|  |  |  |
| --- | --- | --- |
| **Univerzita Hradec Králové**  **Fakulta informatiky a managementu**  **Název katedry** | | |
| **Název bakalářské/diplomové práce**  (podtitul práce)  Bakalářská/Diplomová práce | | |
| Autor: Jméno, příjmení  Studijní obor: Studijní obor | | |
| Vedoucí práce: Titul, jméno, příjmení  Odborný konzultant: Titul, jméno, příjmení  Pracoviště | | |
| Hradec Králové | měsíc rok | |
| Prohlášení:  Prohlašuji, že jsem bakalářskou/diplomovou práci zpracoval/zpracovala samostatně a s použitím uvedené literatury. | | | |
| V Hradci Králové dne 31.10.2021 | | *vlastnoruční podpis*  Jméno a Příjmení | |
| Poděkování:  Děkuji vedoucímu bakalářské/diplomové práce titul, jméno, příjmení za metodické vedení práce a…. | | | |

Anotace

Text anotace – shrnutí cíle, významu práce a výsledky v ní dosažené. Délka minimálně 100 a maximálně 200 slov.

Annotation

Title: Název práce v anglickém jazyce

Anotace v anglickém jazyce. Délka minimálně 100 a maximálně 200 slov.

Obsah

[1 Úvod 1](#_Toc86488194)

[2 Cíl práce 2](#_Toc86488195)

[3 Metodika zpracování 3](#_Toc86488196)

[4 Teoretická část 4](#_Toc86488197)

[4.1 Vizualizace 4](#_Toc86488198)

[4.1.1 Vizualizace obecně 4](#_Toc86488199)

[4.1.2 Anatomie grafu 5](#_Toc86488200)

[4.1.3 Typy vizualizací 5](#_Toc86488201)

[4.1.4 Barvy v datové vizualizaci 5](#_Toc86488202)

[4.2 Jazyk Python 5](#_Toc86488203)

[4.3 Vizualizační knihovny jazyka Python 5](#_Toc86488204)

[4.3.1 Matplotlib 5](#_Toc86488205)

[4.3.2 Seaborn 5](#_Toc86488206)

[4.3.3 Bokeh 6](#_Toc86488207)

[4.3.4 Plotly 6](#_Toc86488208)

[4.3.5 Holoviews 7](#_Toc86488209)

[4.3.6 Pygal 7](#_Toc86488210)

[4.3.7 MidiTime 8](#_Toc86488211)

[4.3.8 Geoplotlib 8](#_Toc86488212)

[4.3.9 WordCloud 8](#_Toc86488213)

[4.4 Způsoby získávání dat 9](#_Toc86488214)

[4.5 Způsoby zpracování dat? 9](#_Toc86488215)

[5 Metodika zpracování 9](#_Toc86488216)

[5.1 Hodnocení knihoven 9](#_Toc86488217)

[6 Praktická část 10](#_Toc86488218)

[6.1 Porovnání vizualizačních knihoven 10](#_Toc86488219)

[6.1.1 Matplotlib 10](#_Toc86488220)

[6.2 Ukázkové úlohy 14](#_Toc86488221)

[6.2.1 Podřazená podkapitola 15](#_Toc86488222)

[7 Shrnutí výsledků 16](#_Toc86488223)

[8 Závěry a doporučení 17](#_Toc86488224)

[9 Seznam použité literatury 18](#_Toc86488225)

[10 Přílohy 20](#_Toc86488226)

Seznam obrázků

[Obr. 1 Název obrázku/grafu/fotografie. 3](#_Toc348517265)

Seznam tabulek

[Tabulka 1 Název tabulky. 3](#_Toc348517268)

# Úvod

Zde vysvětlit problémovou situaci a otázky, které se budou v bakalářské/diplomové práci řešit.

# Cíl práce

Smysl a účel, výzkumné otázky.

# Metodika zpracování

Cíle, hypotézy/ výzkumné otázky, způsob hledání odpovědí na výzkumné otázky včetně metodiky vlastního výzkumu/šetření, literární rešerše.

# Teoretická část

## Vizualizace

### Vizualizace obecně

Definice pojmu „vizualizace dat/informací“ není až tak jednoznačná a jednoduchá, jak by se na první pohled mohlo zdát.

Kniha „Lecture Notes in Computer Science“ z roku 1970 popisuje, že „*Vizualizace informací využívá počítačové grafiky a interakce, aby napomohla lidem v řešení problémů*“ [1]. Vizualizace však nemusí nutně sloužit pouze k tomuto účelu, může najít uplatnění například ve výuce matematiky, kde „*Matematické koncepty, jako čísla, funkce, nebo vektory […] nejsou součástí intuitivního chápání, tak jako reálné fyzické objekty*“ [2]. Vizualizací je však možné pochopení těchto konceptů studentům usnadnit. „*Informační vizualizace nabízí možnost snáze a lépe ukázat klasické vizuální reprezentace matematických formátů, ale také je obohatit o prvky pohybu a interaktivity […] informační vizualizace tedy plní didaktické funkce nezbytné pro výuku matematiky.*“ [2]

Vizualizace může plnit i dekorativní roli, Lorène Fauvelle v článku „Data visualization: definition, examples, tools, advice“ popisuje „Datové umění“ jako nejvyšší formu vizualizace kde „*Získávání informací a postřehů již nestačí, výsledek musí být také vizuálně atraktivní…* “ [3] v rozhovoru s Nicholasem Rougeux popisuje, že „*Datový umělec se nesnaží pouze informovat, ale hlavně vyvolat emoce*“. [4]

Většina zdrojů se však shoduje na dvou základních konceptech „redukce“ a „prostoru“, které popisuje i Lev Manovich v článku „What is Visualization?“: „*Vizualizace informací využívá jednoduchých grafických elementů jako zástupce reálných objektů a vazeb mezi nimi – nezáleží na tom, jestli se jedná o lidi, ceny na burze, příjmy států, nezaměstnanost, nebo cokoliv jiného. Skrze tyto grafické elementy informační vizualizace odhaluje vzorce a struktury v datech. Nicméně, cenou za tuto možnost je extrémní úroveň schematizace. Zahazujeme 99 % toho, co je na objektech specifické, ve snaze nalezení vzorců na zbývajícím 1 % charakteristik.*“ [7]

„*Co mají všechny vizualizační techniky společné, kromě redukce? Všechny využívají prostoru (umístění, velikosti, tvaru a v poslední době i zakřivení a pohybu) pro reprezentaci klíčových rozdílů v datech a zobrazení nejdůležitějších vzorců a vztahů. […] Ostatní, méně důležité vlastnosti objektů jsou pak reprezentovány jinými vizuálními prostředky – odstínem, barvou, vzorem stínování nebo i průhledností.* “ [7]

### Principy datové vizualizace

Základním cílem vizualizace je zjednodušit porozumění datům, ať už hmatatelným (jako jsou počty produktů ve skladu, hladiny řek v čase, nebo rozmístění zastávek autobusu), ale i čistě abstraktním, které nemají fyzickou podobu a jsou tak složitější k pochopení. Aby bylo možné lépe vytvářet grafy zobrazující data srozumitelným způsobem, je nejprve nutné pochopit jakým způsobem člověk zpracovává zrakové vjemy. K tomuto účelu může dle Encyclopedia of Human-Computer Interaction využít zákonů Gestaltismu (česky také „Tvarové psychologie“). Konkrétně jsou při vytváření vizualizací uplatnitelné následující principy:

**Princip blízkosti** – Objekty nacházející se blízko u sebe jsou vnímány jako skupina

**Princip podobnosti** – Objekty stejného, nebo podobného tvaru, barvy či jiných vlastností jsou vnímány jako skupina

**Princip ohrady** – Objekty ohraničené čarou, nebo na stejnobarevném poradí jsou považovány za skupinu

**Princip uzavření** – Vnímání tvaru jako celku i v případě, že obsahuje mezery

**Princip návaznosti** – Objekty zarovnané do čáry/křivky jsou považovány za jeden celek

**Princip spojitosti** – Všechny objekty spojené čarou jsou považovány za jeden celek [18]

Využitím těchto principů lze čtenáře upozornit na konkrétní závislosti nalezené v grafu, ale také vyhledat a upravit části vizualizace, které by mohly vést k nepravdivým závěrům.

### Barvy v datové vizualizaci

Dalším důležitým aspektem vizualizace jsou použité barvy a barevné palety. Ačkoliv mnoho zdrojů se shoduje, že účelem barev není estetika, i ta je pro mnohé uživatele aspektem pro volbu specifických barev v grafu. [20]

Barevné palety lze rozdělit do třech základních skupin. Každá z těchto skupin je vhodná pro vizualizaci jiného typu dat. Sekvenční palety se obvykle skládají z různých odstínů jedné barvy, které se liší svým jasem. Taková paleta je vhodná pro data, která jsou nějakým způsobem příbuzná, protože dle principu podobnosti z kapitoly 4.1.2. budou vnímána jako jedna skupina. [19] Zvláštní podmnožinou sekvenčních palet jsou takzvané „pro lidské vnímání uniformní“ (ang. „perceptually uniform“) barevné palety, kde není využita pouze jedna barva, ale několik barev s proměnlivým jasem tak, aby kontrast mezi stejně vzdálenými hodnotami byl stejný ve všech částech palety a předešlo se tak ztrátě informací, nebo naopak nalezení zdánlivých anomálií v částech s vyšším vnímaným kontrastem. [21]



Obr. 1 Pro lidské vnímání uniformní sekvenční palety knihovny Matplotlib

Zdroj: <https://matplotlib.org/stable/tutorials/colors/colormaps.html>

Dalším typem jsou divergující (rozcházející se) barevné palety, které přechází z jedné kontrastní barvy do druhé přes „inflexní bod“ v jejich středu (obvykle se jedná o velmi světlou barvu, často bílou). Tato barevná paleta je vhodná pro data, která se nějakým způsobem dělí na dvě hlavní skupiny (například kladné a záporné hodnoty, nebo hodnoty, které se dají prezentovat jako positivní a negativní). [19] [20]

Posledním typem jsou kategorické (někdy také kvalitativní, či nominální) barevné palety, používané pro data, které nemají žádné souvislosti a jsou navrženy tak, aby je nebylo možné vnímat jako posloupnost. [20] Ačkoliv některé z těchto palet mohou být velmi obsáhlé, pro zachování přehlednosti se doporučuje nepoužívat více než 8 různých kategorií. [19]

Při výběru palety je také nutné myslet na to, jak budou barvy na čtenáře působit. Dle průzkumu popsaném v práci „Affective Color in Visualization“ je možné rozdělit barevné palety do skupin podle pocitu, který vyvolávají. Konkrétně uklidňující palety se skládají převážně ze světlých, chladných barev s nízkou sytostí (například různé odstíny modré a zelené). Naopak vzrušující se vyznačují vysokou sytostí a využívají hlavně teplých barev, jako je červená, žlutá, nebo oranžová. Asi nejdůležitějším aspektem je, jestli daná vizualizace celkově působí positivním, nebo negativním dojmem. Jako positivní jsou často vnímány syté barvy, často obsahující odstíny modré, zelené, ale i žluté či oranžové, naopak negativně jsou vnímány tmavé barvy, hlavně odstíny červené, hnědé a šedé. [22]



Obr. 2 Skupiny barev dle asociovaných pocitů

Zdroj: Lyn Bartram, Abhisekh Partra a Maureen Stone: Affective Color in Visualization

### Anatomie grafu

### Nejčastější typy grafů

## Jazyk Python

## Vizualizační knihovny jazyka Python

### Matplotlib

Pravděpodobně nejznámější vizualizační knihovna jazyka Python byla vytvořena okolo roku 2003 Johnem Huterem byla původně určena k vizualizaci dat zaznamenaných elektrokortikografií při výzkumu epilepsie. Laboratoř, ve které John Hunter pracoval, měla v té době pouze jednu licenci na softwarový balíček pro analýzu dat, o který se museli všichni výzkumníci dělit. Hunter se tedy rozhodl vytvořit náhradu v prostředí MATLAB. Tato aplikace však nebyla ideální pro vizualizaci dat z mnoha zdrojů (kromě elektrokortikografie například i EEG a magnetické rezonance) uložených na několika serverech. Hunter proto začal vyvíjet novou aplikaci v jazyce Python „EEG viewer and analyzer“, která se postupem času změnila v dnešní knihovnu matplotlib. [5]

V současné době se jedná o volně dostupný open-source projekt, sloužící k tvorbě statických, animovaných i interaktivních vizualizací nejčastěji ve 2D (knihovna však podporuje i trojrozměrné vizualizace). [6]

### Seaborn

Seaborn vznikl jako řešení nejčastějších nedostatků knihovny matplotlib, konkrétně defaultních nastavení vizualizace před verzí 2.0, která byla založena na vizualizacích MATLABu, nižší úrovně API u matplotlibu, která často měla za následek nadbytečný kód a složitost využití datových struktur knihovny pandas. [9]

Seaborn buduje na základech knihovny matplotlib, ke které poskytuje vysokoúrovňové rozhraní, také dokáže využívat datové struktury knihovny pandas. Seaborn na základě specifikace typu grafu zvládne automaticky provázat hodnoty v datech s vizuálními atributy, jako je barva, velikost a styl, propočítat statistické transformace a doplnit ke grafu informativní štítky a legendu. Díky tomu, že Seaborn dokáže vytvořit kompletní vizualizace jedním voláním funkce s minimálním počtem argumentů, je ideálním nástrojem pro explorační analýzu dat. [8]

### Bokeh

Stejně jako matplotlib je knihovna Bokeh open source projektem, finančně podporovaným neziskovou organizací NumFOCUS, která se zaměřuje na financování a propagaci, open source nástrojů ve vědě a výzkumu. [10]

Oproti knihovnám matplotlib a seaborn však Bokeh nevytváří kompletní grafy jednou funkcí, místo toho nabízí širokou škálu nástrojů pro manipulaci s elementy vektorové grafiky (nazývanými „glyphs“), ze kterých lze následně vytvářet jednotlivé vrstvy vizualizace. To dává uživateli mnohem větší kontrolu nad vzhledem finální vizualizace, za cenu větší složitosti kódu, potřebného na její vytvoření. Bokeh dále dovoluje do vizualizace přidávat interaktivní elementy, jako slidery, tlačítka nebo drop-down menu. Knihovna také umožňuje vytvoření „Bokeh serveru“, který zajišťuje možnost streamování dat a složitější uživatelské interakce.

Knihovna se vlastně skládá ze dvou, Bokeh pro Python – sloužící k vytváření vizualizací a BokehJS – knihovna jazyka JavaScript sloužící k renderování vizualizace a zajištění interaktivity ve webovém prohlížeči. [11]

### Plotly

Další open source knihovnou pro datovou vizualizaci je Plotly. Tato knihovna však není omezena pouze na jeden jazyk, lze ji používat v jazycích Python, R, Julia, MATLAB a existují i projekty zpřístupňující ji v jazyce Java a jazycích frameworku .NET. Implementace v jazyce Python je však její zdaleka nejoblíbenější variantou a existují zde dvě možnosti využití knihovny. Jednodušší vysokoúrovňový Plotly Express, který s minimem kódu vytváří kompletní vizualizace a složitější modul Plotly Graphic Objects, který však dovoluje mnohem větší kontrolu nad vytvářenou vizualizací a je pomocí něho možné vytvořit některé typy grafů, nepodporované modulem Express. [12]

Existují také komerční produkty založené na této knihovně, Dash Enterprise - platforma pro analýzu trhu, datové vědy a výzkum v oblasti umělé inteligence a Chart Studio Enterprise – nástroj pro rychlou tvorbu vizualizací přímo ze souborů nebo databází a jejich vkládání do webových stránek. [12]

### Holoviews

Na rozdíl od předchozích vizualizačních knihoven, není knihovna Holoviews schopna sama vizualizovat data a je v tomto ohledu závislá na knihovnách Matplotlib, Bokeh a Plotly. Místo toho se Holoviews soustředí na co možná nejjednodušší manipulaci s grafy jako objekty, které jsou až do chvíle zobrazení nezávislé na jakékoliv vizualizační knihovně. Tyto objekty je také možné skládat do složitějších vizualizací pomocí seznamů a jednoduchých operátorů jako + (pro rozložení grafů vedle sebe) nebo \* (pro překrytí grafů). Práce s daty k vizualizaci je také zjednodušena, u většiny typů grafů stačí data popsat a předat konstruktoru objektu grafu informace o veličinách, které mají být vizualizovány. Dle popisu knihovny tato zjednodušení dovolují uživateli „*soustředit se na to, jakým způsobem chce data prozkoumat a co se snaží ukázat namísto samotného procesu tvorby grafu*“ [13]

### Pygal

Knihovna Pygal je poslední z častěji používaných vizualizačních knihoven jazyka Python. Zaměřuje se převážně na jednoduchou tvorbu interaktivních vizualizací, ideálních pro vložení do webových stránek díky exportu do formátu vektorové grafiky. Nabízí 14 různých typů grafů, včetně geografických vizualizací a relativně široké možnosti přizpůsobení včetně vestavěných stylů a možnosti definovat vlastní styly. [14]

### MidiTime

Knihovna MidiTime neslouží ke klasické vizualizaci dat, naopak se zabývá oborem, který by se dal nazvat „sonifikace dat“. Jedná se o vyjádření dat rozložených v čase jako zvuk (v případě této knihovny jde o formát MIDI, který je možné využít v syntetizátorech a jiných elektronických nástrojích). Knihovnu vytvořil Michael Corey v roce 2015 za účelem znázornění počtu zemětřesení v americkém státě Oklahoma pro rádiové zpravodajství. [15] Takové znázornění dat by mohlo najít využití nejen pro média, jako je rádio nebo podcast, ale mohlo by přispět i k přístupnosti informací na webových stránkách pro lidi se zrakovým postižením nebo dokonce ke tvorbě hudby.

### Geoplotlib

Ačkoliv některé z předcházejících knihoven podporují zobrazování dat na mapě, Geoplotlib se specializuje pouze na zobrazování geografických dat využívající OpenStreetMap jako podklad a OpenGL pro rychlé renderování grafických elementů na mapový podklad. Knihovna podporuje několik různých typů grafů jako body, teplotní mapy, histogramy, nebo i vykreslení tvarů v podobě geografických shapefiles a geojson. [16]

### WordCloud

Poslední vizualizační knihovna se zabývá spíše estetickou než informativní vizualizací, konkrétně v podobě takzvaného Word cloudu (v češtině někdy také „Slovního mraku“). Jedná se o vizualizaci založenou na četnosti slov v textu. Čím častěji se slovo v textu vyskytuje, tím větší bude ve výsledném „mraku“. Kromě velikosti slov dovoluje knihovna WordCloud manipulovat s barvou slov (ať už na základě četnosti, délky, či jiného uživatelem definovaného pravidla) a s celkovým rozvržením „mraku“ pomocí masky založené na libovolném obrázku. [17]

## Způsoby získávání dat

## Způsoby zpracování dat?

# Metodika zpracování

## Hodnocení knihoven

Aby bylo možné využité knihovny snáze porovnávat, budou jejich funkcionality hodnoceny dle následující tabulky:

Tabulka 1 Ukázková tabulka hodnocení

|  |  |
| --- | --- |
| Závislosti knihovny | Seznam knihoven, které jsou nutné pro fungovaní dané vizualizační knihovny |
| Podporované vstupní formáty | Formáty a datové struktury, které je knihovna schopná zpracovat |
| Podporované výstupní formáty | Formáty, ve kterých je možné vytvořené vizualizace ukládat |
| Poskytované typy grafů | Které typy grafů knihovna poskytuje? Existují nějaké často používané grafy, které v této knihovně chybí? |
| Poskytované možnosti přizpůsobení | Jakým způsobem lze vytvořené vizualizace upravovat? |
| Poskytované možnosti interaktivity | Jakým způsobem může uživatel s vytvořenou vizualizací interagovat |
| Jednoduchost použití | Jaká je složitost potřebného kódu pro vytvoření a přizpůsobení vizualizace? |
| Přehlednost a atraktivita výchozích nastavení | Jak přehledná a vizuálně atraktivní je výsledná vizualizace při minimálním použití přizpůsobení? |

Pro každou obecnou vizualizační knihovnu bude vytvořeno několik skriptů podle společného vzoru (projekt Templates). Budou otestovány podporované vstupní formáty, způsoby exportu vizualizace, nabízené možnosti přizpůsobení a hodnocení schopnosti rozlišit větší množství dat v jednom grafu. Aby bylo možné knihovny objektivněji porovnávat, budou tyto aspekty testovány pomocí často používaných typů grafů podporovaných všemi knihovnami, tedy graf sloupcový, spojnicový a bodový.

# Praktická část

## Porovnání vizualizačních knihoven

### Matplotlib

#### Závislosti

Knihovna Matplotlib má následující závislosti:

* FreeType –sloužící k renderování fontů
* libpng –pro manipulaci se soubory typu png
* NumPy – knihovna pro práci s daty ve formátu vícerozměrných polí
* cycler – objekt pro nekonečné cyklické procházení seznamů
* dateutil – rozšíření DateTime modulu jazyka Python
* kiwi – rychlé řešení soustav rovnic a nerovnic

#### Vstupní formáty

Vstupní formáty pro vizualizaci byly otestovány v programu Matplotlib\_Input.py

Matplotlib zvládne data zpracovávat jako standardní seznamy a uspořádané n-tice (tuple) jazyka Python i pole knihovny Numpy. Objekty typu DataFrame z knihovny Pandas zvládne zpracovat přímo pouze pro některé typy grafů, u jiných (například u sloupcového grafu), lze volat metodu .plot() tohoto objektu, která vrátí objekt Axes, který lze snadno vložit jako podgraf do vytvářené vizualizace. Pro vytvoření grafu nelze přímo použít slovník jazyka Python, je nutné jeho části předat jako seznamy.

#### Výstupní formáty

Knihovna Matplotlib zvládne vytvořené vizualizace zobrazovat do okna, které nabízí uživateli určitou míru interaktivity (přibližování grafů, pohyb s osami, úprava podgrafů), export do souboru lze provést přímo z kódu voláním metody .savefig(), nebo z již zmiňovaného okna přes souborový dialog. Matplotlib nabízí široký výběr formátů, od často používaných jako jsou png, jpeg, svg, pdg až po vzácnější formáty, například raw, rgba a tiff.

#### Typy grafů

Podporovány jsou nejrůznější typy dvourozměrných vizualizací od jednoduchých grafů po vizualizace založené na obrazových datech, ale i specializované typy grafů jako je Sankeyův diagram. Vytvářet lze i velké množství trojrozměrných vizualizací. V případě, že nabízené typy vizualizací nejsou dostačující, lze vytvářet i vlastní skládáním existujících vizualizací a manipulací se základními geometrickými tvary.

#### Přizpůsobení vizualizací

Ukázku přizpůsobení vizualizací lze nalézt v programu Matplotlib\_Customisation.py a případ s mnoha daty v Matplotlib\_ManyValues.py

Možnosti přizpůsobení vizualizací jsou velmi hluboké, od rychlého přepnutí celkového stylu grafu, přes jednoduché změny velikosti a barvy objektů při vytváření grafu, až po možnost jednotlivě manipulovat s libovolným elementem hotové vizualizace. Je možné snadno měnit barvu a velikost elementů v závislosti na hodnotách, díky speciálním argumentům a tzv. „colormaps“, které definují gradient barev, ze kterého lze vybírat na základě zobrazovaných dat.

Možnosti přizpůsobení mohou být někdy mírně nekonzistentní ve způsobu jejich použití, například při vytváření sloupcového grafu, lze nastavit barvy jednoduchým doplněním seznamu barev do argumentu „color“, stejným způsobem však nelze použít seznam šrafování pro argument „hatch“, ten přijímá pouze jeden typ šrafování. Pro použití různých šrafování je nutné vytvořit více grafů v jednom diagramu, nebo přistupovat přímo k vlastnost „hatch“ u jednotlivých grafických elementů grafu.





Obr. X a Y Možnosti rozlišení hodnot v knihovně Matplotlib

Zdroj: vlastní zpracování

#### Interaktivita

Jak již bylo zmíněno určitá míra interaktivity je možná pomocí zobrazení v okně knihovny Matplotlib. Pro hlubší interaktivitu je možné využít událostí, které dovolují detekovat klikání, pohyb a tažení myši a reagovat na ně. Je také možné opakovaně volat „animační funkci“ pozměňující zobrazovaná data a vytvářet tak animované grafy.

#### Jednoduchost použití

Knihovna Matplotlib je velmi jednoduchá na použití, k vytvoření vizualizace obvykle není třeba více než jen několik řádků kódu. Až na vzácné případy, jako například šrafování popsané v dřívější kapitole, se způsob volání metod a dosazování argumentů řídí jednoduchými a snadno pochopitelnými pravidly. Ke knihovně je dostupná obsáhlá a velmi detailně zpracovaná dokumentace popisující všechny její součásti, od grafů jako celku až po vlastnosti a metody jednotlivých částí grafu a grafických elementů. V případě, že by informace v dokumentaci nebyly dostačující, je díky rozšířenosti této knihovny velmi snadné je dohledat na jiných webových stránkách.

#### Výchozí nastavení

Výchozí nastavení knihovny pro většinu typů grafů je dostačující k nalezení některých závislostí mezi zobrazovanými daty, ale v mnoha případech nejsou příliš atraktivní pro čtenáře, vzhled lze však snadno změnit výběrem z mnoha zabudovaných stylů.





Obr. X a Y Výchozí vzhled grafů knihovny Matplotlib

Zdroj: vlastní zpracování

#### Shrnutí

|  |  |
| --- | --- |
| Závislosti knihovny | FreeType, libpng, NumPy, cycler, Dateutil, kiwi |
| Podporované vstupní formáty | Seznam, tuple, Numpy pole, Pandas DataFrame (u některých grafů) |
| Podporované výstupní formáty | png, jpeg, svg, pdf, pgf, ps, tiff, raw, rgba |
| Poskytované typy grafů | Velmi široký výběr 2D a 3D grafů, možnost tvorby vlastních vizualizací |
| Poskytované možnosti přizpůsobení | Široké možnosti úprav všech elementů, colormaps, předem definované styly, anotace a legendy |
| Poskytované možnosti interaktivity | Handlery pro události myši a klávesnice, periodicky volané funkce pro animace |
| Jednoduchost použití | Snadno pochopitelný zápis kódu, velmi dobře zpracovaná dokumentace |
| Přehlednost a atraktivita výchozích nastavení | Výchozí vizualizace nejsou příliš atraktivní, nicméně jsou pro mnoho účelů dostačující |

## Ukázkové úlohy

Vlastní text práce.

Tabulka 1 Název tabulky.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Zdroj: citace zdroje, nebo autor, vlastní zpracování

### Podřazená podkapitola

Vlastní text práce.



Obr. 1 Název obrázku/grafu/fotografie.

Zdroj: citace zdroje, nebo autor, vlastní zpracování

# Shrnutí výsledků

Souhrn vlastních výsledků získaných v průběhu řešení problému.

# Závěry a doporučení

Kritická diskuze nad výsledky, ke kterým autor dospěl (soulad výsled-ků  literaturou či předpoklady; výsledky a okolnosti, které zvláště ovlivnily předkládanou práci atd.). Je vhodné naznačit i případné další (popř. alternativní) možnosti zkoumání dané problematiky a otevřené problémy pro další studium.

# Seznam použité literatury

1. PURCHASE, Helen et al. Theoretical Foundations of Information Visualization. In: *Lecture Notes In Computer Science*. 4950. 1970, s. 46–64. ISBN 978-3-540-70955-8. DOI: [10.1007/978-3-540-70956-5\_3](https://doi.org/10.1007/978-3-540-70956-5_3)
2. G. CHIAPPINI a R.M. BOTTINO. *Visualisation in Teaching-Learning Mathematics: The Role of the Computer* [online]. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.39.3360&rep=rep1&type=pdf>
3. LORÈNE FAUVELLE. *Data visualization: definition, examples, tools, advice [guide 2021]* [online]. 2020 [cit. 03.04.2021]. Dostupné z: <https://www.intotheminds.com/blog/en/data-visualization/>
4. INTOTHEMINDS. *What is a data artist? | with Nicholas Rougeux* [online]. 2020 [cit. 16.09.2021]. Dostupné z: <https://www.youtube.com/watch?v=k4D9qgVb17Q>
5. JOHN HUNTER a MICHAEL DROETTBOOM. The Architecture of Open Source Applications (Volume 2): matplotlib. In: [cit. 16.09.2021]. Dostupné z: <http://aosabook.org/en/matplotlib.html>
6. JOHN HUNTER a DARREN DALE. Overview — Matplotlib 3.4.3 documentation. In: . 2021 [cit. 08.09.2021]. Dostupné z: <https://matplotlib.org/stable/contents.html>
7. MANOVICH, Lev. What is visualisation? *Visual Studies*. Routledge, 2011, roč. 26, č. 1, s. 36–49. ISSN 1472-586X. DOI: [10.1080/1472586X.2011.548488](https://doi.org/10.1080/1472586X.2011.548488)
8. WASKOM, Michael. seaborn: statistical data visualization. *Journal of Open Source Software*. 2021, roč. 6, č. 60, s. 3021. ISSN 2475-9066. DOI: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021)
9. VANDERPLAS, Jake. *Python Data Science Handbook: Essential Tools for Working with Data*. 1st edition. vyd. Sebastopol, CA: O’Reilly Media, 2016. ISBN 978-1-4919-1205-8.
10. Sponsored Projects | pandas, NumPy, Matplotlib, Jupyter, + more. In: *NumFOCUS* [online] [cit. 19.09.2021]. Dostupné z: <https://numfocus.org/sponsored-projects>
11. BOKEH CONTRIBUTORS. Bokeh documentation. 2021 [cit. 08.09.2021]. Dostupné z: <https://docs.bokeh.org/en/latest/index.html>
12. Plotly Open Source Graphing Libraries. [cit. 19.09.2021]. Dostupné z: <https://plotly.com/api/>
13. Welcome to HoloViews! — HoloViews 1.14.5 documentation. [cit. 17.10.2021]. Dostupné z: <https://holoviews.org/getting_started/index.html>
14. FLORIAN MOUNIER. Pygal — pygal 2.0.0 documentation. 2016 [cit. 08.09.2021]. Dostupné z: <http://www.pygal.org/en/stable/>
15. COREY, Michael. Turn your data into sound using our new MIDITime library. In: *Reveal* [online] [cit. 17.10.2021]. Dostupné z: <http://revealnews.org/blog/turn-your-data-into-sound-using-our-new-miditime-library/>
16. CUTTONE, Andrea. Geoplotlib - documentation. 5. 9. 2021 [cit. 08.09.2021]. Dostupné z: <https://github.com/andrea-cuttone/geoplotlib>
17. MUELLER, Andreas. *word\_cloud* [online]. 2021 [cit. 24.10.2021]. Dostupné z: <https://github.com/amueller/word_cloud>
18. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* [online] [cit. 30.10.2021]. Dostupné z: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed>
19. ASEEM KASHYAP. 8 Rules for optimal use of color in data visualization. In: *Medium* [online]. 28. 12. 2020 [cit. 09.09.2021]. Dostupné z: <https://towardsdatascience.com/8-rules-for-optimal-use-of-color-in-data-visualization-b283ae1fc1e2>
20. MORELAND, Kenneth. *Diverging Color Maps for Scientific Visualization*. Berlin, Heidelberg: Springer, 2009. Lecture Notes in Computer Science. ISBN 978-3-642-10520-3. DOI: [10.1007/978-3-642-10520-3\_9](https://doi.org/10.1007/978-3-642-10520-3_9)
21. KOVESI, Peter. Good Colour Maps: How to Design Them. *arXiv:1509.03700 [cs]* [online]. 2015 [cit. 30.10.2021]. Dostupné z: <http://arxiv.org/abs/1509.03700>
22. BARTRAM, Lyn, Abhisekh PATRA a Maureen STONE. Affective Color in Visualization. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 2017, s. 1364–1374 [cit. 09.09.2021]. ISBN 978-1-4503-4655-9. Dostupné z: <https://doi.org/10.1145/3025453.3026041>
23. YI, Mike. How to Choose the Right Data Visualization [online]. 27.2.2020 [cit. 31.09.2021]. Dostupné z: <https://cdn2.hubspot.net/hubfs/392937/How-To-Choose-The-Right-Data-Visualization%20(1).pdf>
24. A
25. A
26. A
27. A
28. A
29. a
30. a

# Přílohy

*Oskenované zadání práce*