstupní data jsou kladeny při různých metodách vícerozměrné analýzy různé požadavky. Všechny metody vyžadují úplné datové matice bez jakýchkoliv chybějících dat. U některých metod si lze dovolit odchylky od normálního rozložení dat, některé však vyžadují jejich mnohorozměrné normální rozložení. Vhodnou transformací dat však lze tento problém vyřešit. V případě řádové odlišnosti různých jednotek měřených proměnných je vhodné je převést na stejné měřítko prostřednictvím standardizace dat.

Je mnoho důvodů, kvůli nimž v datech vznikají chybějící hodnoty. Výpadek měřicího přístroje nebo nezodpovězení dané otázky v dotazníku jsou vhodným příkladem. Měli bychom se snažit zajistit, aby chybějící hodnoty nevznikaly, není tomu ale možné zabránit vždy [12].

### 1.3.1 Standardizace dat

Při standardizaci dat používáme statistiky, jež jsou odvozeny z analyzovaného souboru dat (průměr, maximum, rozpětí, směrodatná odchylka atd.). Tímto způsobem se proměnné převádějí na stejné měřítko a přestává záležet na skutečném rozměru příslušné proměnné. Mezi nejčastější úpravy řadíme centrování a standardizaci směrodatnou odchylkou [12].

## 1.4 Mnohorozměrné normální rozdělení

Předpokladem pro použitelnost mnohých klasických statistických metod a postupů je normální rozdělení sledovaných proměnných. Metody založené na tomto předpokladu mohou využívat kompletní matematický aparát, jež je uschován za danou statistickou metodou. Se získanými řešeními se dobře pracuje a metody jsou velmi snadno pochopitelné. Občas však bývá obtížné předpokládané normální rozložení dodržet. V mnohých oborech tento předpoklad není samozřejmý. Musíme však předpokládat normalitu a předpoklad o jedné normálně rozložené náhodné proměnné lze rozšířit na předpoklad simultánního normálního rozložení dvou a více náhodných proměnných. Z předpokladu vícerozměrného normálního rozdělení vycházejí některé vícerozměrné postupy a metody. Užitečnou aproximací různých jiných simultánních rozdělení může také být vícerozměrné normální rozdělení [12].

## 1.5 Asociační koeficienty

V maticové formě jsou uchovávána a zpracovávána vícerozměrná data a všechny vícerozměrné metody jsou založeny na maticové algebře. Matice  $n$  objektů popsána  $p$  proměnnými je základním vstupem vícerozměrných analýz. Na jejím základě je počítána asociační matice (matice vztahů obsahující asociační koeficienty, jež jsou měřítkem podobnosti nebo vzdálenosti dvojice objektů nebo proměnných).

Nejčastěji se pro vazbu parametrů využívá korelace a kovariance. Tímto způsobem vzniklá asociační matice parametrů je podkladem pro faktorovou analýzu a analýzu hlavních komponent. Jako měřítko vztahu pro objekty používáme metriky vzdálenosti či koeficienty podobnosti. Své maximální hodnoty nabývají míry podobnosti v případě identických objektů. Minimální hodnoty nabývají v případě odlišnosti dvou objektů. Obráceně je tomu u vzdálenosti. Podobnost lze případně převést na vzdálenost [12, 13, 26].

## 1.6 Shluková analýza

Informace obsažené ve vícerozměrných pozorováních můžeme využít například roztríděním objektů do několika homogenních skupin (objekty v jedné skupině jsou si podobnější než objekty z různých skupin). Touto tvorbou homogenních skupin objektů se zabývá shluková analýza.

Shlukovou analýzu je vhodné využít zejména tehdy, když se studovaný soubor reálně rozpadá do tříd. Objekty v tomto případě mají tendenci seskupovat se do přirozených shluků. Díky použití vhodných algoritmů můžeme odhalit strukturu



studované množiny objektů a ty poté klasifikovat. Pro popsání rozkladu poté zbývá najít vhodnou interpretaci a charakterizovat vzniklé shluky (skupiny, třídy). Shluková analýza pracuje s asociační maticí podobnosti vzdáleností objektů. Při výběru asociačního koeficientu musíme brát v úvahu metodu shlukování a charakter souboru dat. Způsob výpočtu podobnosti a vzdálenosti je v určitých případech dán konkrétní shlukovací metodou [12, 13, 26].

## 1.7 Ordinační analýza

Při řešení množství problémů se setkáváme se situací, kdy je počet sledovaných proměnných velmi rozsáhlý, nepřehledný a vztahy mezi nimi jsou velmi těžko interpretovatelné. Ordinační je obecné označení pro skupinu metod, které slouží k seřazení objektů podél tzv. ordinační osy tak, aby byl zachován trend a struktura v datech. Ordinační metody umožňují odhalit vztahy mezi proměnnými stejně jako vztahy mezi objekty. Úspěšnost ordinačních metod závisí na struktuře obsažené v datech. Dobře strukturovaná data, tedy data, v nichž existují vztahy mezi proměnnými, umožňují koncentraci podstatné části variability do několika málo ordinačních os.

Rozdíl mezi shlukovou analýzou a ordinací je, že shluková analýza nachází v datech skupiny, klasifikuje objekty nebo proměnné do skupin. Ordinační metody seřazují objekty anebo proměnné podél ordinačních os [13].

## 2 VIZUALIZACE DAT

Vše kolem nás se dá popsat určitými aspekty, jako je například rozměr, velikost a další vlastnosti. Tyto vlastnosti si lze představit jako dimenze. Například jednoduchý spojnicový graf může být dvoudimenzionální. Počítání dimenzí dat může být popsáno jako úroveň složitosti vizualizace. Jak vizualizace začínají být složitější, je více náročné je správně navrhnout a číst z nich. Ačkoliv můžeme narazit na vizualizace s šesti, sedmi nebo i více dimenzemi, nejčastěji jich nebývá více než tři nebo čtyři. Avšak přidání více dat do stejné dimenze nezvyšuje složitost, ale jen objemnost [7, 14, 29].

### 2.1 Průzkum

Průzkum dat je vhodné použít, máme-li velké množství dat a nejsme si jisti jejich obsahem. Chceme-li získat představu o obsahu dat, je potřeba identifikovat jejich vlastnosti.

Nejlépe se průzkum provádí při vysoké úrovni zrnitosti. Data mohou obsahovat hodně šumu, ale při vyřazení mnoha informací hrozí ztráta důležitého obsahu. Tento typ vizualizace je většinou součástí fáze analýzy dat a slouží pro získání představy o obsahu dat [14, 29].

### 2.2 Vysvětlení

Když obsah dat známe a snažíme se předat informace někomu dalšímu, je vhodná vysvětlující vizualizace.

Podle příjemce informací lze upravovat a přizpůsobovat, co je zapotřebí. Jinak řečeno, při editaci je nutno zvolit, která informace je relevantní, a která nikoli.

Je-li průzkumná vizualizace dat součástí analýzy dat, je pak rovněž součástí prezentační fáze. Takovou vizualizaci je možno prezentovat samostatně nebo jako součást obsáhlejší prezentace. V takovýchto prezentacích se objevují verbální nebo psané podporující prvky [14, 29].

## 2.3 Tři hlavní aspekty

Existují tři hlavní kategorie, které vysvětlují vizualizace založené na vztazích mezi třemi důležitými prvky: designér, čtenář a data. Každý z těchto tří prvků má speciální vztah k oběma zbývajícím. Ačkoli je třeba respektovat potřeby a perspektivy všech tří stran v každém vizualizačním projektu, dominantní je nakonec vždy jen jedna [14, 29].

## 2.4 Programování

Zobrazovat data ručně nemusí být zrovna to nejlepší, neboť to může být pomalé a zdolouhavé. Obvykle se proto využívá výpočetní techniky k urychlení tohoto procesu. Zvýšená rychlost nám umožňuje pracovat s mnohem větším množstvím datových souborů tisíců hodnot. Díky výpočetní technice to, co by zabralo i roky úsilí, může být vyobrazeno prakticky ihned. Stejně tak je důležité, že můžeme experimentovat a vylepšovat stávající prezentaci a okamžitě vidět výsledky na výstupu. Tato smyčka (zápisu a opětovného zobrazení) je rozhodující pro opakující se proces vylepšování designu.

V dnešní době se k zobrazení vizuálního výstupu používají počítače. Naším úkolem je tvořit pravidla systému, která jsou poté zpracována softwarem.

Software je jako takový, včetně počítačů, zatím na špatné úrovni pochopení toho, co přesně lidé chtějí, a to proto, že počítače jsou binární. Tím, že počítače nepřijdou napůl cesty směrem k nám, musíme my dojít k nim. A proto je nevyhnutelné učit se, jak správně naprogramovat daný software, přičemž trénujeme komunikaci ve velmi omezené a přesné syntaxi, které počítač rozumí.

Programujeme taky proto, že jde vidět i naše vizuální tvorba. Vizualizujeme data, protože je vzrušující vidět právě něco, co vlastně nikdo předtím nikdy neviděl [28].

## 2.5 Interaktivita

Statické vizualizace nám nabízejí pouze nedostatečné pohledy na data, proto je potřeba vícenásobné zobrazení. Často je potřeba prezentovat ty samé informace různými perspektivami. Počet rozměrů je omezený, když všechny vizuální elementy musí být prezentovány na jenom povrchu a v jednu chvíli. Představování vícerozměrných datových souborů ve statických obrazech je náročné. Pokud nejsou nutné či žádoucí střídavé pohledy a nejsou potřebné při zveřejňování na statickém prostředku, jako je třeba tisk, je pevný obraz ideální.

Interaktivita je skvělá věc, protože činí data přístupná různorodému publiku, oproti těm, kteří jen hledají či již daná data zkoumají. Interaktivní vizualizace nabízí

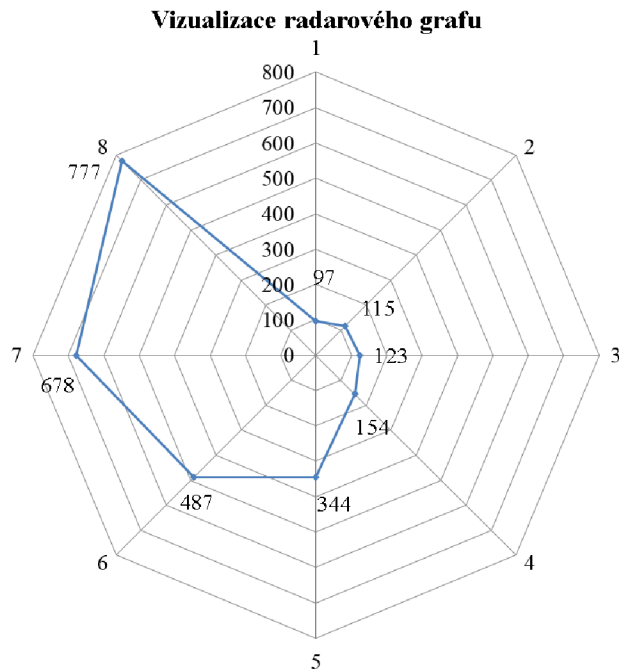
přehled dat společně s nástroji k zobrazení dané informací (detailu). Tím pádem se v dané vizualizaci může pohybovat široké spektrum lidí, od nováčků až po ty, co jsou již s daty důvěrně seznámeni.

V animovaných přechodech a dobře vytvořených rozhraních se může zdát zkoumání dat spíše jako hraní hry, a to díky některým vizualizacím. Interaktivní vizualizace může být skvělou volbou pro ty, kteří si nemyslí, že studium je zrovna zábava [28].

## 2.6 Možnosti vizualizace dat

Máme mnoho druhů, některé jsou méně známé, a jiné zase dobře známé jako například sloupcový graf. Každá vizualizace dat do grafické podoby má určité výhody a nevýhody [24, 29].

- **Graf sloupcový** – Graf sloupcový (viz graf 3.1) patří mezi nejčastěji používané grafy všeobecně. Tyto grafy jsou nejčastěji a nejvhodněji používané k porovnání hodnot. Sloupcový graf a řádkový graf je to samé, jen sloupcový je s vertikálním znázorněním hodnot, a řádkový je s horizontálním. Sloupce nebo řádky mohou mít jinou hodnotu v daném porovnání. Problematika těchto grafů je uvedena v oddíle 3.4.1 [24, 29].
- **Histogram** – Histogramy jsou specializované čárové grafy navržené k zobrazení distribuce hodnot skrze celý rozsah. Celková plocha grafu reprezentuje sumu všech prezentovaných hodnot. Nespojitosť hodnot na vodorovné ose může být libovolná [24, 29].
- **Spojnicový graf** – Spojnicový neboli čárový graf (viz graf 3.5) je velmi užitečný pro spojitá data, kdy je nezávislá proměnná na jedné ose a na druhé závislá proměnná. Pro určité množství dat může být přehlednější než sloupcový graf. Speciálním druhem je časový graf. Je to graf, ve kterém horizontální osa reprezentuje čas a vertikální proměnnou hodnotu. Časové hodnoty se zobrazují zleva doprava [24, 29].
- **Kruhový graf** – Ať již kruhový, výsečový nebo koláčový graf, jedná se o jeden a ten samý, kdy jsou porovnány části celku (viz graf 3.7). Avšak kruhový graf není moc přesný [24, 29].
- **Radarový graf** – Radarový graf (viz graf 2.1) má specifikum, že využívá několik (ne jenom dvě) os a všechny jsou spojeny ve středu. Každá hodnota je vynesena na samostatnou osu [24].



Obr. 2.1: Vizuální ukázka radarového grafu

- **Rozptylový graf** – Rozptylové grafy jsou skvělé pro hledání korelace mezi dvěma druhy kvantitativních dat nebo pro zobrazení dat, která jsou závislá na dvou proměnných. Tří- a vícerozměrná data je možné zobrazit třeba v kruhových grafech [24, 29].
- **Tabulka** (viz tab. 2.1) – I když se to nezdá, tabulky patří mezi prezentaci dat taktéž. Jsou použity tam, kde je potřeba velká přesnost, protože je plníme textovými poli. Efektivní vizualizaci u tabulky může napomoci například rozlišení určitých hodnot pomocí barev [24, 29].

Tab. 2.1: Názorná ukázka tabulky

	Hodnota 1	Hodnota 2	Hodnota 3	Hodnota 4	Hodnota 5
X	193,05	190	189,1	194	194,21
Y	193,00	196,28	198	198,26	194,44

- **Treemaps** – Jedná se o vizualizaci (viz obr. 2.2) proporcionálních dat a hierarchických vazeb v jeden a tentýž čas. Proto jsou výborné pro vizualizaci hierarchických dat. Jednoduše řečeno se jedná o mapu s různě velkými obdélníky, které jsou seřazeny podle určitého specifika. Časté využití této mapy bývá třeba v počítači při hierarchii uložených dat [24, 29].



Obr. 2.2: Vizuální ukázka Treemaps

- **Teplotní mapa** – Je dvourozměrná mapa, která využívá změnu barev nebo jasů k indikování hodnot. Tato metoda se často používá třeba v prezentaci počasí, kde je zobrazena mapa s teplotami [24, 29].
- **Mozaikový graf** – Mozaikový graf je složen ze sloupců, které jsou na sebe naskládány ve dvou směrech, čímž tvoří velký obdélník. Šířka každého sloupce se liší, takže jeho povrch je jen určitá část celku. Suma všech spojených sloupců dohromady dává 100 % [24, 29].
- **Box-plot/krabicový graf** – Box-plot neboli krabičkový diagram je grafické znázornění k odhalení a zobrazení statistických souborů dat, které umožňuje znázornit hodnoty maxima a minima, dolního a horního kvartilu, identifikovat odlehlé hodnoty, popř. posoudit symetrii [23, 30].

### 2.6.1 Legenda, popisky

Pokud se již jedná o pomoc při vizualizaci legendou nebo jednotlivými popisky daných hodnot, musí na ně být kladen důraz, aby byly nerušivého charakteru. Tyto pomocné informace nesmí být daleko od dat, aby čtenář nemusel odskakovat a byly hned zřejmé. Popisky přímo ve vizualizaci jsou vhodné, pokud nemáme ve vizualizaci mnoho hodnot. Jinak by se mohly stát rušivými elementy [29].

## 3 GRAFICKÝ DESIGN

Pro lepší uchopení sdělované informace je vhodné použití grafické formy. Můžeme docílit efektu, kdy i velmi nezajímavá publikace může dosáhnout velmi dobrých pozitivních ohlasů. Použití správného vizuálního efektu může pomoci k jednoduššímu pochopení sdělení. Pokud by efekt sdělení byl spíše rušivého charakteru, může sice zaujmout, ale také může odlákat pozornost od sdělované informace [10, 18, 40].

### 3.1 Barva

Barvy dokáží vyjádřit emoce a pocity. Musíme mít na paměti, že použití barev v různých vizualizacích nemusí vždy pomoci. Barvy se musí brát s rozumem a musí se používat účelně. Volba barev pro jednotlivé základní prvky dokumentu je tedy velmi důležitá.

Vlastnosti barvy můžeme dělit na tři hlavní části. Jas říká, jak světlá nebo tmavá barva je, sytost říká, jak je barva „čistá“, a odstín neboli tón barvy vzniká vzájemným mícháním různých barev [5, 10, 18].

#### 3.1.1 Vnímání barev

Je zapotřebí věnovat pozornost tomu, jak bude použitá barva vnímaná.

- **Prostorové barvy** se využívají buď pro modelování prostoru, kdy nám teplé barvy, tóny do červené a žluté, připadají blíž, a naopak studené, tóny do modré a zelené, vzdálenější. Dají se tak řadit do „vzdálenějších“ i tmavší barvy, smícháním více černé složky, a naopak světlejší do „bližších“.
- **Teplota barvy** – barvy můžeme rozdělit na tři druhy, jedná se o teplé, studené a neutrální.
- **Symbolika barev** – různé barvy vystihují různé emoce a pocity, avšak stejná barva u různých kultur může vyvolávat jiné pocity a emoce.

Barvy a barevné kombinace dokáží velmi dobře působit a přilákat oko čtenáře, ale může se velmi lehko stát, že se u sebe vyskytnou barvy, které k sobě dobře nepasují, jako je například fialový text na černém pozadí. Na to je dobré si dávat pozor [5, 40].

## 3.2 Typografie

První, čeho si všimneme u dokumentu, je jeho vizuální stránka. Proto chceme docílit estetického a příjemného pohledu na daný dokument. K tomu slouží typografie. Každý uživatel, který vytváří dokument, by měl, aspoň částečně, při sazbě znát a řídit se pravidly typografie, pokud se nejedná o odborný nebo rozsáhlý text, zde by měl být uživatel plně seznámen s danou problematikou [5, 10, 40].

Některé typografické zásady:

- Rozdíl mezi pomlčkou a spojovníkem
- Jak napsat správné datum
- Aplikování nezlomitelné mezery
- Použití správných uvozovek
- A další

Webová typografie se lehce liší od obyčejné sazby dokumentů. Při tvorbě dokumentu na webu je snaha, aby daný dokument měl optické záchytné body a byl stručný a výstižný [10, 18, 40].

### 3.2.1 Písmo

Základní rozdělení písma je na bezpatkové a patkové, kdy bezpatkové písmo je vhodnější pro zobrazení na webových stránkách, kdežto patkové písmo je vhodnější pro tištěné a obsáhlé dokumenty. Patkové písmo je lepší pro dlouhé a rozsáhle odstavce, kdežto bezpatkové pro tituly nebo popisky os. Důležité je i správné vybrání rodiny písma, neboť je nesmyslné do odborné publikace dávat nehodící se font, jako je třeba Comic Sans [18, 29, 10].

## 3.3 Inovativní přístup

Při tvorbě něčeho nového je potřeba prvně promyslet, zda je to správným krokem k tomu, aby čtenář pochopil danou problematiku. Musíme taktéž usilovat o to, aby dané znázornění bylo dobře pochopitelné pro uživatele. Když budeme používat stále stejných a dobou prověřených formátů/vizualizací, můžeme si tím ušetřit spoustu práce a času. Avšak pokud bychom používali pokaždé stejných výstupních formátů, nedošlo by k žádnému zlepšení nebo kroku dopředu [10, 29].

### 3.3.1 Názvy, popisky

Vizualizaci tvoříme přece pro uživatele, abychom sdělili danou informaci. Musíme brát v potaz, že čtenář musí rozumět, není tedy vhodné čtenáři přidávat extra práci doplňováním přebytného či nadbytečného textu [10, 29].



### 3.3.2 Barva

Různé barvy v lidech mohou vyvolávat různé emoce, či vyvolávat různé asociace dané kulturním vývojem. Dále různé kombinace, například červená a modrá, mohou vyvolat silně kontrastní dojem, čehož lze využít při sdělování informací pomocí různých grafů nebo diagramů [10, 29].

### 3.3.3 Symbolika

Je nutné brát v úvahu i to, že důležitým faktorem jsou i tvarové vlastnosti. Kolik odlišných hodnot si dokáží uživatelé zapamatovat a odlišit od sebe? U symbolů včetně barev je nutné vybírat vhodné kombinace. Například postupným vybíráním barev z barevné palety lze docílit dojem postupné změny nějakého stavu [10, 29].

### 3.3.4 Dodržení formy

Tak jako při psaní různých prací nebo dopisů je důležité držet si jistou formu zpracování, tak totéž platí o vizualizaci dat [10, 29].

### 3.3.5 Na co si dát pozor

Čemu se radši vyhnout z hlediska vývoje vizualizace dat.

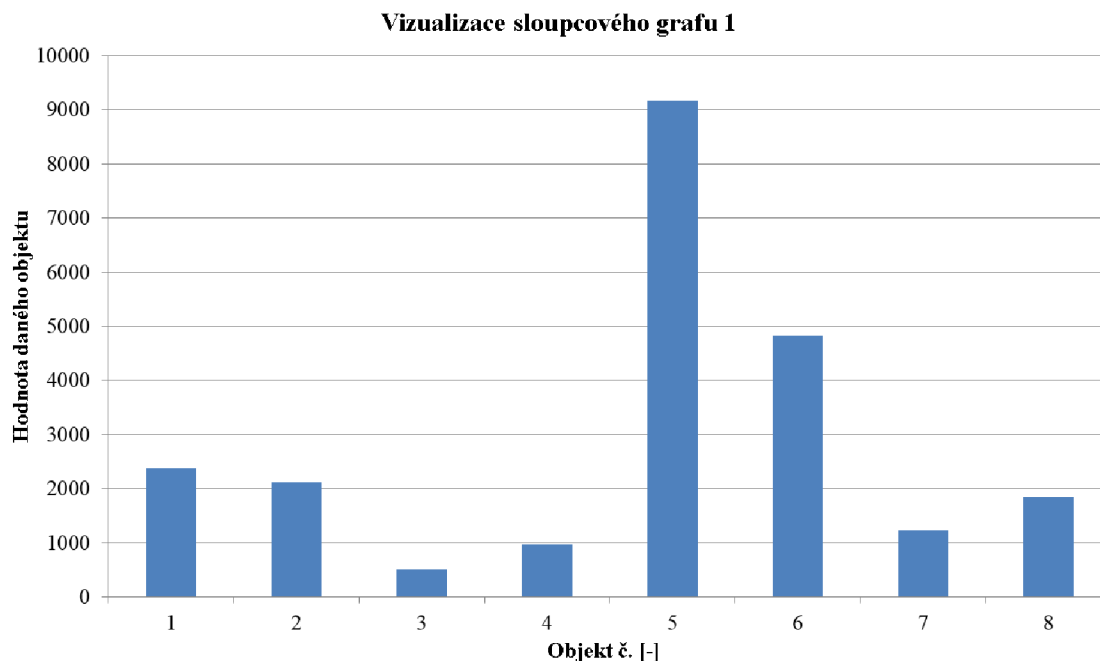
- **Stíny** – Ať už u písma, obrázků, přechodů nebo kdekoli jinde, pokud se nejedná o zvýraznění 3D objektů, jsou stíny zcela nežádoucí stránkou, která může čtenáře jen rušit [10, 29].
- **Barevné přechody** – Přechody přidávají značnou změnu obyčejnému objektu, může změnit dvojrozměrný objekt na trojrozměrný tím, že světlejší část se může jevit blíže a tmavší zase vzdáleněji. Při nesprávném použití může barevný přechod působit čtenáři problémy s pochopením vizuální stránky a významu daného objektu [10, 29].

## 3.4 Přehlednost grafů

Vizualizace dat v grafech jednoznačně zvyšuje přehlednost daných informací. Při větších objemech dat a pokud je potřeba přesnějších hodnot, je jednoznačně lepší tabulka. Abychom ale využili graf při této problematice, je zapotřebí jej vylepšit třeba přidáním popisek, popř. tabulky pod grafem. Důležitou myšlenkou však zůstává fakt, že při tvorbě grafu je zapotřebí, aby byl dobře navržený, a tím pádem plnil svou funkci. Grafy totiž slouží jako efektivní nástroj pro prezentaci dat. K vysvětlení grafu slouží popisky, komentáře a legenda [21, 32].

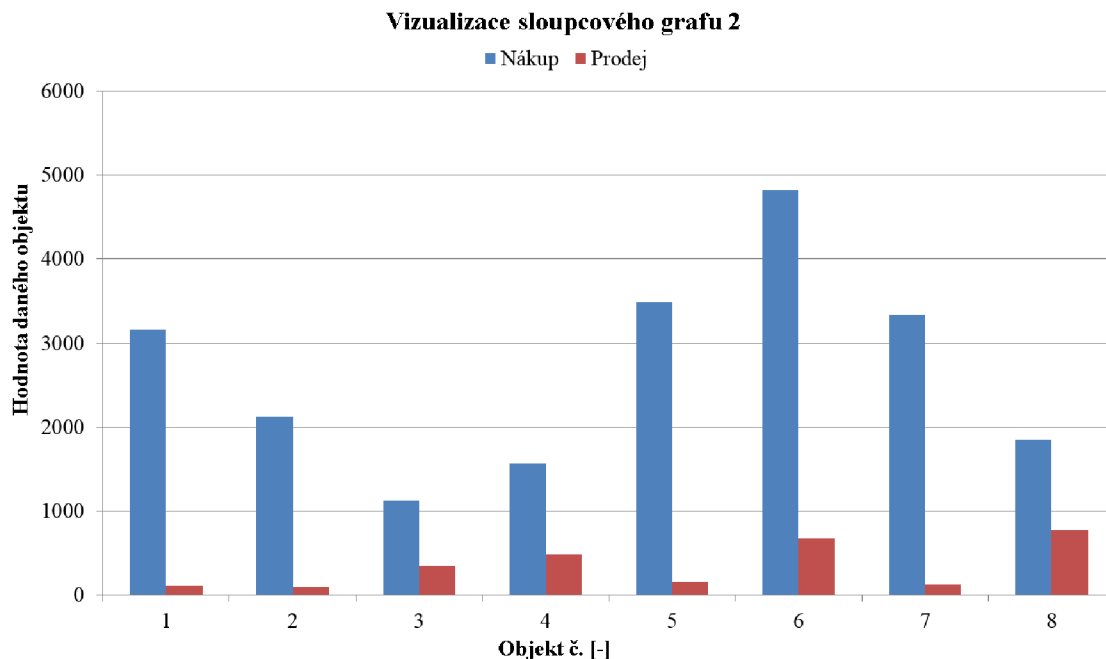
### 3.4.1 Porovnání hodnot

Pro porovnání hodnot byla zvolena vizualizace na sloupcových grafech. Graf se může ze začátku jevit jako v pořádku (viz graf 3.1), ale opak je pravdou. Graf není moc dobře čitelný, je tu jasný převis mezi jednotlivými hodnotami. Velmi dobře zde můžeme vidět rozdíl mezi jednotlivými hodnotami, ale přesnější informace o hodnotách objektu nemáme. Pokud bychom chtěli získat přesnější informace o hodnotách objektu, museli bychom použít přinejmenším lupu a pravítko [21, 32].



Obr. 3.1: Sloupcový graf: čemu se vyhnout, prezentace problému v rozdílu mezi jednotlivými hodnotami

Při porovnávání dvou hodnot v grafu může nastat problém, který vidíme v další prezentaci sloupcového grafu (3.2). Obě hodnoty z něj nelze vyčíst kvůli nepoměru y-os. Proto je výhodnější doplnit si graf o další y-osu ve správném měřítku vůči porovnávaným hodnotám (viz graf 3.3) [21, 32].



Obr. 3.2: Sloupcový graf: čemu se vyhnout, prezentace problému v rozdílu mezi jednotlivými hodnotami

### 3.4.2 Spojnicový graf

Spojnicový graf vypadá na první pohled velmi dobře (viz graf 3.4). Problém je s velmi velkým a zbytečným místem pod křivkou. Je možné, že jako v uvedeném příkladu může dojít ke zkreslení hodnot. Protože většina grafů má volenou osu od nuly, může právě tento případ nastat, což právě u spojnicových grafů nebývá vhodné. Pro estetičtější řešení je dobré posunout začátek osy o určitou hodnotu, jako lze vidět u grafu 3.5. Pro lepší přehlednost je velice vhodné do grafu umístit popisky a značky [21, 32].

### 3.4.3 Grafika u grafů

Nesmíme zapomenout na grafickou vizualizaci grafu, neboť čitelnost je jedna z důležitých funkcí grafu. Musíme brát v potaz to, že kvůli špatně zvolenému grafickému pojetí můžeme celý graf znehodnotit a uvést až do nečitelné podoby (viz graf 3.6). Je potřeba volit správné barvy. Pokud víme, že graf bude zobrazován pouze na monitorech, můžeme využít i méně sytých barev, naopak při promítání za použití dataprojektoru se mohou málo kontrastní barvy jevit stejné či podobného odstínu a budou samozřejmě špatně čitelné (viz graf 3.6). Proto je vhodné volit barvy v kontrastním poměru (například viz graf 3.7) [21, 32].

- **3D grafy** – Mezi další velmi oblíbené grafy patří 3D grafy. 3D grafy jsou sice velice pěkná věc i po vizuální stránce, ale velmi neužitečná, pokud chceme z tohoto grafu získat nějaké přesnější informace. Tyto grafy jsou používány především laickým publikem, které občas nebaví stále pozorovat nudné 2D grafy v prezentacích. U 3D grafu mohou vznikat komplikace jako například zakrytí některých hodnot jinou hodnotu nebo natočením grafu (viz graf 3.8). Tím, že se některé hodnoty nacházejí na vzdálenějších místech, se nám mohou jevit jako nižší, přestože tomu tak nemusí ve skutečnosti být [21, 32].

#### 3.4.4 Logaritmické měřítko

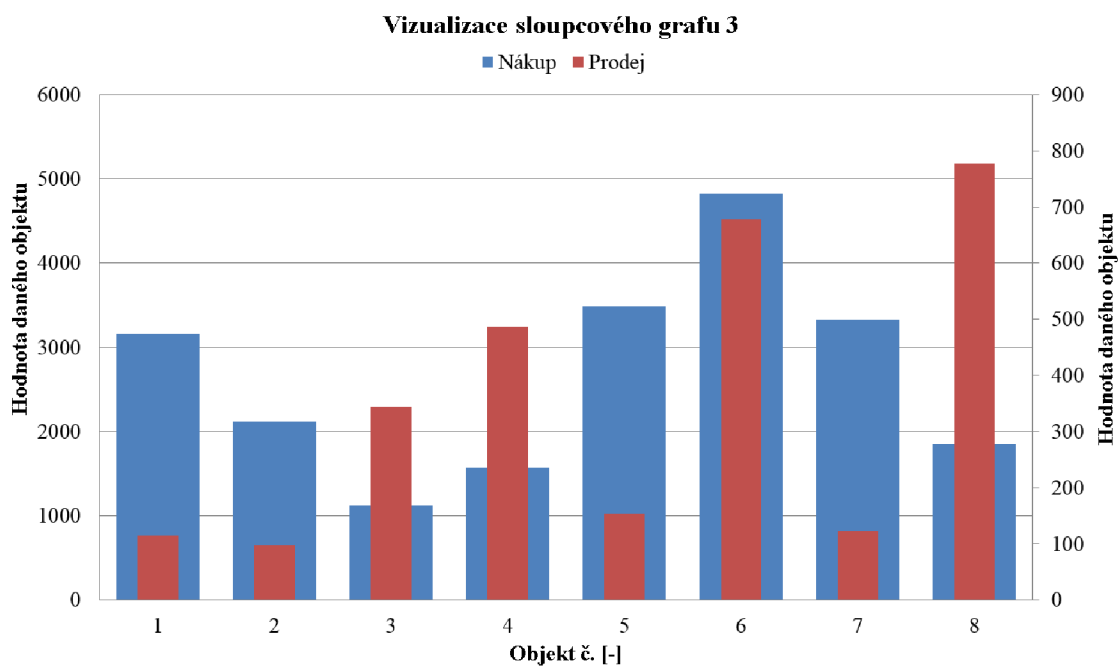
Při použití logaritmického měřítka bychom měli nejprve čtenáře upozornit, že takovýto graf chceme použít. Předjdeme díky tomu možným budoucím komplikacím. Tyto grafy totiž mohou být pro čtenáře komplikovanější, o to více, pokud grafu chybí osy. Logaritmické měřítko je vhodné používat jen tam, kde je to nutné, nejlépe je se mu vyhnout úplně, pokud je graf cílen na laické publikum [21, 32].

#### 3.4.5 Nepopsané osy

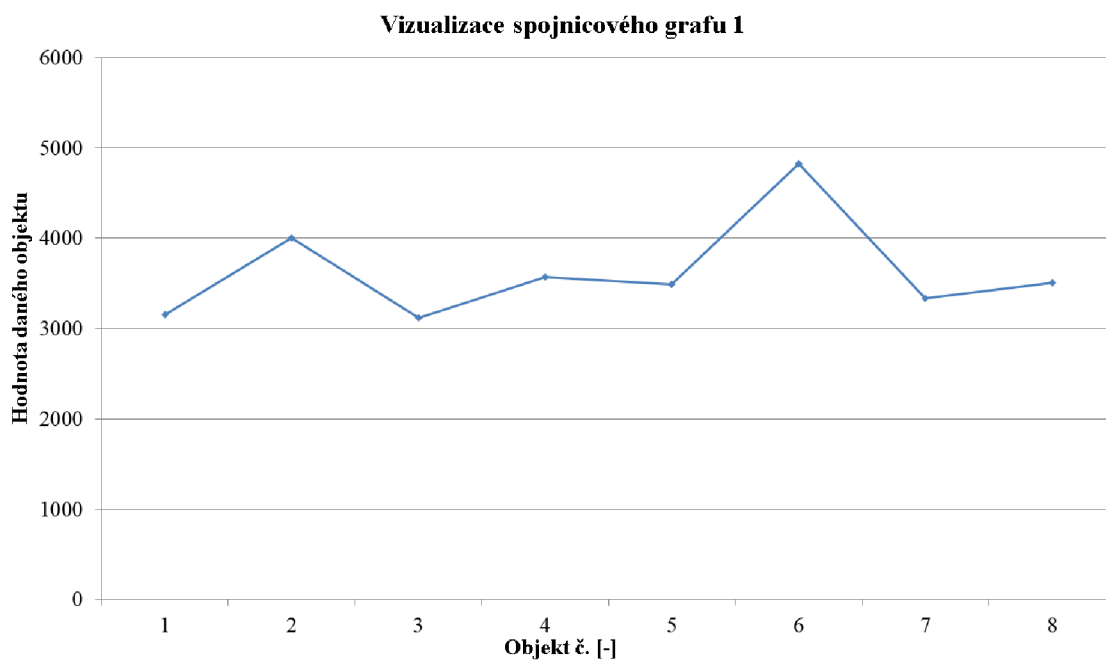
Velice důležitou a často opomíjenou věcí je právě popisování os grafu. I přesto k tomuto opomíjení často dochází. Jedná se snad o nejhorší věc, které se můžete u grafu dopustit. Takový graf totiž nemá skoro žádnou hodnotu, i když existují výjimky. Nepopsaných os může využívat ve svém grafu např. firma, která nechce uveřejňovat konkrétní hodnoty. V takovém případě ale vždy vyvstává otázka, co ji k tomuto kroku vede [21, 32].

#### 3.4.6 Zdroj dat

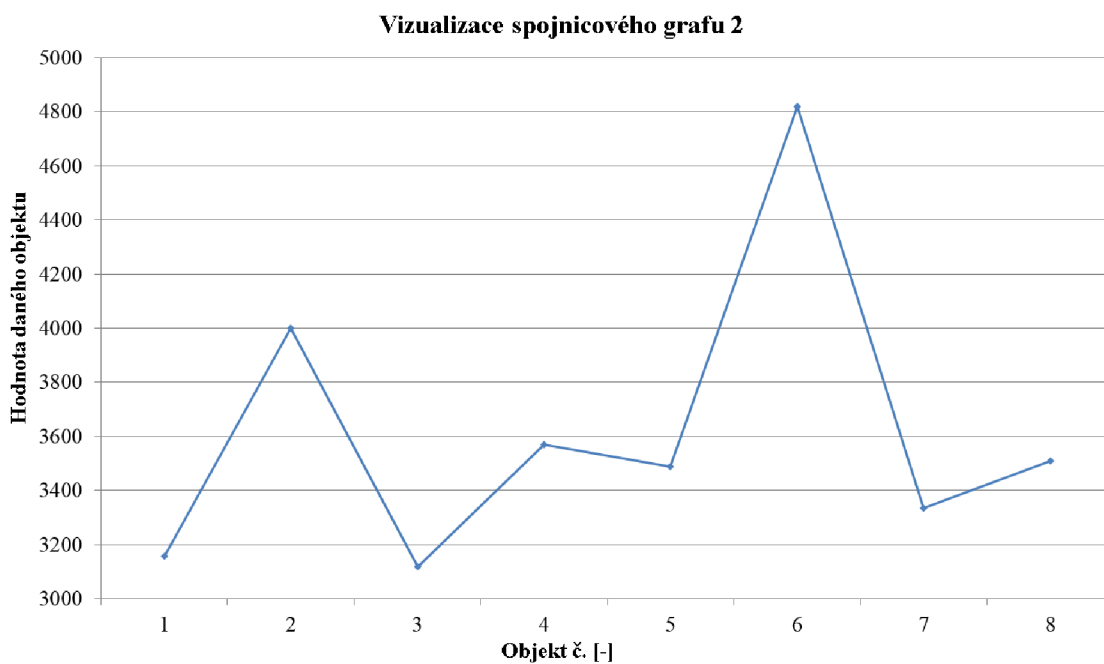
Grafy by měly být objektivním znázorněním dané problematiky, ale bez dohledání zdrojů vznikají mnohdy komplikace. Kupříkladu se může stát, že na takový graf následně vůbec nebude brán zřetel. Problém může nastat, pokud jsou čerpána data z více zdrojů. Pakliže nejsou tyto zdroje správně upraveny, může dojít k situaci, kdy každý zdroj pracuje s odlišnou definicí [21, 32].



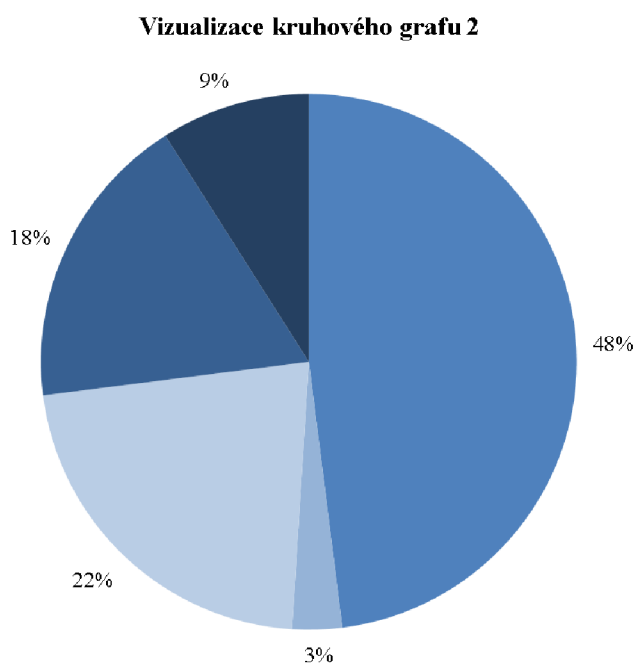
Obr. 3.3: Sloupcový graf: jak to udělat správně, prezentace problému v rozdílu mezi jednotlivými hodnotami



Obr. 3.4: Spojnicový graf: čemu se vyhnout, prezentace problému velkého a zbytečného místa pod křivkou

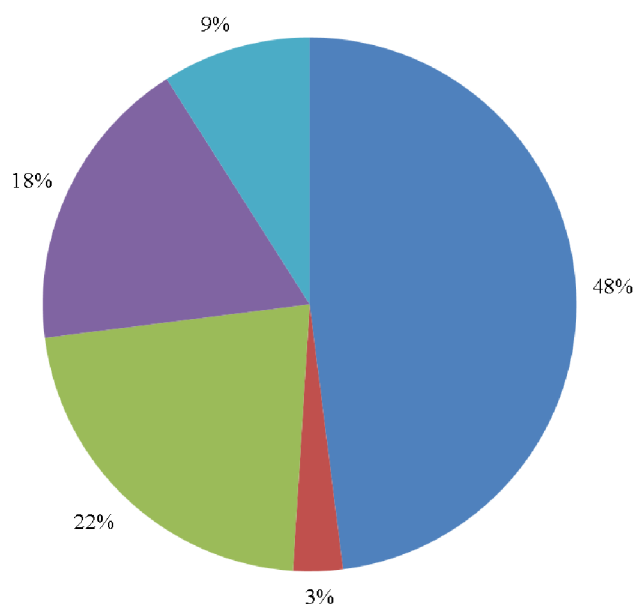


Obr. 3.5: Spojnicový graf: jak to udělat správně, prezentace problému velkého a zbytečného místa pod křivkou



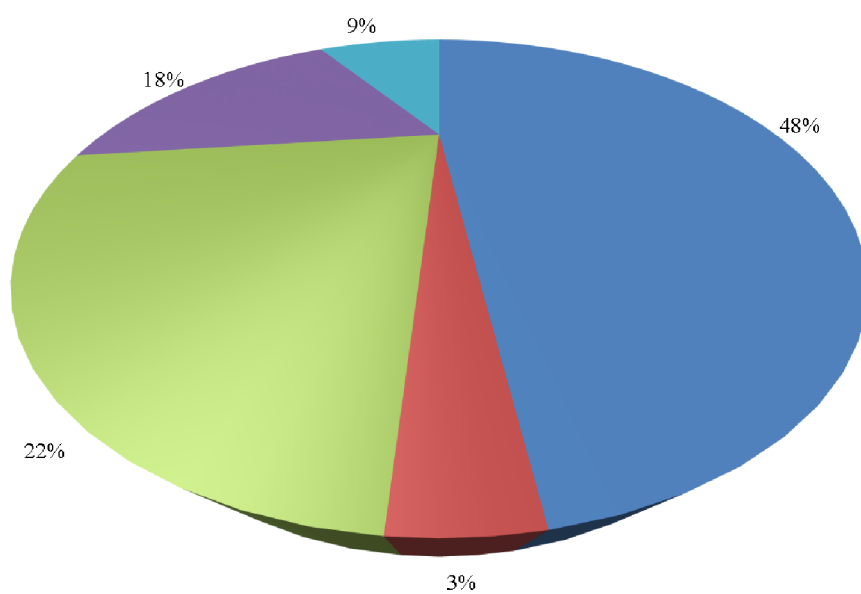
Obr. 3.6: Kruhový graf: čemu se vyhnout, prezentace problému špatně zvoleného kontrastu

**Vizualizace kruhového grafu 1**



Obr. 3.7: Spojnicový graf: jak to udělat správně, prezentace problému velkého a zbytečného místa pod křivkou

**Vizualizace 3D kruhového grafu**



Obr. 3.8: Kruhový graf: čemu se vyhnout, prezentace problému 3D grafů

## 4 VÝVOJ WEBOVÝCH STRÁNEK

Při tvoření jakýchkoliv webových stránek je nutné se nejprve zamyslet nad něčím, nad čím normální uživatelé neuvažují.

Představte si situaci, kdy jdete do obchodu a něco konkrétního hledáte, například jablko. Jako návštěvník oceníte například popisky oddělení, že cenovky patří k danému zboží, a mnoho dalších věcí. U vývoje webových stránek platí totéž. Uživatel se musí cítit dobře.

Aby byly webové prohlížeče schopné zobrazit webové stránky, je nutné, aby tvorba těchto stránek dodržovala určitá pravidla a doporučení.

Hlavními aspekty webu jsou funkčnost a vizuální stránka.

Funkčnost webu nám definuje, pro jakou cílovou skupinu je zaměřen, jak chceme uživatele oslovit a jak s nimi komunikovat. Potom je potřeba znát základní funkce webu jako navigační panel a podobně.

Vizuální stránka nám vlastně tvoří pouze grafický design pro funkcionalitu webu a dodává stránce vzhled. Často design určuje první dojem.

Webové stránky musí být někde spuštěny. V dnešní době se přístup k většině webových stránek zajišťuje pomocí internetu, ale existují taky lokální webové stránky, spuštěné na web serveru pouze pro lokální síť, kdy nejčastější využití je v podnikové sféře.

Internet a webové stránky by se daly popsat jako skupina propojených stránek, které si vyměňují server s klienty.

Web server jsou počítače, na kterých běží specializovaný software připojený do internetu. Díky tomu mohou vytvářet například HTTP server. Nejznámější je Apache a ten je taky nejpoužívanější.

Pro vývoj webu je možnost nainstalovat odlehčené verze webových aplikací na svůj počítač, tudíž máme možnost rychlého a efektivního vývoje webu v lokálních podmínkách, aniž bychom přistupovali na vzdálený stroj se specializovaným software [1, 9, 22, 38, 39].

### 4.1 EasyPHP

Jedná se o souhrnný balík aplikací určený převážně pro vývojáře webových aplikací. Lze v něm nalézt jak server Apache, PHP<sup>1</sup> ale i MySQL<sup>2</sup> [8].

---

<sup>1</sup>PHP – skriptovací jazyk

<sup>2</sup>MySQL – databázový systém



## 4.2 Webové prohlížeče

Jedná se o programy, pomocí nichž lze prohlížet webový obsah. Prohlížeč vystupuje jako klient, který komunikuje s web serverem pomocí protokolu HTTP(S)<sup>3</sup>. V dnešní době musí prohlížeč umět spoustu věcí od zobrazení jednoduchého textu po zobrazení složitých animací (a mnohem více).

Existují samozřejmě mnoho druhů prohlížečů, ale liší se jen uživatelským prostředím, neboť jsou ve většině případů postaveny na vykreslovacích jádrech od níže zmíněných prohlížečů.

Vykreslovací jádro je software, který pomocí HTML, CSS a dalších zobrazí daný obsah webové stránky. Každý internetový prohlížeč má své vykreslovací jádro [16, 36, 42].

### 4.2.1 Google Chrome

Jeden z nejmladších prohlížečů, který se velmi rychle dostal mezi nejpoužívanější prohlížeče na světě. Díky Google a jeho inovacím se tento prohlížeč rychle vyhoupl na výsluní a stal se oblíbeným i u odborníků, a to zejména díky své rychlosti, snadnému ovládní a bezpečnosti [11, 16, 42].

### 4.2.2 Internet Explorer

Dříve nejpoužívanější prohlížeč. Bohužel jeho obliba poslední dobou spíše klesá, neboť po dlouhou dobu byl vývoj spíše přehlížen, čímž prohlížeč značně utrpěl. Oproti ostatním prohlížečům má IE (Internet Explorer) velkou výhodou, neboť je součástí instalace OS Windows. V poslední době se Microsoft snaží dohnat své soupeře na poli internetových prohlížečů vylepšováním podpory tohoto prohlížeče [16, 17].

### 4.2.3 Mozilla Firefox

Tento prohlížeč se i bez zástupce mezi nadnárodními podniky (Google a další) a výraznější propagace dostal mezi nejpoužívanější prohlížeče. Hlavní výhodou Firefoxu je bohatá podpora rozšíření a vývojářská přívětivost. Stále zaujímá značnou část na trhu internetových prohlížečů [16, 42].

### 4.2.4 Safari

Safari je internetový prohlížeč určený především pro Mac OS [42].

---

<sup>3</sup>HTTP – komunikace mezi serverem a klientem, HTTPS – zabezpečená komunikace

## 4.2.5 EDGE

Microsoft se chtěl oprostít od špatné pověsti, která provází IE, a tak stvořil nový prohlížeč s novým jádrem, který uvedl jako novinku v novém operačním systému Windows 10. I přes to všechno se tomuto prohlížeči moc nedaří, neboť mu stále chybí některé technologické standardy, které již dávno podporuje konkurence [16, 17, 42].

Existují různé druhy vykreslovacích jader:

- Gecko – vývoj společností Mozilla
- Trident – vývoj společností Microsoft
- Blink – vývoj společností Google
- a další

## 4.3 HTML

HTML je značkovací jazyk, který je základním jazykem pro tvorbu webových stránek [22].

### 4.3.1 JavaScript

JavaScript je skriptovací jazyk dovolující dynamicky a interaktivně pracovat s HTML a CSS na straně klienta (na rozdíl od PHP). Běžně je psán JavaScript do textového souboru a potom nahrán do webové stránky. Také ale může být přímo psán do zdrojového kódu. Představuje rychlý a jednoduchý způsob, jak si vyzkoušet svůj kód [15, 25].

### 4.3.2 CSS

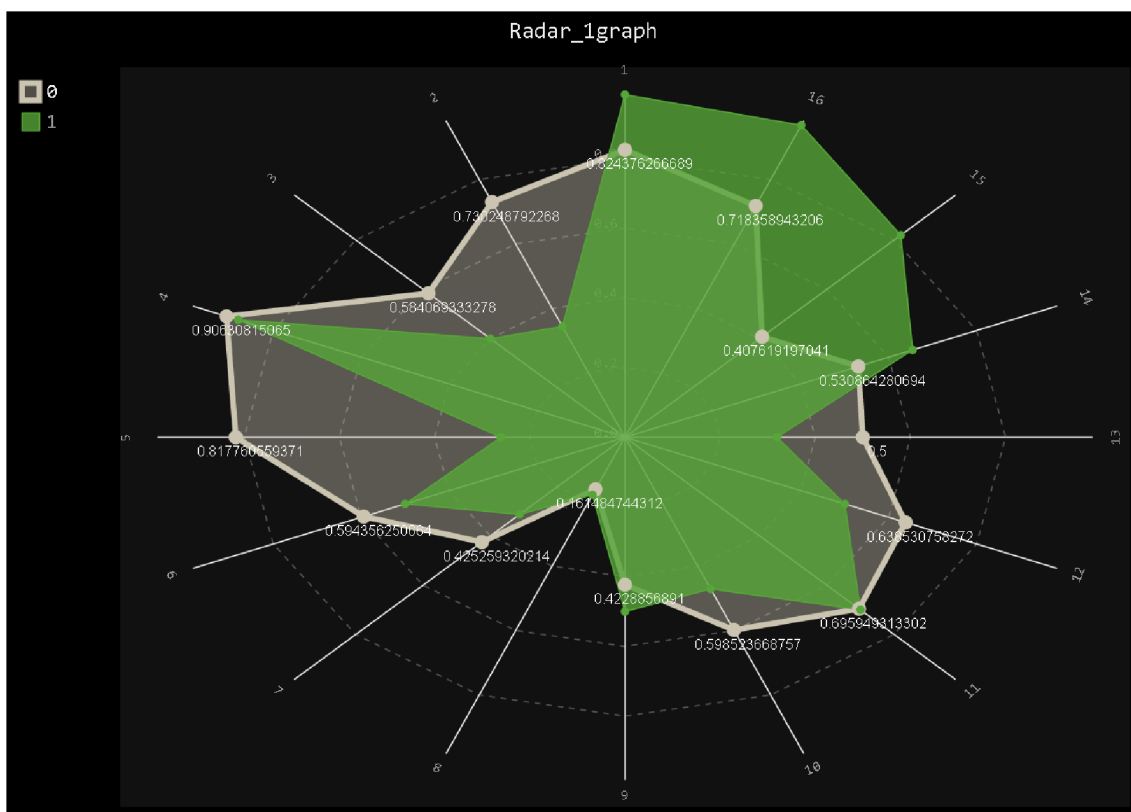
Jedná se o jazyk pro popis způsobu zobrazení webových stránek. Stylopis (soubor CSS) definuje vizuální styl webových stránek pro vykreslení v prohlížeči. Umožňuje vývojáři definovat vzhled od různých prvků až po vzhled celého uživatelského rozhraní. CSS styly se skládají ze selektorů a vlastností. Selektory jsou následovány vlastnostmi, seskupenými ve složených závorkách. Vlastnost a její hodnota je oddělena dvojtečkou a řádek je zakončen středníkem. Samozřejmě mohou stejné vlastnosti být aplikovány na více selektorů najednou oddělením selektorů čárkou [35, 39].

## 5 SVG

Scalable Vector Graphics neboli škálovatelná vektorová grafika je grafický formát, který oproti běžným formátům, jako jsou JPG a další, není popsán mřížkou bodů (rastrové formáty), ale je tvořen pomocí grafických objektů (vektorová grafika), z nichž se dá obrázek vykreslit. Nedochází ke zkreslení při zmenšení nebo zvětšení daného obrázku. Formát SVG je založen na XML. XML umožňuje zjednodušení práce tvorbou vlastních tagů. Jedná se o tzv. nadstavbu HTML (XHTML).

Pomocí JavaScriptu a SVG lze vytvořit velmi zajímavou webovou grafiku (viz obr. 5.1). Hlavními výhodami SVG jsou tedy možnost měnit rozměry bez ztráty dat, lze s ním snadno manipulovat pomocí CSS nebo JavaScriptu a další.

Pro generování SVG souboru je také možné použít grafického software, jako je například Illustrator, nebo kód prostě napsat ručně [37, 39].



Obr. 5.1: Vizuální ukázka interaktivního výstupu SVG souboru zobrazeného pomocí webového prohlížeče

## 6 PYTHON

Jedná se o velmi známý, používaný a rychle se vyvíjející jazyk. Python má zajímavé vlastnosti. Jedná se především o jeho jednoduchou syntaxi a modulárnost. Je beztypovým a skriptovacím jazykem a také objektově orientovaným dynamickým jazykem. Tvůrcem tohoto programovacího jazyka je programátor Guido van Rossum, který vytvořil tento jazyk v roce 1990. Autor pochází z Holandska. Jazyk Python je vlastně následníkem jazyka ABC. Python lze nalézt jako základní implementaci v Linuxu.

Momentálně existují dvě verze Pythonu, a to Python 2.7 a Python 3.5. Tyto verze nejsou mezi sebou kompatibilní. Podpora verze 2.7 již byla ukončena. Rozdíly mezi těmito verzemi jsou hlavně v zápisu syntaxe. Obě verze jsou zmíněny kvůli problematice různých aplikací, které jsou napsány nad 2.7 a stále se vyskytují v linuxových distribucích. Python jako takový má spoustu implementací jako CPython, Jython a další.

Každá implementace se může lišit od základní Python dokumentace. CPython je momentálně nejvíce používanou a nejrozšířenější implementací. V základní verzi můžeme najít seznamy, slovníky, celá čísla a mnoho dalšího. Podporuje objektově orientované programování počítané s vícerozměrnou dědičností. Celý kód je vytvořen z mnoha balíčků. Je možné stále přidávat další funkce nebo jiný objekt, který je obsažený v dalším balíku, a to pomocí volby „importovat daný balík“. Tento jazyk podporuje jak vyvolání, tak zachycení výjimek.

Další vlastností Pythonu je schopnost programově prozkoumat daný objekt a rozhodnout o jeho identitě, vlastnostech a schopnostech. Python se vyznačuje datovými typy, které jsou silně dynamicky typované, a automatickou správou paměti, kdy na rozdíl od jazyku např. C je nutné nejprve manuálně alokovat a uvolňovat paměť. Operace nad nekompatibilním typem způsobí vyvolání výjimky [2, 41].

### 6.1 Continuum Anaconda

Anaconda je platforma skládající se z Pythonu a směsi různých balíčků, které jsou skládány tak, aby vytvořily hezké prostředí pro zpracování dat. Rozdíl mezi Anaconda a Anaconda3 je prakticky v tom, že Anaconda3 používá nejnovější verzi Pythonu, kdežto Anaconda používá verzi 2.7, která je vnímaná spíše jako konzervativní. Krása tohoto řešení spočívá v jeho jednoduchosti, neboť tento program obsahuje obrovské množství balíčků pro práci s jazykem Python [4].

Mezi základní balíky tohoto programu patří například:

- **Knihovna NumPy** – NumPy knihovna se specializuje na numerické zpracování pro Python. Pracuje se zde s  $n$ -rozměrným polem [3, 31].

- **Knihovna SciPy** – Knihovna SciPy je využívána pro vědecké výpočty v programovacím jazyku Python. Jde o open source knihovnu. SciPy knihovna obsahuje velké množství balíčků, které pokrývají velkou část z oblasti matematiky. Tato knihovna umí zpracovávat data z různých prostředí, jako je například Matlab a další [3, 34].
- **OS knihovna** – Tato knihovna zajišťuje funkce operačního systému jako například číst a zapisovat soubory [33].
- **Knihovna CSV** – Jak již název napovídá, je to knihovna, která umí zpracovávat CSV soubory [33].

Knihovna, u které bylo potřeba dodělat import:

- **Knihovna Pygal** – Knihovna Pygal umí vykreslovat dynamické SVG grafy, které jsou napsány v programovacím jazyku Python. Tato knihovna není v základu obsažena, proto je potřeba ji doimportovat [27].

## 6.2 PyCharm

PyCharm ve verzi 5 je vývojové prostředí, které má funkce pro doplňování, nápovědu apod. Toto vývojové prostředí staví na interpretu zvaném IPython. IPython umí daný projekt otevřít přes HTTP rozhraní jménem notebook. PyCharm je vyvíjen firmou JetBrains [20].

## 7 APLIKOVANÁ ČÁST

Kvůli tomu, že většina publika nebude velmi zdatná při ovládní specializovaných programů, jako je například Matlab, a také kvůli problémům s licencováním těchto programů je velmi vhodné vytvoření takového, který dokáže vzít určená data a nechat je danému publiku prezentovat, a to bez znalosti specializovaných programů. Dá se toho docílit tím, že tyto výstupy exportujeme do formátu, který je možné otevřít pomocí webového prohlížeče, jenž se v dnešní době nachází v drtivé většině počítačů. Aplikovaná část je zaměřena na tvorbu interaktivních výstupů pomocí programovacího jazyku Python. Aby mohla být data prezentována pomocí webových technologií, je nejprve nutné vstupní data zpracovat. K této realizaci byl vybrán právě programovací jazyk Python. Jako vývojové prostředí byl zvolen PyCharm (viz sekce 6.2).

V první řadě bylo potřeba nainstalovat Python 3.5 a následně Anacondu. Anaconda už v základu obsahuje základní knihovny jako SciPy nebo NumPy, ale pro tvorbu interaktivních grafů bylo zapotřebí ještě naimportovat další knihovnu, a to knihovnu Pygal.

Na internetu lze dohledat velice snadno návody na instalaci daných knihoven včetně Pythonu a PyCharmu. Všechny použité knihovny, vývojová prostředí nebo skripty jsou buďto open source, nebo free software.

### 7.1 Vstupní nastavení

Jako základní vstup je dána matice, která je výsledkem analýzy řečových signálů u pacientů s Parkinsonovou nemocí (viz obr. 7.1).

Data jsou uložena v souboru CSV. Jedná se o tabulkový formát, který je tvořen sloupci a řádky. Jednotlivé hodnoty jsou odděleny buďto středníkem, nebo čárkou. Existují země, kde je použita u desetinného čísla čárka namísto tečky, a proto je definován znak středník pro oddělení hodnot. Jednotlivé hodnoty jsou uzavřeny do uvozovek. Pokud se v hodnotě vyskytují uvozovky, jsou tyto uvozovky zdvojeny.

V matici se vyskytují sloupce X1 až X16, data v těchto sloupcích nejsou v generovaných tabulkách vyjádřena. Jsou graficky znázorněna v grafech. Jedná se o simulovaná data. Ostatní hodnoty jsou využity pro tvorbu tabulky ve webovém prohlížeči. [19]



- **pygal-tooltips.js** – při najetí kurzorem způsobí zprůhlednění daného grafu a zapínání a vypínání daného grafu pomocí vlevo vytvořeného postranního navigačního systému
- **pygal-tooltips.min.js** – při najetí na daný vrchol v grafu zobrazí jeho přesnou hodnotu

Nepředpokládalo se více vstupů, proto byla zavedena pevná metoda načtení daného souboru.

## 7.2.2 Blok zpracování

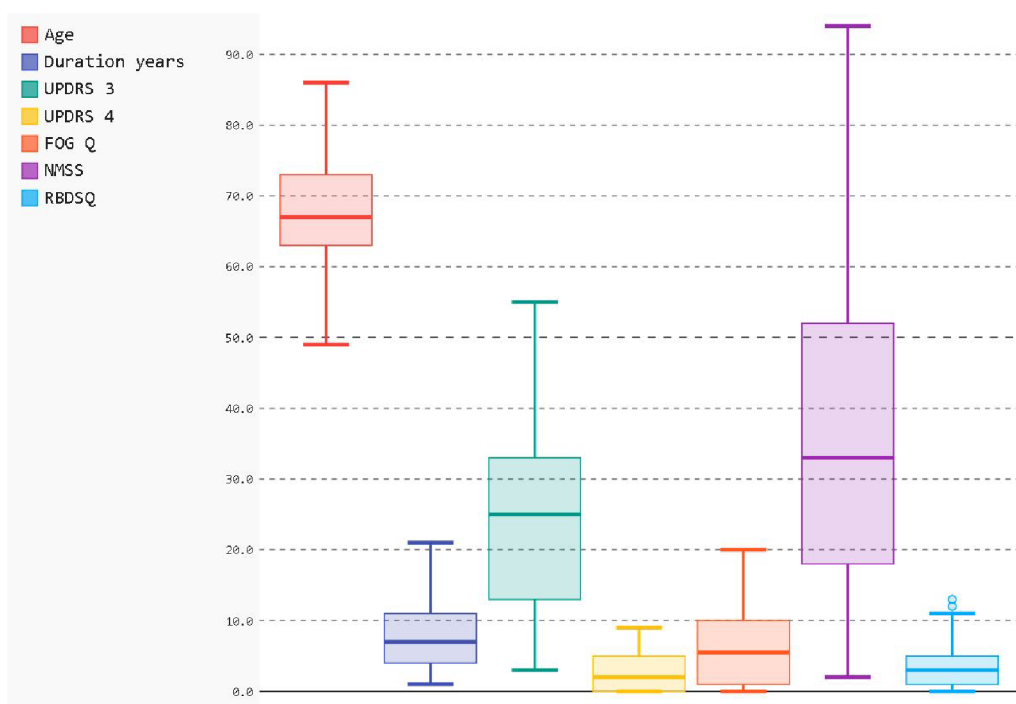
- Nejprve je nutné určení souboru ke zpracování. Program pracuje s jedním vstupním souborem, jehož jméno buď zadá uživatel jako parametr programu nebo se použije název „data.csv“.
- Po načtení a zpracování vstupního souboru je tvořena struktura HTML tabulky. Vytvořený řetězec obsahující hlavičku a patičku je uložen v proměnné *table\_string*.
- *radar\_chart* vytvoří kostru pro radarové grafy. Určí se zde, že se jedná o radarový graf, že se mají používat výše zmíněné JavaScripty, ukázat popisky os a další.
- Pomocí for cyklu jsou tvořeny jednotlivé grafy, jeden odpovídá právě jednomu řádku z hodnot ze sloupců *X1* až *X16* uložených ve vstupním souboru.
- Vzhledem ke struktuře HTML tabulky je potřeba další cyklus, který z dat stejných jako pro tvorbu grafu vytvoří jeden řádek těla tabulky.
- *box\_plot* určí, že se jedná o box-plot graf, že se mají používat výše zmíněné JavaScripty, ukázat popisky os a další.
- *csv\_file.seek(0)* nastavuje čtecí ukazatel souboru na začátek.
- Dále pomocí for cyklu naplníme jednotlivé seznamy v seznamu seznamů prvky ze sloupců, které můžeme definovat, které jsou následně použity pro tvorbu box-plot grafu.
- Z dříve naplněného seznamu seznamů je dalším for cyklem vytvořen box-plot graf.
- Následně jsou uloženy výstupy.



### 7.2.3 Blok výstupu

Výstupem jsou jasně definované soubory (*radar\_grafy.svg*, *box\_grafy.svg* a *tabulka.html*).

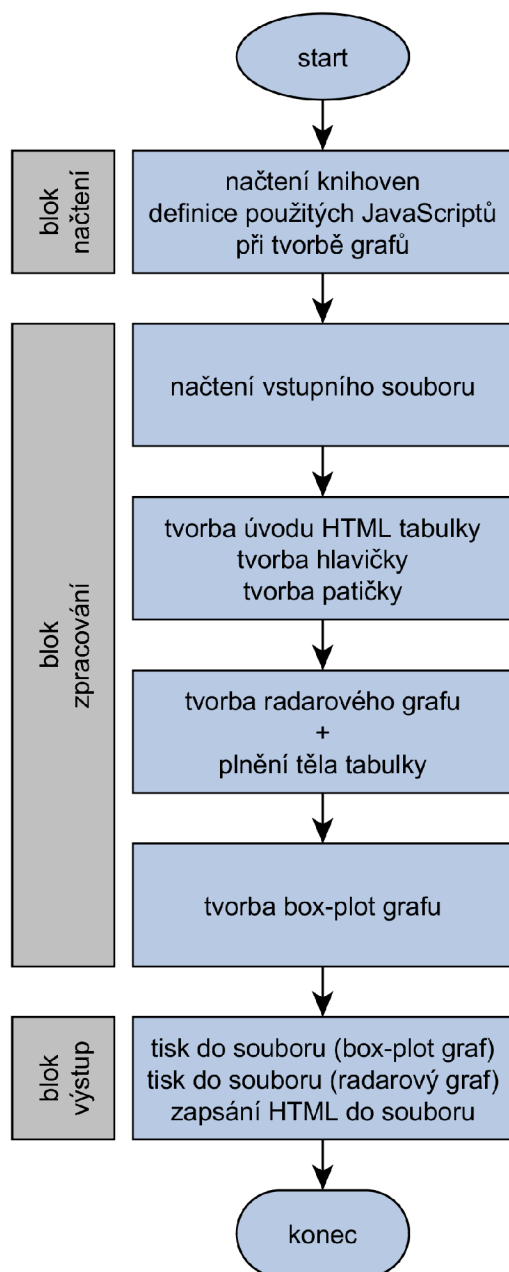
- **tabulka.html** – Pro vytvoření HTML tabulky byly použity všechny sloupce, kromě *X1* až *X16*, které jsou použity při vizualizaci grafů, a prvního sloupce označujícího index.
- **Radar\_grafy.svg** – Díky možnostem knihovny Pygal, jako je dobrá podpora komunity, neustálý vývoj atd., a díky JavaScriptu byl vytvořen jediný interaktivní SVG soubor radarových grafů, překrývajících se přes sebe, ze sloupců *X1* až *X16* ze všech uvedených řádků, kromě řádku obsahujícího názvy. Celkově 70 grafů přes sebe v jednom souboru (viz obr 7.5).
- **Box\_grafy.svg** – Byl vytvořen SVG soubor s box-plot grafy ze sloupců hodnot *Age*, *Duration years*, *UPDRS 3* a *4*, *FOG Q*, *NMSS* a *RBDSQ* (viz obr 7.2).



Obr. 7.2: Vizuální ukázka box-plot grafu

## 7.3 Popis kódu HTML

Vývoj webové stránky byl proveden třemi základními kroky: import CSS, import modulů a skriptů psaných v JS a popis provázání SVG s tabulkou.



Obr. 7.3: Zjednodušený diagram aplikace vytvořené v Pythonu

### 7.3.1 Import CSS

Kaskádové styly, které jsou uvedeny v hlavičce stránky, slouží k definování vizuálního stylu webové stránky po vykreslení v prohlížeči. Jazyk kaskádových stylů CSS umožňuje vývojáři definovat vzhled různých prvků až vzhled celého uživatelského rozhraní prostřednictvím atributů, tříd a identifikátorů [35, 39].

- **jquery.dataTables.css** – hlavní činností tohoto modulu je definování vizualizace HTML tabulky
- **nacitani.css** – vlastní definice stylu celého webu, vizualizace načítání a interakcí s tabulkou

### 7.3.2 Import JavaScriptu

JavaScript dovoluje dynamicky a interaktivně pracovat s HTML a CSS na straně prohlížeče. Toho je využito jak při interaktivní práci s tabulkou, tak při provázání vloženého SVG souboru s tabulkou pro vybírání vhodných grafů podle řádků tabulky [15, 25].

- **html5-3.6-respond-1.1.0.min.js** – umožňuje využití HTML5 ve starších verzích IE
- **jquery.js** – knihovna pro usnadnění práce s HTML DOM<sup>1</sup> objekty
- **jquery.dataTables.js** – modul pro vizualizaci a správné fungování HTML tabulky
- **interní skript na konci index.html** – vlastní skript umožňující interaktivní vazbu SVG s tabulkou

### 7.3.3 Provázání SVG s tabulkou

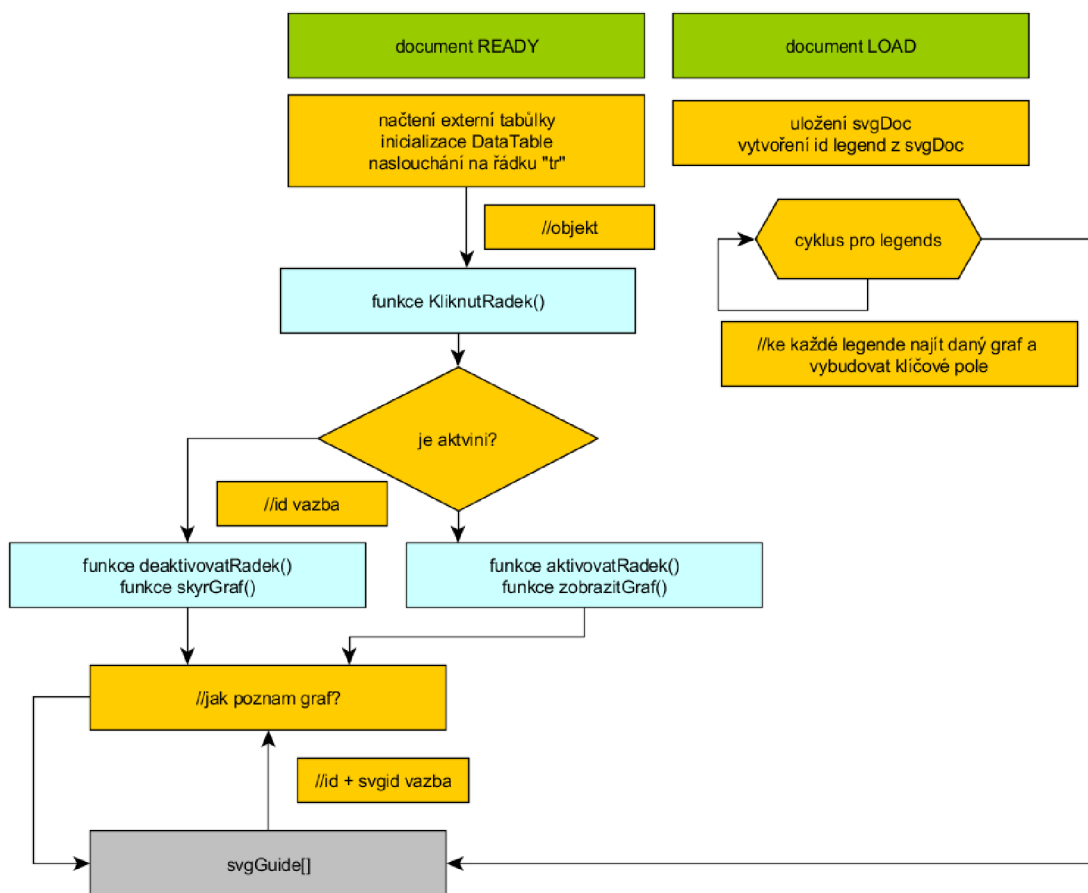
SVG i tabulka jsou funkční, stačí pomocí webového prohlížeče otevřít daný soubor. Bohužel chybí provázání tabulky s daným SVG souborem, a tudíž mezi sebou tyto nekomunikují a SVG soubor nereaguje na práci s tabulkou. Proto bylo potřeba dopsat vlastní interní skript reagující na akce prováděné v tabulce, aby se projevil ve vloženém SVG. Daný kód se nachází v hlavním dokumentu *index.html*, je spuštěn po inicializaci ostatních skriptů, aby reagoval na různé stavy načítání.

Skript má definované dva bloky: *LOAD* a *READY*. Pro zjednodušený náhled viz obr. 7.4.

- V první části *READY* se načte tabulka (*tabulka.html*) a aplikuje se modul *DataTables*, kde jsou registrovány kliknutí na řádky, objekt kliknutého řádku je předán vlastním funkcím.

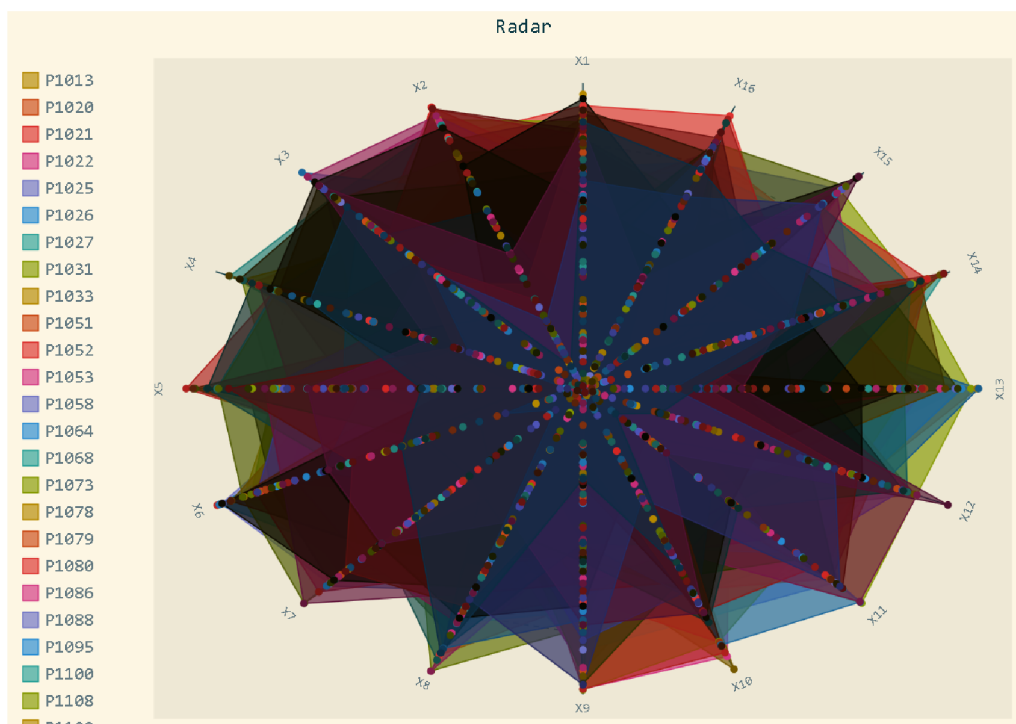
---

<sup>1</sup>DOM – model umožňující práci s objekty



Obr. 7.4: Zjednodušený diagram aplikace vytvořené JS

- Pokud dojde ke kliknutí na řádek, zavolá se funkce *klinutRadek()*, která provede zjištění, zda řádek v tabulce, na který bylo kliknuto, je aktivní či naopak, podle toho jsou pak volány další funkce.
- Při zpracování daného řádku je hledán první sloupec, je definovaný tagem `<td>` a obsahem jsou popisky (např. *P1013*). Vezme se tedy obsah tohoto prvního sloupce a uloží se jako proměnná *hodnota*.
- Zjistí se, zda je řádek aktivní či nikoliv. Pokud je řádek aktivní, jsou zavolány funkce na deaktivaci, a naopak.
- Jsou definovány prázdné globální proměnné *svgGuide*, *svgDoc* a *zobrazeneLegendy*.
- Tyto proměnné slouží k provázání výsledků z *LOAD* s *READY* skripty.
- Následně ve druhé části *LOAD* jsou již veškeré externí soubory načteny vč. SVG. Tento objekt SVG je uložen do globální proměnné *svgDoc*.
- SVG je momentálně načteno se všemi grafy, které jsou přes sebe (viz obr. 7.5).



Obr. 7.5: Radarový graf: zobrazení všech 70 grafů najednou

- Tento nechtěný stav průběhu je zakryt načítacími bannery, které informují o odhadovaném stavu v procentech.
- Pro vytvoření *svgGuide* a pro následnou manipulaci se SVG objekty je vytvořena lokální proměnná *legends* pomocí *svgDoc*, při jejím vytváření je nejdříve vytvořeno dočasné pole *souhrn*.

- Bylo nutné v SVG nalézt takovou strukturu, která je dostatečně unikátní, aby se daly potřebné informace nalézt a dále bylo možné s nimi pracovat (popisky grafu nebo grafy). Struktura legend v části *legends* je dostatečně unikátní (viz obr. 7.6).

```

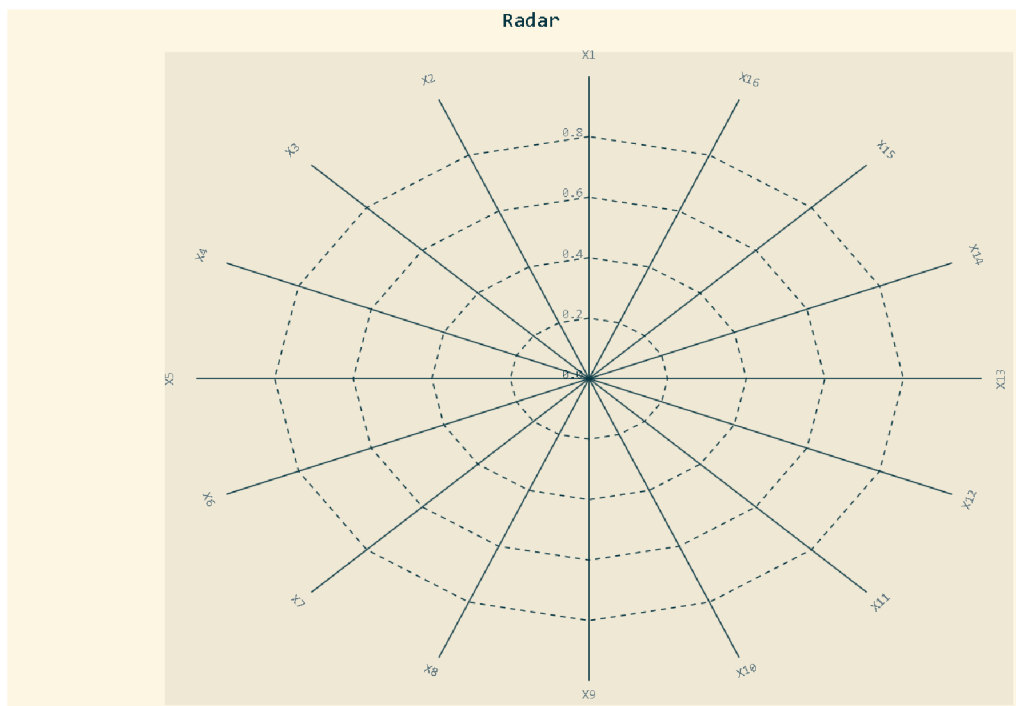
<text class="value" x="569.5024035144074" y="198.4609334889224">0.788491368992254</text>
<text class="value" x="486.82144250700935" y="160.43779224261058">0.652483490761369</text>
<text class="value" x="435.814675661818" y="99.77440983523333">0.775056849000975</text>
</g>
</g>
<g class="plot tooltip-overlay" transform="translate(112, 46)">
  <g class="tooltip" style="opacity: 0" transform="translate(0 0)">
    <rect class="tooltip-box" height="0" rx="0" ry="0" width="0" />
    <g class="text" />
  </g>
</g>
</g>
<g class="legends" transform="translate(10, 56)">
  <g class="legend reactive activate-serie" id="activate-serie-0">
    <rect class="color-0 reactive" height="12" width="12" x="0.0" y="1.0" />
    <text x="17.0" y="11.2">P1013</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-1">
    <rect class="color-1 reactive" height="12" width="12" x="0.0" y="22.0" />
    <text x="17.0" y="32.2">P1020</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-2">
    <rect class="color-2 reactive" height="12" width="12" x="0.0" y="43.0" />
    <text x="17.0" y="53.2">P1021</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-3">
    <rect class="color-3 reactive" height="12" width="12" x="0.0" y="64.0" />
    <text x="17.0" y="74.2">P1022</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-4">
    <rect class="color-4 reactive" height="12" width="12" x="0.0" y="85.0" />
    <text x="17.0" y="95.2">P1025</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-5">
    <rect class="color-5 reactive" height="12" width="12" x="0.0" y="106.0" />
    <text x="17.0" y="116.2">P1026</text>
  </g>
  <g class="legend reactive activate-serie" id="activate-serie-6">

```

Obr. 7.6: Vnitřní zobrazení SVG struktury pro definování popisků v legendě

- Souhrn je tvořen z *id* (definuje popisky, *P1013*), *legendId* (definuje id popisku, *activate-serie-0*) a *graphClass* (definuje graf v SVG, *serie-0*). Pro tvorbu *svgGuide* je vytvořen cyklus pro naindexování všech potřebných objektů z SVG.
- Následně je pomocí proměnné *hodnota* tvořen *svgGuide[]*, kde ke každému id (*P1013*) jsou přiřazeny objekty.
- Každý naindexovaný graf je možné rovnou skrýt.
- Lokální proměnná *legend*, do které je uloženo id grafu, a *graph*, do které je uložena třída grafu.
- Plot je objekt, kde je vyhledáván daný graf pomocí *reactive*, při nalezení v *svgDoc* je uložen.
- *plotOverlay* vyhledává textové popisky grafu, hledá id grafu uvnitř třídy *text-overlay*.
- Vyhledání v proměnné *legend*, do které byly uloženy id popisku (*activate-serie-0*), pomocí *.hide()* schová legendu.
- Pomocí funkce *.hide()* jsou schovány jak grafy, tak i popisky.

- U proměnné *plotOverlay* jsou popisky skryty pomocí CSS, stanou se tak průhlednými.
- Náhled načteného SVG je na obrázku 7.7.



Obr. 7.7: Radarový graf se skrytými grafy i popisky

Zde se přechází do bloku *READY*. Byly inicializovány veškeré potřebné části a vytvořena nezbytná pole pro další práci skriptů. Dále jsou popsány funkce reagující na kliknutí na řádek v tabulce, které využívají těchto připravených dat. Načítací bannery jsou v tomto okamžiku skryty.

- U funkce *zobrazitGraf()*, je zpracováváno jen *svgGuide* a hodnota, ve které se nachází id řádku z tabulky (*P1013*).
- Do lokální proměnné *legend* je uložen objekt s daným id. Dále je lokální proměnná *plot*, která vyplní graf.
- Pro zobrazení výplně grafů je použita funkce *.show()* a zobrazení popisku grafu je řešeno pomocí CSS. Protikladem je funkce *.hide()*.
- Kvůli problematice držení jednotlivých pozic u zobrazování popisku v legendě je využita seřazovací funkce, aby se jednotlivé popisky zobrazovaly pod sebou, a ne na předem určených souřadnicích. Proto je nutné tyto souřadnice přepisovat.
- Řazení legend funguje dvoufázově. Nejprve manipulace s polem *legend* k vypsání, poté samotné vypsání.

- Funkce *vypsatLegendy()* používá metodu *.sort()* pro seřazování popisků v legendě. Dále je řešena podmínka, že pokud jsou již nějaké popisky k zobrazení, tak je vypíše.
- Každý popisec v grafu má jisté souřadnice. Tyto souřadnice jsou přepisovány v závislosti na počtu zobrazených grafů, tedy popisků.
- Aby byly popisky jeden za druhým, a ne s mezerami (např. po kliknutí na první a čtvrtý řádek tabulky by vznikly dvě mezery mezi zobrazenými popisky), je nutné tyto souřadnice přepsat.
- Vnořený for cyklus pro každý popisec grafu vypíše popisec pod předchozí podle souřadnic vypočtených z indexu popisku.
- Mění se tedy pouze y-složka. Rozdíl mezi zobrazením jednotlivých popisků v legendě je 21 pixelů, kdy první popisec je posunut o 11,2 pixelů vůči horní hranici vizualizace a souřadnice checkboxu je posunuta o absolutní hodnotu 1 pixelu.
- Skryje se popisec, změní se y-souřadnice jak textu, tak i checkboxu v daném popisku a znova se zobrazí.
- Funkce *srytGraf()* dělá totéž jako funkce *zobrazitGraf()*, až na to, že je inverzí, tudíž ze zobrazeného grafu uděla skrytý.

Kvůli dlouhému načítání webové stránky byly do aplikace (aby nevypadala nečinná) přidány načítací bannery, které informují uživatele o průběhu, aby se zakryla skutečná podoba nezpracovaného SVG se všemi grafy.

Celkový výstup lze vidět na obrázku 7.8.

## 7.4 Blok problematiky

Během práce byla zaznamenána určitá úskalí a chyby, na které zde bude upozorněno.

Zobrazování a vývoj webové stránky byly testovány v prohlížečích Chrome 50, Firefox 46 a IE11.

Firefox jakožto výchozí prohlížeč po dobu testování nehlásil žádnou chybu, weboví vývojáři tohoto prohlížeče se snaží být velmi vývojářsky přívětiví.

U IE11 nastává problém jen u bezpečnostního upozornění – chybové hlášení, které upozorňuje uživatele na spouštějící se službu na pozadí. IE eviduje v základu jakýkoliv JS, který musí být následně uživatelem odkliknut a povolen, pokud je spouštěn z určitých souborů z file:/// (viz obr. 7.9). Dále nastává problém s jedním z JavaScriptů, který přestane poskytovat interaktivitu SVG grafu (najatím kurzoru na daný graf by se měly zobrazit dané hodnoty). Tento problém nebyl vyřešen.

Například Chrome, Vivaldi, Opera a další prohlížeče na jádru Blink zakazují defaultně cross-domain requests (cross-domain requests jsou povoleny u JavaScriptu

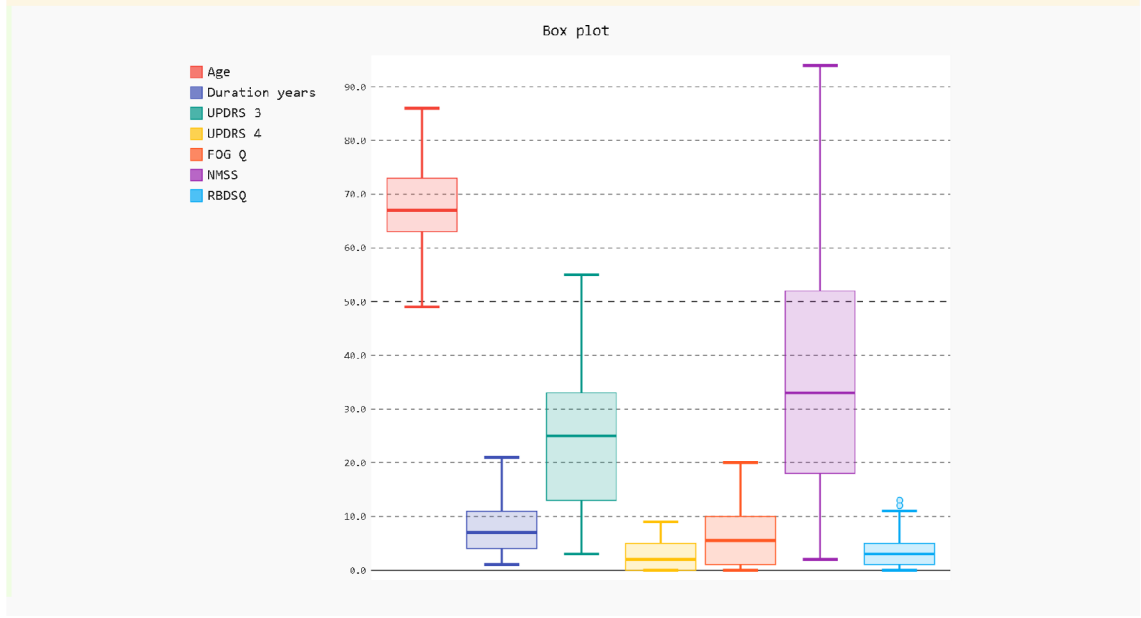
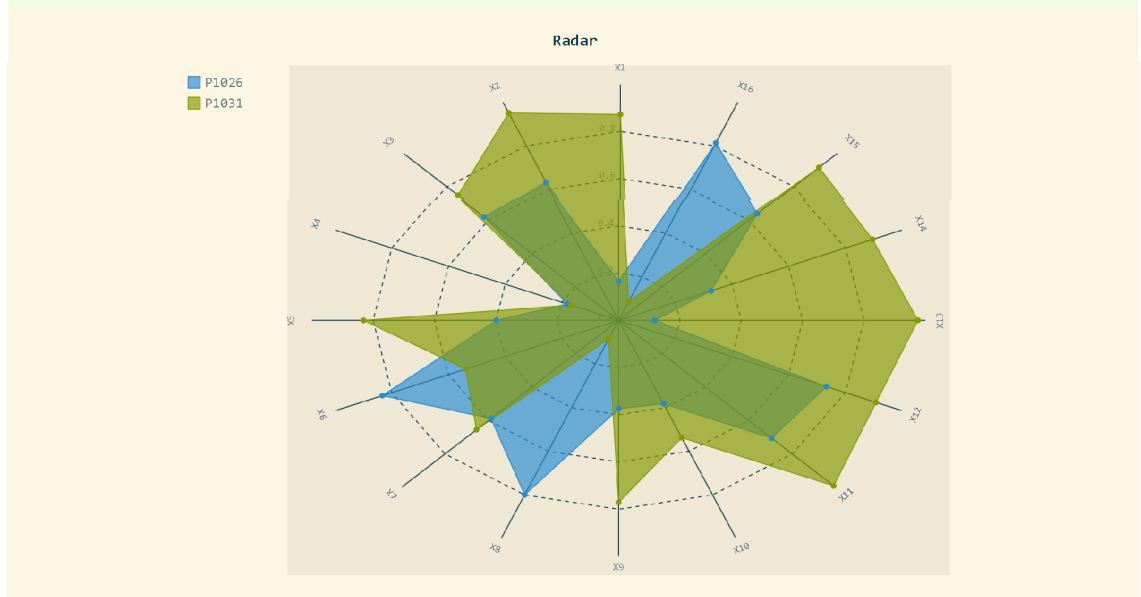


Show 10 - entries

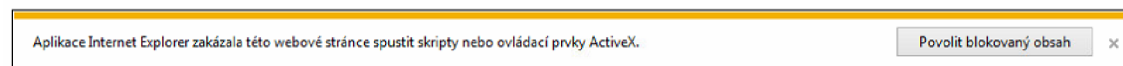
Label	Gender	Age	Duration years	UPDRS 3	UPDRS 4	FOG Q	NMSS	RBDSQ	LED	ACE R	MMSE	BDI
P1013	female	74	6	12	4	1	19	3	535	93	30	13
P1020	female	64	3	8	1	8	35	13	180	87	24	6
P1021	female	65	11	5	3	7	18	4	600	90	28	6
P1022	female	72	7	6	2	1	28	3	880	84	29	9
P1025	female	64	6	8	5	2	15	6	1138	98	38	10
P1026	female	73	8	12	1	1	49	3	540	86	30	17
P1027	female	65	2	8	2	2	12	1	740	92	25	4
P1031	female	59	5	10	3	2	54	1	918	90	38	20
P1033	female	73	4	14	3	3	40	1	650	80	27	18
P1041	female	62	4	13	3	1	7	4	390	88	28	3

Showing 1 to 10 of 70 entries

Previous 1 2 3 4 5 6 7 Next



Obr. 7.8: Celkový pohled na box-plot graf, radarový graf a tabulku



Obr. 7.9: IE detekuje na lokálním spuštění JS. Hláška pro upozornění uživatele.

a CSS, takže jediné omezení je u HTML elementů, jako je embed, object, iframe a další podobné externí zdroje), což znamená, že nemůže být načten SVG soubor na stránce spuštěné přímo jako file:/// . Na stránkách s doménou *NULL* se specificky také zakazuje spouštět externí soubory, přestože jsou také z domény *NULL*. Je to kvůli bezpečnostnímu riziku. Možným řešením je zobrazení chybového hlášení, že v tomto prohlížeči to nefunguje, jakmile je detekováno, že prohlížeč zakazuje cross-domain requesty [43].

Pokud prohlížeč povolí lokální spuštění jquery, popř. nezakazuje cross-domain requesty, nedochází k žádným komplikacím, jako například u Firefoxu. Pokud vzniká chyba, má prohlížeč striktní zákaz spouštění určitých souborů z file:/// a je třeba použít http://, což je nejjednodušší řešení.

Doporučení je vyhnout se používání file:/// a stáhnout si například program EasyPHP. Do složky WWW stačí nahrát svůj HTML soubor a v prohlížeči se zadá *localhost/* nebo *127.0.0.1* a je to.

Existuje samozřejmě více variant software pro použití lokálního serveru, nejenom EasyPHP. Ten je zde zmíněn, neboť byl využit právě při testování funkčnosti webové stránky.

Možným dalším řešením, i když ne moc vhodným, je nekládat soubory pomocí předloh, ale vložit přímo jak tabulku, tak i SVG soubor. U SVG souboru bylo nutné odstranit `<?xml?>` část. Tímto způsobem se dá obejít bezpečnostní politika cross-domain requestu, ale zdrojový kód se stává rozsáhlým a složitým. V prohlížečích s jádrem Blink nastává další problém, a to v renderování daného SVG, kdy se stane čistě nehybným prvkem. Neaplikují se na něj JavaScripty pro jeho interaktivnost.

## 8 ZÁVĚR

Tato práce se zabývá problematikou vizualizace dat. Díky rychlému technologickému vývoji je kolem nás nepřehledné množství informací. Abychom tyto informace mohli využít a sdílet s ostatními, je potřeba nejprve je analyzovat. Dále je nutné tyto informace vizualizovat a sdílet se čtenáři, to nám zajišťuje právě vizualizace dat. Proto se v práci pojednává o této problematice. Následně je rozebrána problematika při tvorbě prezentace dat.

Praktická část je zaměřena na tvorbu interaktivních výstupů z vstupního souboru dat po analýze řečových signálů u pacientů s Parkinsonovou nemocí. K tomuto účelu byl použit programovací jazyk Python. Ze zaslaných vstupních dat pro tvorbu programu bylo za pomoci importovaných knihoven a skriptů psaných v JS nakonec docíleno výstupních interaktivních grafů a HTML tabulky. Tyto grafy a SVG soubory byly implementovány do cílového HTML, provázání tabulky s daným SVG souborem proběhlo pomocí skriptu. Výstup byl tvořen tak, aby byl co nejjednodušší, dobře čitelný a zároveň velmi uživatelsky přívětivý.

Firefox, jakožto výchozí prohlížeč pro testování této práce, nehlásil celou dobu žádnou chybu a byl velmi vývojově přívětivý. Práce byla testována i v jiných prohlížečích. Při testování v jiných prohlížečích se vyskytovala specifická omezení, která se dají řešit různými způsoby. Asi největším problémem byla „police security“ u Blink jádra, jádra používaného v dnešní době u většiny prohlížečů. Pokud prohlížeč povolí lokální spuštění jquery a nezakazuje cross-domain requests, nedochází k žádným komplikacím. Pokud vzniká chyba, má prohlížeč striktní zákaz spouštění určitých souborů z file:/// a je třeba použít http://, což je nejjednodušší řešení. U IE11 se muselo uživatelsky povolit, že se daný JavaScript může spustit. Kvůli oznámení Microsoftu, který stojí za vývojem IE, že zastavuje podporu starších verzí IE než IE11, nebyl brán zřetel při vyvíjení webové stránky na kompatibilitu se staršími verzemi tohoto prohlížeče.

Při testování, zda dané prohlížeče na lokálním serveru mají nějaké problémy, byl použit program EasyPHP. Překážka, která se nakonec objevila jen u IE11, spočívala v tom, že přestal fungovat jeden ze skriptů v SVG (při najetí kurzorem nezobrazoval na daném bodě hodnoty). Bohužel se tento problém nepodařilo vyřešit.

Největší efektivita by se dala docílit, kdyby se grafy i SVG zpracovávaly čistě skripty psanými v JS. Vzhledem k tomu, že se jedná o dvoustupňovou práci – Python pro generování SVG a HTML pro generování celkového výstupu – je zapotřebí, aby JS potřebná data k práci ze SVG vytáhl a na základě arrays s potřebnými daty a indexy pak může JS pracovat s daným SVG a díky tomuto přístupu i s jakýmkoliv SVG generovaným v Pygal knihovně. Má to i své výhody. Do budoucna by se dalo uvažovat o celkovém použití *node.js*, aby se grafy i SVG zpracovávaly čistě JS.

## LITERATURA

- [1] Apache HTTP server. *BANAN* [online]. 2004 [cit. 2016.05.20] Dostupné z: <<https://www.banan.cz/napoveda/slovnicek-pojmu/apache-http-server>>.
- [2] BÍLEK, Petr. Úvod do jazyka Python. *sallyx.org* [online]. 2005 [cit. 2015.12.10] Dostupné z: <<http://www.sallyx.org/sally/python/python1.php>>.
- [3] BRESSERT, Eli. *SciPy and NumPy: [optimizing & boosting your Python programming]*. Sebastopol, CA:O'Reilly, 2012. 57s. ISBN 1449305466.
- [4] *Continuum* [online]. [cit. 2015.12.10] Dostupné z: <<https://www.continuum.io/>>.
- [5] DAMBORSKA, Lenka a HABROVANSKA, Pavlína. *Typografie* [online]. 2010 [cit. 2016.05.20] Dostupné z: <<http://typografie.webnode.cz/>>.
- [6] DLOUHÁ, Jana, RESSNER, Pavel, ŠIGUTOVÁ, Dana a ZAMIŠKOVÁ Gabriela. *Neurologie pro praxi* [online]. 2010, 5 s. [cit. 2015.12.10] Dostupné z: <<http://www.neurologiepropraxi.cz/pdfs/neu/2010/02/10.pdf>>.
- [7] DZEMYDA, Gintautas, Olga KURASOVA a Julius ŽILINSKAS. *Multidimensional data visualization: methods and applications*. York: Springer, 2013. 250s. Springer optimization and its applications, v.75. ISBN 978-1-4419-0235-1.
- [8] *easyphp.org* [online]. 2000 [cit. 2015.12.10] Dostupné z: <<http://www.easyphp.org/>>.
- [9] Funkčnost webu. *advertising4u.cz* [online]. 2009 [cit. 2016.05.20] Dostupné z: <<http://www.advertising4u.cz/funkcnost-webu.html>>.
- [10] GOMEZ-PALACIO, Bryony. a Armin. VIT. *Graphic design, referenced: a visual guide to the language, applications, and history of graphic design*. Beverly, Mass.: Rockport Publishers, 2009. 400s. ISBN 1592534473.
- [11] HALA, Michal. Chrome získává vedení mezi prohlížeči na desktopu. *svethardware* [online]. 2016 [cit. 2016.05.20] Dostupné z: <<http://www.svethardware.cz/chrome-ziskava-vedeni-mezi-prohlizeci-na-desktopu/42283>>.

- [12] HARUŠTIAKOVÁ, Danko, JARKOVSKÝ, Jiří, LITTNEROVÁ, Simona a DUŠEK, Ladislav. *Vícerozměrné statistické metody v biologii*. Brno: Akademické nakladatelství CERM, 2012, 110s. ISBN 978-80-7204-791-8.
- [13] HOLČÍK, Jiří. *Analýza a klasifikace dat*. Brno: Akademické nakladatelství CERM, 2012, 111s. ISBN 978-80-7204-793-2.
- [14] HŮLEK, Richard. *Metody vizualizace časoprostorových dat* [online]. Brno, 2014. 52s. [cit. 2015. 12. 10] Dostupné z: <[https://is.muni.cz/th/172775/fi\\_m/hulek-diplomova-prace.pdf](https://is.muni.cz/th/172775/fi_m/hulek-diplomova-prace.pdf)>.
- [15] CHAMPEON, Steve. JavaScript: How Did We Get Here? *archive.oreilly* [online]. 2001 [cit. 2016. 05. 20] Dostupné z: <[http://archive.oreilly.com/pub/a/javascript/2001/04/06/js\\_history.html](http://archive.oreilly.com/pub/a/javascript/2001/04/06/js_history.html)>.
- [16] Internetový prohlížeč. *Help Mark* [online]. 2013 [cit. 2016. 05. 20] Dostupné z: <<http://www.helpmark.cz/slovníkpojmu/33-internetovy-prohlizec>>.
- [17] JAHODA, Bohumil. Microsoft Edge: připravte se na prohlížeč, který nahradí Internet Explorer. *LUPA* [online]. 2015 [cit. 2016. 05. 20] Dostupné z: <<http://www.lupa.cz/clanky/microsoft-edge-pripravte-se-na-prohlizec-ktery-nahradi-internet-explorer/>>.
- [18] JANAK, Robert. *TYPOGRAFIE* [online]. 2001 [cit. 2016. 05. 20] Dostupné z: <<http://www.typografie.unas.cz/>>.
- [19] JANEČEK, Tomáš. Jak otevřít CSV soubor v OpenOffice. *webzpravodaj* [online]. 2013 [cit. 2016. 05. 20] Dostupné z: <<http://www.webzpravodaj.cz/242-jak-otevrit-csv-soubor-v-openoffice/>>.
- [20] *JetBrains* [online]. 2000 [cit. 2015. 12. 10] Dostupné z: <<https://www.jetbrains.com/pycharm/>>.
- [21] KASÍK, Pavel. Věříte grafům? Podívejte se, jak jednoduché triky používají manipulátoři. *idnes.cz* [online]. 2015 [cit. 2015. 12. 10] Dostupné z: <[http://technet.idnes.cz/manipulace-grafy-statistika-dgj-veda.aspx?c=A151023\\_164547\\_veda\\_pka](http://technet.idnes.cz/manipulace-grafy-statistika-dgj-veda.aspx?c=A151023_164547_veda_pka)>.
- [22] KOSEK, Jiří. *htmlguru* [online]. 2010 [cit. 2016. 05. 20] Dostupné z: <<http://htmlguru.cz/>>.

- [23] Krabičkové grafy. *IBM Knowledge Center* [online]. [cit. 2016. 05. 20] Dostupné z: [https://www.ibm.com/support/knowledgecenter/SSEP7J\\_10.2.0/com.ibm.swg.ba.cognos.ug\\_cr\\_rptstd.10.2.0.doc/c\\_id\\_obj\\_boxplots.html%23id\\_obj\\_boxplots?lang=cs](https://www.ibm.com/support/knowledgecenter/SSEP7J_10.2.0/com.ibm.swg.ba.cognos.ug_cr_rptstd.10.2.0.doc/c_id_obj_boxplots.html%23id_obj_boxplots?lang=cs).
- [24] LASÁK, Pavel. *Přehled 40. netradičních grafů vytvořených v Microsoft Excelu*. [online]. 2014 [cit. 2015. 12. 10] Dostupné z: <http://office.lasakovi.com/excel/grafy/netradicni-grafy-v-ms-excelu/#27>.
- [25] MAJDA, David. Do hlubin implementací JavaScriptu: 1. díl – úvod *zdroják.cz* [online]. 2008 [cit. 2016. 05. 20] Dostupné z: <https://www.zdrojak.cz/clanky/do-hlubin-implementaci-javascriptu-1-dil-uvod/>.
- [26] *Matematická biologie: E-learningová učebnice* [online]. [cit. 2015. 12. 10] Dostupné z: <http://www.http://portal.matematickabiologie.cz>.
- [27] MOUNIER, Florian. *Pygal* [online]. 2015 [cit. 2015. 12. 10] Dostupné z: <http://www.pygal.org/>.
- [28] MURRAY, Scott. *Interactive data visualization for the web. 1st ed.* Sebastopol, CA: O'Reilly Media, 2013, 255s. ISBN 1449361080.
- [29] NOAH, Iliinsky a STEELE, Julie. *Designing data visualizations*. Sebastopol, CA: O'Reilly, 2011. 110s. ISBN 9781449312282.
- [30] NOVÁK, Václav. K čemu je nám krabicový graf? Karmické zkoumání. *novak.blog.respekt* [online]. 2007 [cit. 2016. 05. 20] Dostupné z: <http://novak.blog.respekt.cz/k-cemu-je-nam-krabicovy-graf-karmicke-zkoumani/>.
- [31] *NumPy* [online]. 2013 [cit. 2015. 12. 10] Dostupné z: <http://www.numpy.org/>.
- [32] OŠMERA, Lubomír. Co je zabijákem přehlednosti grafu – pohled pod lupou. *aps-brno.cz* [online]. 2005 [cit. 2015. 12. 10] Dostupné z: <http://excel.aps-brno.cz/?p=301>.
- [33] *Python v3.1.5 documentation* [online]. 2012 [cit. 2015. 12. 10] Dostupné z: <https://docs.python.org/3.1/index.html>.
- [34] *scipy.org* [online]. 2008 [cit. 2015. 12. 10] Dostupné z: <https://docs.scipy.org/doc/scipy/reference/tutorial/general.html>.

- [35] Slovník internetových výrazů. *ok-business* [online]. [cit. 2016.05.20] Dostupné z: <<http://www.ok-business.cz/ok-business/slovník-internetovych-vyrazu/>>.
- [36] STANCLIFT, Michael. A quick look at browser engines (Trident, Gecko, WebKit, Presto). *getjins* [online]. 2008 [cit. 2016.05.20] Dostupné z: <<https://getjins.wordpress.com/2008/09/10/a-quick-look-at-browser-engines-trident-gecko-webkit-presto/>>.
- [37] SVG. *jecas* [online]. 2015 [cit. 2016.05.20] Dostupné z: <<http://jecas.cz/svg>>.
- [38] Tvorba webu. *Webovky.knihovna.cz* [online]. 2010 [cit. 2016.05.20] Dostupné z: <<http://webovky.knihovna.cz/jak-na-to/tvorba-webu/>>.
- [39] *tvorba-webu.cz/* [online]. 2013 [cit. 2016.05.20] Dostupné z: <<http://www.tvorba-webu.cz/>>.
- [40] VELEK, Daniel. *ZÁKLADY TYPOGRAFIE* [online]. Kladno, 2010. 47s. [cit. 2015.05.20] Dostupné z: <<https://www.spskladno.cz/stahuj.php?id=2406>>.
- [41] VORČÁK, Ján. *Nástroje na analýzu kódu v jazyku Python a vizualizáciaich výstupu* [online]. Brno, 2012. 40s. [cit. 2015.12.10] Dostupné z: <[https://is.muni.cz/th/359426/fi\\_b/thesis.pdf](https://is.muni.cz/th/359426/fi_b/thesis.pdf)>.
- [42] Webové prohlížeče. *maxzone* [online]. [cit. 2016.05.20] Dostupné z: <<http://www.browsers.maxzone.eu/>>.
- [43] WEST, Mike. An Introduction to Content Security Policy *html5rocks* [online]. 2012 [cit. 2016.05.20] Dostupné z: <<http://www.html5rocks.com/en/tutorials/security/content-security-policy/>>.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IE	Internet Explorer
JS	JavaScript
PHP	Hypertext Preprocessor
SVG	Scalable Vector Graphics
XML	Extensible Markup Language



# SEZNAM PŘÍLOH

A Obsah přiloženého CD

57

## A OBSAH PŘILOŽENÉHO CD

Obsahem přiloženého CD je zip soubor (vizualizace\_xburia17.zip) ve kterém se nachází:

**css** – obsahuje použité CSS styly

- jquery.dataTables.css – formátování HTML tabulky
- nacistani.css – formátování celé stránky

**img** – obsahuje použité obrázky

- obrázky jsou použity při inicializaci HTML tabulky

**js** – obsahuje použité JavaScripty

- html5-3.6-respond-1.1.0.min.js – JavaScript pro starší IE
- jquery.js – knihovna pro usnadnění práce s HTML DOM objekty
- jquery.dataTables.js – modul pro vizualizaci a správné fungování HTML tabulky
- \_readmi\_vizualizace\_xburia17.txt – popis jednotlivých souborů
- box\_grafy.svg – vygenerovaný interaktivní box-plot graf
- data.csv – vstupní data pro aplikaci Python
- diplomka\_python – zdrojový kód aplikace napsaný v Pythonu
- docasne\_svg\_pro\_orientaci\_mezi\_tridami.xml – dočasný soubor k náhledu
- index.html – hlavní vytvořený dokument
- index\_bez\_externiho\_nacistani.html – záložní hlavního dokumentu (bez externího načítání, všechno aplikováno uvnitř, jen pouze pro názornou ukázkou obejití některých určitých bezpečnostních politik)
- radar\_grafy – vygenerovaný interaktivní radarový graf
- tabulka.html – vygenerovaná HTML tabulky