Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

Data Science: Principy, technologie a znalosti

Bakalářská práce

Autor: Tomáš Brzek
Studijní obor: AI3

Vedoucí práce: prof. RNDr. Hana Skalská, CSc.

Hradec Králové Duben 2017

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 27.4.2017 Tomáš Brzek

Poděkování:

Mé poděkování patří prof. RNDr. Haně Skalské, CSc. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnovala.

Anotace

Tato bakalářská práce je zaměřena na představení oblasti Data Science jako nového přístupu k datové analýze. Úvodní část popisuje vznik a formování tohoto pojmu včetně náplně datové vědy, její porovnání vůči podobným zájmovým oblastem a aktuální stav této vědy v České Republice. V teoretické části práce jsou definovány pojmy, které jsou pro Data Science klíčové, také je vysvětlena forma uložení dat a rozveden proces datové analýzy. Dále se v teoretické části nachází možnosti pro vizualizaci výsledků z procesu datové analýzy a jednotlivé algoritmy, jež jsou k jejich dosažení využívané.

Jedním z hlavních cílů je vytvoření přehledu v pojmech souvisejících s Data Science a upřesnění procesů, jež za ní stojí. K příkladům v praktické části jsou využity programovací jazyk R a integrované vývojové prostředí RStudio. Hlavní náplň praktické části tvoří deskriptivní statistická analýza vybraných souborů.

Annotation

Title: Data Science: Principles, technologies and knowledge

The aim of this bachelor thesis is to introduce Data Science as a new approach to data analysis. The introductory part describes the origin and formation of this concept including its main objectives, its comparsion with similar science areas, and its current state in the Czech Republic. In the theoretical part of the thesis are defined key concepts that are crucial for Data Science, also explained forms of data storage, and data analysis process. Further, in the theoretical part there are data visualization options used for presenting the results from data analysis process and the individual algorithms which are used for their realization.

One of the main objectives is to create an overview of concepts related with Data Science, and clarify processes standing behind it. The programming language R and the integrated development environment RStudio are used for examples in the practical part. Main content of the practical part consists of a descriptive statistical analysis of selected files.

Obsah

Seznam obrázků

Seznam tabulek

Seznam diagramů

1 Úvod

Data jsou všude kolem nás, jen v různých podobách. V každodenním životě je možné se s daty setkat pravidelně, skvělým příkladem je nákupní seznam. Nejprve jsou na seznamu položky, co v domácnosti chybí, podle lístku je poté proveden nákup v obchodě. Při nákupu obsluha hotovostní přepážky oskenuje čárový kód z obalu výrobku a kasa zaregistruje cenu. Počítač na skladu položku odečte a případně upozorní manažera skladu, že zásoba klesá, a proto by se zboží mělo doobjednat. Kus informace z lístku končí pak jako součást tabulky pro manažera obchodního domu. Na cestě od tužky a papíru k manažerovi do kanceláře prošla data mnoha transformacemi, ať už byly tyto transformace řízeny lidmi nebo počítači. Data jsou jedním z nejcennějších nalezišť informací v oblasti podnikového rozhodování a kladení obchodních cílů.

Dle magazínu Harvard Business Review se pracovní pozice datového vědce jeví jako nejpřitažlivější prací 21. století. Podstatou Data Science je zarytí se do jádra problému a zkoumání dat pod jejich povrchem. Data mohou mít účel až ve chvíli, kdy jsou na ně položeny ty správné otázky, tyto otázky si datový vědec musí umět, jak vytvořit, tak zodpovědět. Klíč k úspěchu tkví v precizní vizualizaci nalezených výsledků a výstupů z celého procesu Data Science. To znamená, že zobrazení každé informace směrem k vedení musí být jasné a pádné.

Tato bakalářská práce si klade za cíl popsat principy Data Science. Popsat jednorozměrná a vícerozměrná data. Odhalit fenomén posledních let v podobě Big Data. Následně detailně prozkoumat celý proces datové analýzy, který začíná sběrem dat, pokračuje přes jejich čištění a explorační analýzu, a končí prezentováním výsledků. Prozkoumat formy prezentací výsledků a formy požadavků uživatelů dat na jejich zobrazení. Pomocí programovacího jazyka R a programu R-Studio namodelovat aplikační modely pro tuto práci.

Pro aplikační modely budou využita data z Českého statistického úřadu, konkrétně sčítání obyvatel z roku 2011 a z repositáře UCI Machine Learning Repostiory bude použit dataset Adult, který je druhý z nejpoužívanějších datasetů, hlavně díky počtu obsažených pozorování.

Struktura práce je volena s ohledem na postupné představování a uvádění do problematiky Data Science. Koncepce práce se ubírá takovým způsobem, aby člověk, který nikdy o pojmu Data Science neslyšel, byl schopný pochopit jak její hlavní náplň, tak podružné problémy.

Motivací pro zpracování práce pojednávající o Data Science, je uvedení tohoto nového přístupu k analýze dat. Osobní motivací byla především dychtivost prozkoumání a prostudování mě neznámé oblasti Data Science a programovacího jazyka R. Dalším důvodem je možnost vyzkoušení aplikace deskriptivní statistiky pomocí jazyka R na reálných datech.

2 Cíl práce a metodika zpracování

Hlavním cílem práce je prostudování a shrnutí problematiky Data Science do smysluplných celků a vysvětlení hlavních principů a metod Data Science pomocí literární rešerše.

Data Science se v informatice jako celek definoval do své nynější podoby pouze několik let nazpět a v této oblasti se jedná o nový termín. Z minulosti byla oblast působnosti Data Science rozdělená prostřednictvím několika dalších, jako jsou například Data Mining nebo Data Analysis a nebylo tak možné přesně vytyčit hranice této vědy až do nedávna.

Hlavní náplní této práce je zmapování minulosti a počátku vzniku a formování Data Science, dále popis jednotlivých polí působnosti a vysvětlení termínů tykajících se této vědy.

Přístup k vypracování tohoto přehledu byl zvolen především díky oboru, který studuji. Jedná se o informatický pohled na současný stav datové vědy, jelikož ta se prolíná mezi několika oblastmi, kde jednou z nich je informatika. Vzhledem k širokému záběru Data Science a s ohledem na můj studijní obor, bude do této oblasti nahlíženo spíše jednostranně, což může zabraňovat celkovému nadhledu nad problematikou. Výhodou tohoto přístupu je hlubší proniknutí z oblasti statistiky nebo programování.

V této práci je zmapována historie oblasti Data Science, jsou porovnány její nové přístupy k analýze dat vůči ostatním. Dále jsou vysvětleny Big Data, uvedeny dvě nejčastější formy uložení dat a jejich struktura, podrobně rozebrán proces datové analýzy a vizualizace výsledků a jako poslední teoretickou částí je představen programovací jazyk R i program R-Studio.

Po přečtení teoretické části by měl každý získat základní náhled do této oblasti a mělo by být snadnější si uvědomit proces, který stojí za pojmem Data Science.

V praktické části bude pomocí programovacího jazyka R provedena deskriptivní statistická analýza výběrového soboru. Cílem ukázkové aplikace je, pomocí základních prvků deskriptivní statistiky, porovnat nalezené hodnoty v souboru a výsledky následně vizualizovat.

Metody využité v praktické části obnášejí získání datových souborů a jejich popis. Následuje příprava dat, jejichž součástí je selekce a odstranění dat nepotřebných. Hlavní náplní praktické části je deskriptivní statistická analýza souboru dat a vizualizace výstupů dosažených touto analýzou pomocí grafů a diagramů.

3 Data Science

3.1 Historie

Počátek pojmu Data Science, má pouze krátkou historii, jelikož se objevil až kolem roku 2000. Spolu s historií pojmu magazín Forbes (10) spojuje i pokusy o jeho definování a také i jeho použití. Roku 1962 se statistik John W. Tukey zmínil o své zálibě v datové analýze a roku 1977 vydal knihu Exploratory Data Analysis, kde od sebe odlišil explorační a konfirmační analýzu dat, ale zároveň tvrdil, že by se měli držet ruku v ruce. V tomto roce je také vytvořena organizace The International Association for Statistical Computing, kdy jejím hlavním cílem je utvářet ucelenou statistickou metodologii.

Rok 1989 znamenal výrazný růst převážně v oblastech dobývání znalostí z databází a o pět let později, tedy v roce 1994, časopis BusinessWeek, zveřejnil článek na téma shromažďování informací o uživatelích a predikci nákupu zboží.

V roce 1996 je pojem Data Science poprvé použit v názvu konference v Japonsku. Data Science se dostal do nynější podoby až v roce 2001, kdy byla uvedena publikace, která vysvětlovala vznik tohoto nového přístupu k datové analýze a vytvořila plán, jak k Data Science přistupovat.

Rok 2002 představoval vznik žurnálu Data Science Journal, kde byly řešeny jednotlivé oblasti, jako jsou datové systémy a aplikace. Následně v roce 2003 vychází Journal of Data Science, kde je zmíněno, že vše, co se týká dat, je nazýváno pojmem Data Science.

3.2 Náplň datové vědy

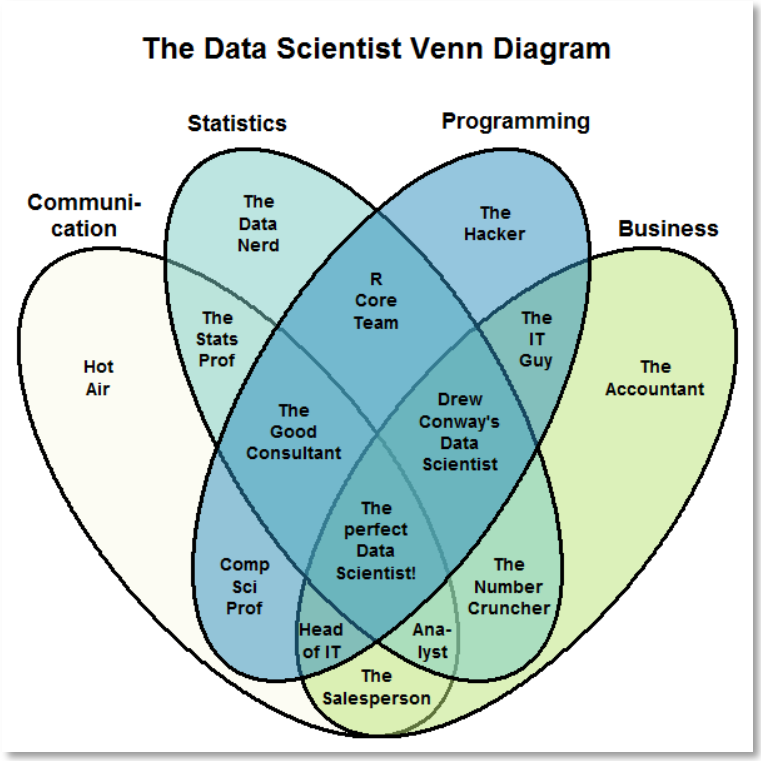
Věda o datech nebo datová věda, obojí je možné použít na vcelku moderní termín Data Science, avšak tento pojem se zatím do českého jazyka moc nevžil. V této kapitole bude vysvětleno, o čem Data Science je a jak by měl vypadat ideální datový vědec.

Tento obor bývá často zaměňován s obory, jako jsou datová analýza nebo datové inženýrství, a to hlavně z důvodu, že si jsou od pohledu velmi blízké, avšak Data Science se od těchto oborů z větší části liší. Dle serveru innoarchitech (11), který založil sám tvůrce článku, by se jako základní prvky dokonalého datového vědce, ať už pochází z jakéhokoliv profesního pozadí, měli pokládat tyto čtyři prvky:

- Obchodní doména
- Statistika a pravděpodobnost
- Počítačová věda a programování
- Verbální i psaná komunikace

Tyto hodnoty by měli datovému vědci pomoci k dosažení například lepšího podnikového rozhodování nebo dosažení stanovených obchodních cílů. Výborné protknutí těchto hodnot znázorňují i Vennovy diagramy, kde jejich verzí k Data Science

je nepřeberné množství, ale dle uvedených čtyřech stavebních kamenů, jej dle mého názoru nejlépe reprezentuje tato verze.



Obrázek 3.1 - Vennův diagram oblasti Data Science (autor: Stephan Kolassa, zdroj: http://www.prooffreader.com/2016/09/battle-of-data-science-venn-diagrams.html)

Jako perfektního datového vědce označuje tento diagram osobu, která je expertem hned ve čtyřech oblastech od komunikace, statistiky, programování až po obchod. Je však velkou výzvou někoho takového najít.

V mém studijním oboru na Univerzitě Hradec Králové s názvem Aplikovaná Informatika, jsem měl příležitost se setkat se všemi čtyřmi oblastmi z tohoto diagramu, ale nejvíce času jsem věnoval počítačovým sítím a programování, které jsou taky hlavní náplní studia. Mimo to jsem ve třetím ročníku, díky předmětu pravděpodobnost a statistika, nahlédl na data jako na zdroj informací, které se dají dále vyhodnocovat a zpracovávat. Tyto skutečnosti jsou především důvod, proč se tato moje práce dále odvíjí spíše informatickým pohledem na věc, a proč je praktická část tvořena pomocí programovacího jazyka R.

Mezi nejčastější cíle Data Science patří predikce, klasifikace, doporučení, detekce anomálií, rozpoznávání nebo segmentace či optimalizace. Aplikace každé kategorie z velké míry záleží na konkrétním plánu dané společnosti (11).

Velký ohled je nutné brát i na obratnost v komunikaci datového vědce. To je dáno hlavně tím, že se výsledky procesu Data Science prezentují uživatelům, kteří mohou mít jen minimální technologické vzdělání. Proto je nutné výstup z analýzy přizpůsobit obecenstvu tak, aby získalo stejný nebo alespoň podobný náhled na problém jako datový vědec.

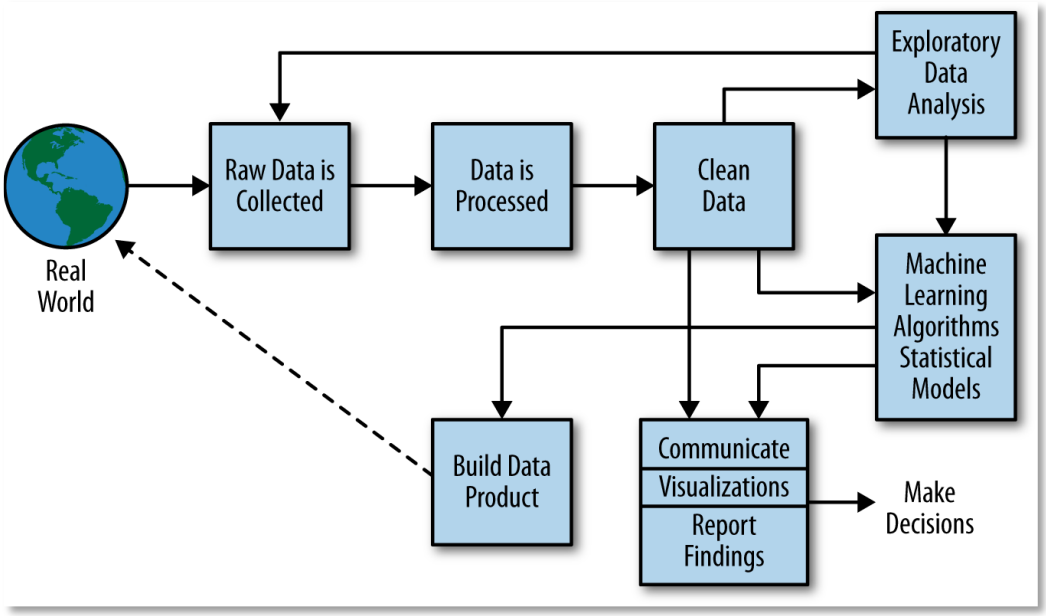
Porovnání datového analytika a datového vědce (11). Datoví analytici často nemají tak velké zkušenosti v programování a orientují se spíše směrem do více vizuálních nástrojů jako je IBM SPSS, SAS atp. Datoví vědci pro své cíle využívají především programovacího jazyka R nebo Pythonu. Dalším rozdílem je, že datový analytik většinou dostává úkoly a otázky od vedení, kdežto datový vědec sám ví, jaké cíle společnost má a otázky si proto pokládá sám.

Porovnání datového vědce a datového inženýra (11). Datoví inženýři jsou odpovědni především za architekturu dat či za připravení jejich infrastruktury. Podle innoarchitech bude jejich popularita v současné době Big Data stále stoupat. Musejí být mnohem více obeznámeni s možnostmi jejich uložení a navrhování designu databází. Vyplňují nefunkční požadavky jako škálovatelnost, spolehlivost, trvanlivost, dostupnost a zálohování. Oproti datovým vědcům se specializují z větší části na technické operace s daty.

3.3 Data Science v České Republice

Data Science se do České Republiky, jako pracovní pozice i jako pojem, integroval mnohem později, než tomu tak bylo u západních zemí. To je způsobené hlavně stářím této vědy. Vyhledávač Google najde relevantnější výsledky při zadání „datová věda“ než při zadání „věda o datech“, z čehož vyplývá, že se Data Science překládá spíše jako datová věda. Je ale nutné zmínit, že i při zadání prvního pojmu výsledky nepřekypují množstvím informací a spíše se zachovává originální název. Při hledání v těchto českých překladech pojmu na vyhledávači Google, ať už v prvním nebo druhém tvaru, se není možné dočkat více relevantních odkazů či stránek. Až na pár zpravodajských článků a jeden blog, který se věnuje přímo datové vědě, nic podstatného.

Zajímavějším tématem je sledování pracovních nabídek spojených s pojmem Data Science v České Republice. Při vyhledání pracovních pozic obsahujících pojem „Data Science“ na serveru indeed.com v Česku se vyskytne 419 nabídek, kde je z toho 301 v Praze, 44 v Brně a zbytek je buď celostátní nebo rozdělený do zbylých krajů. Ve výsledcích hledání se nachází i nadnárodní korporace jako Cisco Systems, Apple, Amazon nebo Barclays. Mnoho z nalezených nabídek tvoří i operátoři nebo banky. Zbytek jsou povětšinou firmy, zabývající se primárně logistikou, farmacií nebo vývojem software. Pouze pro srovnání, vyhledáním pracovních nabídek Data Science na americké verzi stránek indeed, dostaneme zhruba 127 tisíc výsledků a je možné si vybírat z korporací jako Microsoft nebo FedEx.



Obrázek 3.2 - Data science/Data analysis proces (zdroj: SCHUTT, Rachel a Cathy O'NEIL. Doing data science. ISBN 978-1-449-35865-5)

Člověk, který se rozhodne pro činnost v oblasti Data Science, musí být připraven na rozvíjení sama sebe ve čtyřech oblastech zároveň. Mnoho společností má na svědomí sběr dat svých uživatelů, otázkou však je, zdali jsou tato data správně využita nebo zdali jsou vůbec využita. Kolikrát se totiž v datech může skrývat rozšíření potenciálu společnosti. Tuto skutečnost si velké korporace byli nuceni uvědomit jako první a museli na ní jako první zareagovat. Možná i to byl důvod k tak rychlému růstu poptávky a zájmu o tuto oblast. Data Science je aktuálně v informatice často probíraným tématem a je možné si všimnout, že vzniká stále více pracovních pozic se stejnou nebo podobnou náplní.

4 Big Data

S neustálým rozvojem informačních technologií se nedávno objevil i vcelku nový pojem zvaný jako Big Data. Tento pojem lze definovat jako označení datasetů, které jsou natolik velké a komplexní, že si s nimi tradiční aplikace pro datové zpracování nemohou poradit v rozumném čase (12). Hlavními charakteristiky těchto dat se považují takzvané 4 V (13):

- *Volume (Objem)*: Odkazuje na velikost dat
- *Velocity (Rychlost)*: Jak rychle data dorazí a také jak rychle se ustálí
- *Variety (Typ)*: Data mají různou formu (Strukturovaná, nestrukturovaná/Text, multimédia)
- *Veracity (Věrohodnost, přesnost či správnost)*: Určuje nejistotu a nedůvěryhodnost směrem k datům díky jejich necelistvosti či nekonzistentnosti

Jako nejvhodnější příklady pro reprezentaci jsou společnosti, kde se Big Data vyskytují zcela běžně nebo jsou i jejích hlavním principem:

- Bankovní společnosti: Veškeré platby kreditní kartou a platby přes internet mohou být monitorovány za účelem prevence finančního podvodu či praní špinavých peněz
- Sociální sítě: Perfektní příklad, kde jsou Big Data hlavním produktem, čím více má sociální síť uživatelů, tím více má i dat o nich samotných
- Softwarové společnosti (např. Google): Chytré telefony jsou v dnešní době obdařené veškerými druhy senzorů, od akcelerometrů až po snímače otisků prstů, a proto velké společnosti vědí, kdy ráno uživatel zvedne telefon ze stolu, jakou cestou chodí do práce či jaké si prohlíží webové stránky s produkty, cíleně se pak dle toho může objevovat reklama při procházení internetu (na oblečení, pracovní pozice) atp.

Z příkladů je více než jasné, že sběr a následná analýza těchto dat může mít obrovský dopad na fungování a marketingový chod firmy. Proto je zájem a rozruch kolem Big Dat v současné době stále na vzestupu. Další definicí nám to potvrdí McKinsey Global report z roku 2011:

> *„Big Data jsou taková data, jejichž škála, rozdělení, diverzita a/nebo včasnost budou vyžadovat využití nových technických architektur a analýz které vnesou nadhled a otevřou nové zdroje pro obchodní hodnoty.“*
>
> *McKinsey & Co.; Big Data: The Next Frontier for Innovation, Competition, and Productivity*

Z výše uvedeného vyplývá, že něco jako Big Data je záležitostí poslední desítky, maximálně pár desítek let. Je nutné si však uvědomit, že obrovská kvanta dat byla generována lidmi již před začátkem našeho letopočtu, třeba v alexandrijské knihovně - tato knihovna byla již od 3. století př. n. l. považována jako centrum vzdělanosti až do

roku 48 př. n. l., to je přes 250 let. Sice nelze dokázat v jakém období a kolik přesně měla knihovna obsahovat svitků, ale uvádí se, že se mělo jednat o stovky tisíc až miliony. Všechna data na svitcích, stejně jako Big Data dnešní, nelze zpracovat v nějakém rozumném čase a tradičními metodami, proto nelze brát jako fakt či dogma, že jsou Big Data zbrusu novou věcí. Stejně tak, když data splňují podmínky 4 V, nelze o nich bezpodmínečně říci. že se jedná o Big Data.

Shrnutí poznatků z uplynulé kapitoly:

- Ceny senzorů jsou tak zanedbatelné, že je možné sbírat více dat než v minulosti.
- Ceny úložišť za poslední roky stále klesají (14). To znamená ukládání více dat bez užitku či smyslu.
- Lidé sdílejí stále více informací na sociálních sítích.
- Vědecký pokrok ve strojovém učení, které formuje základy pro techniky dolování znalostí.
- V případě, že dataset dosáhne určité velikosti, konvenční testy statistické významnosti postrádají smysl.

S čím je nutné vždy počítat (15):

- GIGO (Garbage In, Garbage Out) - Užitečnost dat závisí pouze a jenom na tom, jak pečlivě jsou sbírána, a poté předzpracována. Pokud jsou data na vstupu k ničemu, budou stejně tak data na výstupu.
- Pokud jsou v hledáčku anomálie, platí pravidlo, že čím větší je kolekce dat, tím větší šance na jejich nalezení.
- Propojením datasetů s jinými, je možno docílit zefektivnění hledání a nalezení výsledku či zcela nových pohledů na věc.

5 Popis jednorozměrných a vícerozměrných dat

5.1 Data

Data jsou pro následnou statistickou analýzu nejčastěji reprezentována (či transformována tak aby odpovídaly) maticí *n* x *d* o *n* řádcích a *d* sloupcích, kde řádky odpovídají objektům a sloupce jejich atributům či vlastnostem.
Matice *n* x *d* je dána následovně

$$D = \begin{array}{c|cccc} & X_1 & X_2 & \dots & X_d \\ \hline x_1 & x_{11} & x_{12} & \dots & x_{1d} \\ x_2 & x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & x_{n1} & x_{n2} & \dots & x_{nd} \end{array}$$

kde x_i značí i-tý řádek, což je i-tice daná jako

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id})$$

a kde X_j značí j-tý sloupec, což je j-tice daná jako

$$X_j = (x_{1j}, x_{2j}, \dots, x_{nj})$$

Záleží na oblasti aplikace pro kolekci dat nebo datový soubor, a proto se mohou řádky také často nazývat jinak než objekty (např. položky, záznamy, příklady, transakce atp.) a sloupce jinak než vlastnosti (např. dimenze, proměnné, atributy, pole atp.) Podle počtu objektů n je velikost dat a podle atributu d dimenze dat. Analýza jednoho atributu se nazývá jednorozměrná (univariační) analýza, současná analýza dvou atributů se nazývá dvourozměrná (bivariační) analýza a analýza dvou a více prvků je vícerozměrná (multivariační) analýza (1, s. 5).

	ID (X_1)	**VEK** (X_2)	**POHLAVI** (X_3)	**VYSKA** (X_4)	**VAHA** (X_5)	**POJISTENI** (X_6)
X_1	1	21	Muž	192	90	Ano
X_2	2	56	Muž	178	79	Ano
X_3	3	43	Žena	165	58	Ano
X_4	4	62	Muž	173	83	Ano
X_5	5	82	Žena	150	57	Ano
X_6	6	14	Muž	156	48	Ne
X_7	7	37	Žena	163	61	Ano
X_8	8	54	Žena	162	76	Ne
X_9	9	12	Muž	135	42	Ano
X_{10}	10	19	Muž	198	93	Ne

Tabulka 5.1.1

Tabulka 5.1.1 obsahuje výběr z databáze evidující záznamy o pacientech. Ukázková tabulka je matice o rozměrech 10 x 6. Každá položka je jeden pacient a jeho atributy jsou ID, VEK, POHLAVI, VYSKA (cm), VAHA (kg), POJISTENI. První řádek je 5-tice

$$x_1 = (1, 21, \text{Muž}, 192, 90, \text{Ano})$$

Dle výše uvedeného záleží, jakou oblast data reprezentují, a proto maticové uspořádání není vždy jasně daným pravidlem. Složitější kolekce dat jako je například text, obrázek, video, zvuk atp. potřebují speciální techniky pro datovou analýzu, není však nemožné tyto kolekce dat transformovat do maticového formátu. Pokud jsou data v tzv. surovém formátu, dají se transformovat do maticového mnoha metodami, kde jednou z nich je technika zvaná Extrakce příznaků. Například knihovna s hudbou by byla do matice transformována tak, že by jednotlivé hudební soubory tvořily řádky a sloupce jejich atributy, jako třeba frekvence, tón, bpm atd (1, s. 5).

5.2 Atributy

Pokud jsou data ve formě matice či tabulky, pak obsahují objekty a jejich atributy. Atributy se podle způsobu jejich měření nebo získání dělí na další typy, protože hodnotu může nabývat jak číslovka, tak text. Číslovka jako například výška, může být měřena v různých jednotkách (například stopy nebo centimetry). Některé atributy také mohou nabývat nějakých maximálních a minimálních hodnot (například věk), kdežto atribut ID nemusí mít maximální hodnotu vůbec stanovenou. Atributy se tedy třídí hlavně dle domény, tedy toho, jakou hodnotou jsou naplněna.

Číselný/Numerický atribut (1, s. 6), jinak zvaný kardinální atribut, je takový, který má reálnou číselnou hodnotu. Například věk s doménou (Věk) = ℕ, kde ℕ reprezentuje přirozená čísla. Dělí se v základu na dva typy:

- *Intervalově-měřené*: Pro tyto atributy dávají smysl pouze operace jako součet nebo rozdíl. Příkladem intervalového atributu je možné uvést dny v roce. Když je dnes 16. ledna a před 8 dny bylo 8. ledna, dává smysl říci, že je o 8 dní více než 8. ledna. Není však možné tvrdit, že je dvakrát více jak 8. ledna.
- *Poměrově-měřené*: U těchto atributů dávají smysl veškeré operace mezi hodnotami. Příkladem poměrového atributu je věk, kdy někdo, komu je 16 let může tvrdit, že je dvakrát starší než někdo, komu je 8 let.

Při důkladnějším pohledu se dají kardinální atributy dále dělit ještě na (2, s. 35):

- *Diskrétní*: Atribut, který je počitatelný či počitatelně nekonečný, tedy nabývá pouze určitých hodnot
- *Spojité*: Mohou nabývat jakýchkoliv hodnot – věk člověka, který se dá podle potřeby přesnosti měřit buď v letech, dnech, hodinách atd.

- *Dichotomická (Binární)*: Je speciální druh diskrétního či nominálního atributu, který nabývá pouze dvou hodnot

Kategorický atribut (1, s. 6) je takový, který má předurčené hodnoty předem. Typovým příkladem zde může být například pohlaví {Muž, Žena}, dosažený stupeň vzdělání atp. Opět se dále dělí na dva typy:

- *Nominální*: Hodnoty těchto atributů jsou neseřazené a dává smysl porovnávat pouze hodnoty, které jsou si rovny. Tento typ atributu je tedy například pohlaví, protože má smysl porovnat jeho hodnoty.
- *Ordinální*: Zde jsou hodnoty seřazené a porovnávání prvků je zcela běžné. Ordinálním atributem jsou třeba dosažené stupně vzdělání, kdy je například střední vzdělání považováno za vyšší než základní vzdělání).

Poslední klasifikací je rozdělení na **atributy závislé a nezávislé**. Hlavní povahou u této klasifikace je uvědomění si, co je „příčinou“ a co „následkem“. V případě dvou jevů, kdy jeden je příčinou a druhý následkem, je uveden nezávislý atribut jako příčina a jako atribut závislý je následek. Věta „Pokud bude zítra pršet, tak nepůjdu ven“ jasně představuje kauzalitu čili příčinu a následek. Rozhodnutí ovlivňuje počasí, konkrétně to že „bude zítra pršet“ je nezávislý atribut (či proměnná), a to že „nepůjdu ven“ je závislý atribut.

5.3 Popis jednorozměrných dat

Jedná se o jednu z nejjednodušších forem uložení dat. Kvůli jejich jednoduché struktuře není možné do nich vkládat žádné složité konstrukty. Jednorozměrná data toho mají mnoho společného s poli, které jsou primitivním datovým typem v programovacích jazycích. Mohou sloužit především jako prostor pro uložení číselných řad.

Jednou ze společných vlastností spolu s poli, je indexace jednotlivých elementů, tedy přístup k jednotlivým proměnným určitého pole je řešen pomocí tzv. indexu. Kupříkladu založení jednorozměrných dat v jazyce R:

```
> jednoRD <- sample(1:10, 10, replace = TRUE)
> jednoRD[3]
 [1] 3
> jednoRD
 [1] 9  3  3  7  5  2  3  9 10  5
```

První řádek vygeneroval pole náhodných deseti čísel s rozmezím od 1 do 10. Pomocí hranatých závorek a uvedení čísla je možné se zeptat na daný index v jednorozměrných datech, zde R vypíše 3tí pozici v poli *jednoRD*. Na posledním řádku je vypsaný vektor v se všemi svými prvky. Jednorozměrná data tedy jsou řádek elementů, které lze adresovat pomocí jejich pozice.

5.4 Popis vícerozměrných dat

Vícerozměrná, či pro začátek dvourozměrná, data jsou z pohledu uložení dat a jejích následné analýzy více k užitku a také je to nejčastější forma prezentace dat pro uživatele. Některé datové konstrukty totiž znesnadňují jejich uložení do jednorozměrné formy, a i přesto, že by taková možnost byla – například tabulka sčítání lidu v ČR by se dala reprezentovat jednorozměrně jako: *okres1, součetLidu1, okres2, početLidu2* atd., tak by se s přibývajícími atributy zvětšovala náročnost dělení pole, ale i jeho velikost a diverzita datového typu. Příklad založení matice v R:

```
> x <- c(1, 2, 3)
> y <- c(5, 6, 7)
> rbind(x, y)
  [,1] [,2] [,3]
x    1    2    3
y    5    6    7
> cbind(x, y)
     x y
[1,] 1 5
[2,] 2 6
[3,] 3 7
```

První dva řádky představují vytvoření obyčejného vektoru (či jednorozměrného pole) *x* a *y*. Poté je zde funkce rbind nebo cbind, která je schopna spojit vektory čísel do matice čili dvourozměrných dat. Záleží již na konkrétní funkci, je možné pozorovat, že funkce rbind spojí vektory řádkově, zatímco cbind spojí vektory do sloupců. Za povšimnutí stojí také fakt, že se jemně změnil výpis, kterým R prezentuje data. U řádků je uvedeno interpunkční znaménko čárky (,), čímž R signalizuje, že se jedná o matici, a že číslo zastupuje buď sloupec nebo řádek. Opět se pomocí této notace odkazuje na dané pozice v matici, jako např.:

```
> matice <- rbind(x, y)
> matice[, 3]
x y
3 7
> matice[2, 3]
y
7
```

Toto byl jednoduchý příklad, jak mohou vypadat dvourozměrná data. Problém ale může nastat tehdy, když se dimenze dat zvětšuje a roste počet pozorovaných nezávislých proměnných, tedy roste počet sloupců. Jako tomu může být například u analyzování zcela náhodného profilu na sociální síti. Takový profil obsahuje ohromné množství vlastností – jako například jméno, příjmení, datum narození, telefonní číslo, místo bydliště, studium, rodinné vztahy, oblíbené filmy/knihy/muzika. Toto je pouze zlomek výčtu informací, co o sobě může uživatel, například na Facebooku, uvést. Podobným způsobem je možné si uvést i vyhodnocení filmů od jednotlivých uživatelů ve filmové databázi. Touto cestou je možné sestavit velký dataset informací, které většinou už není jednoduché smysluplně zpracovávat.

Existují však metody, kterými lze upravit i vysoce dimenzionální data, a to takovým způsobem, aby se na těchto datech dali lépe provádět statistické analýzy či

extrakce jistých důležitých informací. Redukce dimenze tedy bývá zcela nezbytným krokem i u strojového učení (3), kde odstraňuje irelevantní data, zvyšuje efektivnost učení a zpřesňuje srozumitelnost výsledku.

Metody pro redukci dimenze se v oblasti data-miningu zpravidla dělí do dvou skupin (16, s. 2):

- *Selekce proměnných (feature selection)* – výběr pouze takových proměnných které dávají smysl
- *Extrakce příznaků (feature extraction)* – nahrazují pozorovaná data jejich kombinací, převádějí tato data do menší dimenze, kde je potřeba provést všechny výpočty, ale přesto je nutné pozorovat výsledky všech proměnných, kde podle zobrazení se dále rozdělují na:
  - *Lineární* – do této kategorie patří asi nejznámější metoda pro redukci dimenze PCA – Principal component analysis neboli Analýza hlavních komponent
  - *Nelineární*

6 Struktura dat

Při práci s daty je možné data nalézt v několika různých formách. Některé formy byly uvedeny, jsou jimi již zmíněný text, multimédia, finanční data a jiné. Tyto formy dat jsou pouze specifické příklady.

Data lze rozdělit do 4 primárních kategorií (4, s. 6-7):

1. *Strukturovaná data* – Sem jsou zařazena data, které mají definovaný formát a strukturu (jako třeba transakční data, CSV soubory, tabulkové soubory – obrázek č. 6.1, nebo tradiční datasety z relačních databází)
2. *Polo-strukturovaná data* – Jsou data v takovém formátu, který podporuje parsování (metody pro úpravu na strukturovaná data), mezi ně se řadí např. formát XML – obrázek č. 6.2
3. *Kvazi-strukturovaná data* – Textová data s nepravidelným formátem, mohou být s dostatkem úsilí a času naformátovány (data z clickstreamu – obrázek č. 6.3)
4. *Nestrukturovaná data* – Nemají předepsanou žádnou strukturu, hlavně díky jejich rozmanitosti, jsou to různé druhy obrázků, videí, textových dokumentů atp. – obrázek č. 6.4

Strukturovaná data

Table 1. Monthly Population Estimates for the United States: April 1, 2010 to December 1, 20

Year and Month	Resident Population	Resident Population Plus Armed Forces Overseas	Civilian Population
2016 [1]			
January 1	322 064 447	322 314 446	320 898 323
February 1	322 225 731	322 467 657	321 050 046
March 1	322 393 448	322 625 721	321 207 313
April 1	322 563 614	322 791 661	321 375 258
May 1	322 740 344	322 966 471	321 547 576
June 1	322 928 068	323 155 543	321 732 199
July 1	323 127 513	323 348 770	321 925 074
August 1	323 345 274	323 566 531	322 142 835
September 1	323 563 193	323 784 450	322 360 754
October 1	323 778 180	323 999 437	322 575 741
November 1	323 974 632	324 195 889	322 772 193
December 1	324 142 480	324 363 737	322 940 041

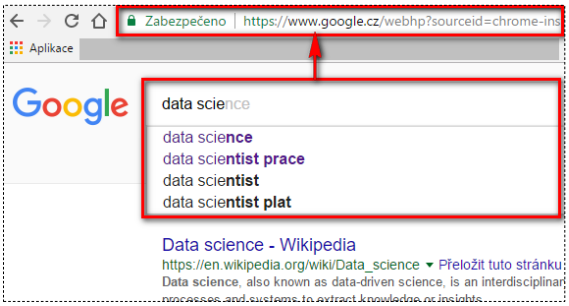
Obrázek 6.1 - Strukturovaná data (zdroj: Vlastní)

Polo-strukturovaná data

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" el
  <xs:element name="xypair">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xaxis"/>
        <xs:element ref="yaxis"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="xaxis">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="property"/>
```

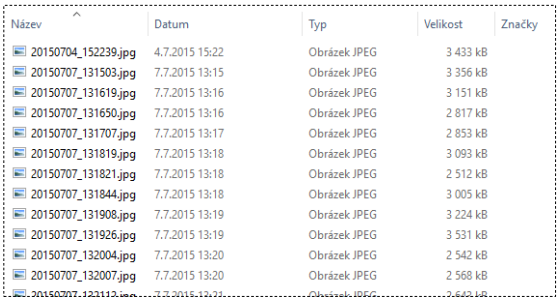
Obrázek 6.2 - Polo-strukturovaná data (zdroj: Vlastní)

Kvazi-strukturovaná data



Obrázek 6.3 - Kvazi-strukturovaná data (zdroj: Vlastní)

Nestrukturovaná data



Obrázek 6.4 - Nestrukturovaná data (zdroj: Vlastní)

Podstatným faktem ale je, že se data podle zdrojů nachází až z 80 % v nestrukturované formě (4, s. 5) (5, s. 22). Proto je možné nestrukturovaná data pokládat za nejpopulárnější formu uložených dat, což nemusí být ani tolik překvapivé, protože data, které jsou námi tvořena každý den, jako jsou například různé poznámky, prezentace, data při procházení webu atp., jsou nestrukturovaná. Strukturovaná data mají jasně daný

tzv. mustr, podle kterého jsou uložena. U nestrukturovaných dat není podle čeho se řídit, u relačních databází jsou jasně daná schémata či mapy databází, díky kterým je možné se v nich orientovat, v jazyce XML jsou to tagy, url odkazy z clickstreamu, lze taky po vynaloženém úsilí rozdělit ale u nestrukturovaných dat nic jako orientační schéma či pomůcky, nalézt nelze.

Poslední poznámka zodpovídá otázku, kde lze Big Data nalézt. Jedná se o dva hlavní datové zdroje – Interní/Externí. Konkrétnější příklady jsou v obrázku č.6.5.

Obrázek 6.5 - IBM. Where does big data come from? (zdroj: http://www.ibmbigdatahub.com/infographic/where-does-big-data-come)

7 Analýza dat

Pod pojmem analýza dat si je možné představit jakýsi základní přehled o datovém souboru, ze kterého je prováděna extrakce informací. Pokud jsou již nějaká data k dispozici, jejich shlédnutí umožní následné rozdělení dle našich požadavků do různých kategorií, s různým druhem vlastností. Prohlédnutím lze také nalézt anomálie, jako odlehlé hodnoty, duplikáty atp., toto ovšem nebude platit pro datové soubory s několika tisíci či desítkami tisíc záznamů. Další a detailnější přehledy už jsou poté získány pomocí statistických aplikací na datový soubor (6, s.1).

7.1 Proces datové analýzy

Pokud ovšem není k dispozici nic jiného než myšlenka, například na zlepšení obchodních strategií nebo provedení výzkumu, je nutné si položit otázku, co je potřeba zjistit a jakým způsobem to zjistit. Také se bude jinými způsoby poměřovat například průměrná výška člověka a jinými zase psychické či fyzické zdraví člověka, jeho inteligence, sebevědomí a jiné. Na všechny parametry se využívají úplně jiné míry, a proto je dobré si uvědomit veškeré překážky a problémy, co mohou v dané oblasti nastat. Nyní bude krok za krokem vysvětleno (7, s. 41-42), co vlastně datová analýza obsahuje, a jakým způsobem jsou její úkoly řešeny.

Požadavky na data

Ve chvíli, kdy jsou známy veškeré překážky, začíná nastavování požadavků, které jsou na data kladena, to znamená, že pro předchozí příklad by to mohlo vypadat následovně – výška člověka v centimetrech, inteligence jako hodnota IQ, zdraví – má mnoho aspektů (váha, hladina cukru v krvi, genetické vady atp.)

Sběr/Shromažďování dat

Je fází obnášející samotný proces sběru dat z libovolných zdrojů. Ty mohou být různého charakteru a původu. Může být řešen dotazníky, případně veřejně dostupnými informacemi. Taky se jedná o zmiňovaný sběr dat ze senzorů, internetu (online zdroje), videonahrávek, pohovorů atd. U internetových zdrojů je však potřeba si ověřit, jakým způsobem byly data sbírány, aby byly relevantní a nezavádějící.

Zpracování a sumarizace dat

Pokud byla předchozí fáze provedena svědomitě, mělo by se jednat o minoritní záležitost. Jde o umístění dat, ideálně do strukturované formy (tj. řádky a sloupce), pro jejich co nejjednodušší a nejpohodlnější analýzu.

Čištění dat

V případě, že jsou data ve vyhovujícím formátu, ještě to nemusí znamenat, že jsou správně. Podstatný krok, který je nutný udělat před samotnou analýzou dat, je jejich čištění. Jedná se o velice důležitou součást procesu analýzy, jelikož se při jejich shromažďování může lehce stát chyba, může dojít k mylnému zadání, založení

duplicitních pozorování či data mohou zcela chybět, opět u tohoto faktu velmi záleží na předchozích fázích.

Je potřeba pečlivě kontrolovat, zda jednotlivé proměnné odpovídají variantám, které jsou v dotazníku, jinými slovy probíhá ověřování, jestli se rozložení hodnot pohybuje pouze v rámci škály, ve které bylo měřeno (2, s. 76). Také je nutné si dát pozor na hodnoty, které leží až podezřele vysoko či nízko v oboru hodnot a jsou spíše nepravděpodobné. Datová analýza je nazývá odlehlé hodnoty či extrémně odlišné hodnoty a často bývají chybou například v přidání řádu do číslovky (platba byla ve výši 4 000/ platba byla ve výši 40 000 atp.). Pokud se však nejedná o chybu při záznamu dat, je potřeba se rozhodnout, jak s takovými hodnotami naložit. Tyto hodnoty mohou následně zkreslit průměr souboru, což je nežádoucí. U čištění dat se hodí pamatovat na pravidlo, které se nemění, tedy „ze špatných úsudků mohou vzejít jen špatné předpoklady“ (2, s. 76).

Explorační analýza dat

Kritická a nedílná součást analýzy, kterou definoval v knize Exploratory data analysis (8) její autor John Tukey. Je to přístup k datové analýze, také reprezentuje filozofii vykonávání statistiky a zahrnuje velkou škálu metod (hlavně grafických) (17) a to:

1. Maximalizovat vhled do dat/datasetu
2. Odhalit základní strukturu
3. Extrahovat důležité proměnné
4. Detekovat odlehlé hodnoty a anomálie
5. Otestovat základní předpoklady
6. Rozvinout model
7. Určit optimální nastavení faktorů

Hlavním aspektem EAD jsou grafické nástroje statistické analýzy, jako příklady jsou – box plot, histogram, metoda PCA (redukce dimenze) a další.

Modelování dat a algoritmus

Data jsou nyní v přijatelné formě a je možné na ně aplikovat potřebný algoritmus. Následně záleží pouze na tom, jaký problém je potřeba vyřešit, zda predikční či klasifikační problém nebo stačí pouze jednoduše statisticky soubor popsat. Jednotlivé algoritmy jsou popsány v kapitole Algoritmy pro datovou analýzu.

Interpretace, sdělení výsledků a vyvození závěrů

Po veškeré odvedené práci nastává čas k vyvození výsledků ze zkoumání, na datech prováděných. Ať už jsou ve finále data interpretována komukoliv, je nutné to provést v takové formě, v jaké je koncový uživatel vidět potřebuje.

Alternativou zde pak je vytvoření „datového produktu“, což si lze představit jako různé hodnotící algoritmy, spam filtry, doporučující systémy atp. Tento datový produkt se pak vrací zpět do reálného světa a data jsou tak využita

k produkování dalších dat, dalšími uživateli. Takto vzniká zpětnovazební smyčka, díky které je možné zjistit například – velké množství lidí poslechlo vaše doporučení a zakoupilo tento produkt (7).

7.2 Koncový uživatelé dat a vizualizace informací

7.2.1 Požadavky uživatelů dat

Když se jedná o interpretaci dosažených výsledků z datové analýzy, tak se jedná o vizualizaci dat někomu, v nějaké formě. Jaká to bude forma, záleží na koncovém uživateli, konkrétněji hlavně na jeho znalostech, protože určitě bude chtít jiné informace vidět manažer oddělení a jiné jeho zaměstnanec. Pokud se tato myšlenka nevezme v potaz, lehce pak může nastat následující situace. Uživatel či společnost má k dispozici velké množství dat, ale jsou jim k ničemu, protože si nejsou schopni pokládat ty správné otázky, na které by data měla být schopna odpovědět. Tento seznam zmíní 10 nejčastějších požadavků od uživatelů, na vizualizaci dat, které jsou výstupem z datové analýzy (18).

	NÁZEV POŽADAVKU	POPIS POŽADAVKU	PŘÍKLAD
1	**Získaní hodnoty**	Specifický soubor případů – najití atributu případů	*Jak dlouhý je film Titanic?*
2	**Filtrování**	Podmínky na atributy – najití případu který jim vyhovuje	*Jaké komedie vyhráli ocenění?*
3	**Vypočítání odvozené hodnoty**	Soubor případů – spočítat a agregovat jejich numerické hodnoty	*Kolik je v ČR výrobců automobilů?* *Průměrný příjem kalorií člověka za den?*
4	**Najití extrému**	Hledání extrémní hodnoty atributu v rámci datasetu	*Který režisér komedií by získal nejvíce ocenění?*
5	**Třídění**	Soubor případů – seřadit je dle ordinální hodnoty	*Řazení oblečení dle velikosti.* *Řazení slov dle abecedy.*
6	**Určit rozsah**	Soubor případů a zájmový atribut – najit rozpětí hodnot v rámci datasetu	*Jaký je rozsah délky filmů?* *Jaký je rozsah cen za notebooky?*
7	**Popsat rozdělení**	Soubor případů a kvantitativních atributů – charakterizovat rozložení atributů přes dataset	*Jaké je věkové rozdělení studentů na vysoké škole?*
8	**Najití anomálie**	Hledání a identifikace anomálií – odlehlých hodnot	*Jsou nějaké odlehlé hodnoty v mzdovém výpise zaměstnanců firmy?*
9	**Shlukování**	Soubor případů – hledání clusterů (shluků) atributů s podobnými hodnotami	*Je znám nějaký shluk stejně dlouhých filmů?*
10	**Harmonizace**	Soubor případů a dva atributy – jsou hledány smysluplná spojení mezi těmito atributy	*Je nějaká souvislost mezi množstvím cukru v potravině a obezitou?* *Mají různá pohlaví různé preference při placení v obchodě?*

Tabulka 7.2.1

7.2.2 Vizualizace informací

Tabulka výše zaznamenala konkrétní případy požadavků uživatelů dat, nyní následuje shrnutí (9) toho, co je potřeba učinit, když je následně vyžadována reprezentace či interpretace dosažených výsledků. Co je to vizualizace dat a co si před ní uvědomit.

Krok číslo jedna. Definovat problém. Jak jinak lépe zjistit jádro problému než strávit nějaký čas s koncovými uživateli připravovaných dat. Prozkoumat nač budou data potřeba a co vlastně budou reprezentovat. *„Proč je tato reprezentace potřeba? Je potřeba pro komunikaci něčeho? Něčeho nového? Či snad k ověření hypotéz?“*. Je nutné brát v potaz veškeré dovednosti uživatelů (zde je možno říci, že i lidský faktor) a vizualizační model tomu dostatečně přizpůsobit.

Krok číslo dva. Jaké povahy jsou data, které mají být reprezentována? Pro vizualizaci informací je jejich rozdělení následující, jsou data typu, kvantitativního (čísla, seznam čísel atd.), ordinálního (data mají svoje pořadí a dají se porovnávat – jako př. dny v týdnu) nebo kategorického (bez řazení – pro př. jména ulic ve městě).

Krok číslo tři je určení počtu dimenzí. Což se rovná počtu atributů v datasetu. Na dimenzionalitě dat závisí, jakým způsobem budou reprezentována. Dalším parametrem je, že atributy jsou závislé a nezávislé, proto dle počtu závislých atributů jsou děleny kolekce na univariační (jedna dimenze závisí na druhé), bivariační (dvě závislé dimenze), trivariační (tři závislé dimenze) a multivariační (4 a více dimenzí).

Krok číslo čtyři. Data mohou mít lineární formu (vektory, kolekce atd.), mohou být dočasná (mění se v čase), prostorová nebo geografická (jsou spojena s plánem budovy, místy na světě) hierarchická (data na disku, struktura organizace) či mohou být v podobě sítě (vztah mezi entitami).

Krokem číslo pět pak je rozhodnutí, jakým způsobem bude s daty nakládáno. Tyto typy akcí jsou rozděleny následovně. Vizuální reprezentace statická (neměnící se data na papíře či obrazovce), transformovatelná (uživatel si může zvolit jakým způsobem data modifikovat, upravit vstupy) nebo manipulovatelná (volba konečného zobrazení dat, přiblížení detailu v datech).

Pro zjednodušení a finální shrnutí je zde tabulka 7.2.2.

PROBLÉM→	TYP DAT→	POČET DIMENZÍ→	STRUKTURA DAT→	TYP INTERAKCE
Komunikace	Kvantitativní	Univariační	Lineární	Statická
Průzkum	Ordinální	Nivariační	Dočasná	Transformovatelná
Potvrzení	Kategorické	Trivariační	Geografická	Manipulovatelná
		Multivariační	Hierarchická	
			Síť	

Tabulka 7.2.2

7.3 Algoritmy pro datovou analýzu

Volba použitého algoritmu závisí pouze na tom, čeho má datová analýza dosáhnout, a hlavně v jaké formě je cílový dataset. Odpověď na otázku, čeho má datová analýza dosáhnout, implikuje pokládání těch správných otázek na data. Je tedy více než

nutné se zaměřit na oblast dat, ze které jsou k dispozici a dle toho jejich obsah zpracovávat. Stručně řečeno – volba algoritmu záleží na potřebách a cílech analýzy co daná společnost potřebuje. Níže budou opsány a vyjmenovány jedny z nejčastěji využívaných a nejvíce známých algoritmů (4) v oblasti Data Science.

- **Shluková analýza**
  - shlukování již bylo zmiňováno v kapitole Požadavků uživatelů dat
  - jedná se o hledání shluků podobných objektů
  - algoritmus strojového učení – technika bez učícího souboru, jednotky stejné skupiny musí být odlišné od jednotek ostatních skupin
  - zájmu a určí jejich rozdělení
  - populární metoda – k-průměry (k-means)
  - oblasti použití – procesování obrázku, medicína, segmentace zákazníků
- **Asociační pravidla**
  - deskriptivní metoda pro určování vztahů v datasetu – nekontrolovatelná
  - často využívané pro dolování transakcí z databází
  - populární a nejzákladnější algoritmus Apriori
  - oblasti použití – analýza nákupního košíku, clickstream analýza, nástroje pro doporučení produktů při nakupování online
- **Regrese**
  - je používána pro zjištění vlivu, které má soubor proměnných na výsledek proměnných, které jsou v oblasti zájmu
  - proměnné v oblasti zájmu jsou závislé proměnné, ty další jsou nezávislé či vstupní
  - regresní analýza odpovídá na otázky typu – *„Jaký je očekávaný příjem člověka?“*
  - regresní analýza řeší úlohu nalezení modelu (nejčastěji v matematickém vyjádření), kterým lze popsat vztah vstupních nezávisle proměnných na závisle proměnnou, popis závislosti uvažuje náhodnou chybu, která je zahrnutá v modelu
  - oblasti použití lineární regrese – trh s nemovitostmi, předpovídání poptávky na trhu
  - oblasti použití logistické regrese – medicína (reakce na léčbu), možnost schválení půjčky klientovi, předpovězení defektu výrobku
- **Klasifikace**
  - Vytvoření tzv. třídiče s ohodnocenými příklady, podle něj se poté hodnotí další neviděné objekty
  - kontrolovatelná metoda – začíná s předznačenými prvky (s učícím souborem)
  - široce využívané hlavně v predikci
  - oblasti použití – spam filtry, diagnóza pacientů se srdečním onemocněním

    - rozhodovací stromy (predikční stromy) – stromová struktura příčiny a následku
    - populární algoritmy – ID3, C4.5
    - Naïve Bayes – metoda založená na Bayesově theorému
- **Analýza časových řad**
    - snaží se o namodelování základní struktury pro poznatky sbírané v čase
    - oblasti použití – slevy pro zákazníky (oblečení v určitou roční dobu), plánování náhradních dílů pro výrobky
    - model ARIMA – Autoregressive Integrated Moving Average
    - využívá Box-Jenkinsonovu metodologii
- **Textová analýza**
    - reprezentace, procesování a modelování textových dat pro získání vhledu
    - nedílnou součástí je dolování textu a hledání relací mezí vzorci
    - její nevýhodou je vysoká dimenzionalita dat
    - MapReduce & Hadoop
    - Hadoop je framework, který nabízí zpracování obrovských datasetů na mnoha zařízeních rozdělených do clusterů
    - hlavní komponentou Hadoopu je HDFS (Hadoop distributed Filesystem) – dokáže spolehlivě skladovat data, každý soubor je alespoň třikrát replikován na nezávislých zařízeních
    - Map-Reduce je metodou pro distribuování, která je založena na rozdělení problému (Map), vyřešení po částech a složení všeho zpět dohromady (Reduce)
    - oblasti použití – gigantické korporace jako LinkedIn nebo Yahoo!
- **Databázová analýza**
    - popisuje analýzu v rámci databázové aplikace, odstraňuje potřebu přesouvat data do statistického nástroje
    - je schopna poskytnout výsledky v reálném čase
    - oblasti použití – fraud detection (zkoumání podvodných transakcí), doporučování produktů na základě historie, již zmiňovaná cílená reklama
    - hlavním jazykem pro analýzu databází je SQL

Analýza dat je denní problematikou „datového vědce“ a vyplňuje zde svůj plný potenciál. Složitost datové analýzy se skrývá z velké části ve složitosti dat, která jsou poskytnuta a jaké jsou na ně kladené nároky. Datová analýza si většinou klade za cíl vylepšit obchodní politiku společnosti, to znamená, že veškerá zjištění, co bude mít za výsledek, je nutné investovat zpět do společnosti ať už ve formě business plánu nebo nějakých složitějších implementací, jako jsou KPI (Key Peformance Indicator – Klíčový Indikátor Výkonu), která jsou často využívána například v call-centrech.

8 Programovací jazyk R & RStudio

Programovací jazyk R (19), neboli česky takzvané „Erko“, je prostředím a nástrojem pro výpočet statistické analýzy dat a jejího grafického zobrazení. Jedná se o programovací jazyk, který je podobný jazyku S, R je pod záštitou GNU project, což je stručně řečeno svobodný software, který mohou uživatelé svévolně upravovat dle svých potřeb a poté ho distribuovat (20). Jazyk R sám o sobě má velké množství statistických nástrojů, ale jeho nejsilnější stránkou, také hlavně díky licenci, pod kterou vznikl, jsou knihovny. Jejich výhody však budou zmíněny později, nyní něco krátce k historii (21) a k tomu, jak R vznikalo.

8.1 Historie a vznik R

Historie a vznik jazyka R se datuje již do roku 1990, kdy se setkali pánové Ross Ihaka a Robert Gentleman. Rozhodli se, že společně napíší program, který bude sloužit k jejich soukromým testovacím účelům.

Robert Gentleman se roku 1992 přesouvá za Ihakou na oddělení statistiky Aucklandské Univerzity, kde společně chtějí vyvinout jazyk dostatečně dobrý na to, aby s ním byli schopni vyučovat základy statistiky na univerzitě. Rozhodli se, že převezmou syntaxi jazyka S, který byl vyvinutý v Bell Laboratories a jako název dají R, který i zároveň označuje první písmeno obou jejich křestních jmen.

V roce 1994 vzniká počáteční verze jazyka a po zvážení různých možností distribuce dávají jazyk pod licenci GPU a začínají s jeho distribucí prostřednictvím internetu, zároveň bylo založeno emailové vlákno, kde mohli uživatelé komunikovat s vývojáři.

Rok 1996 však ukázal, že založení zmíněného vlákna nebyl nejlepší způsob, jak se s uživateli spojit, jelikož autoři byli v neúnosné záplavě emailů s popisem bugů (chyb aplikace) a nápady na zlepšení. Spravovat emailové vlákno se stalo nemožným, a proto bylo na čase přibrat další vývojáře.

Do roku 1997 se datuje složení vývojářského jádra jazyka R. Pánové Kurt Hornik a Fritz Leisch založili CRAN což je známé jako The Comprehensive R Archive Network (doslovně přeloženo jako rozsáhlý síťových archiv R) a slouží jako celosvětový FTP a webový server, který ukládá a shromažďuje nejaktuálnější verzi jazyka R a jeho dokumentaci.

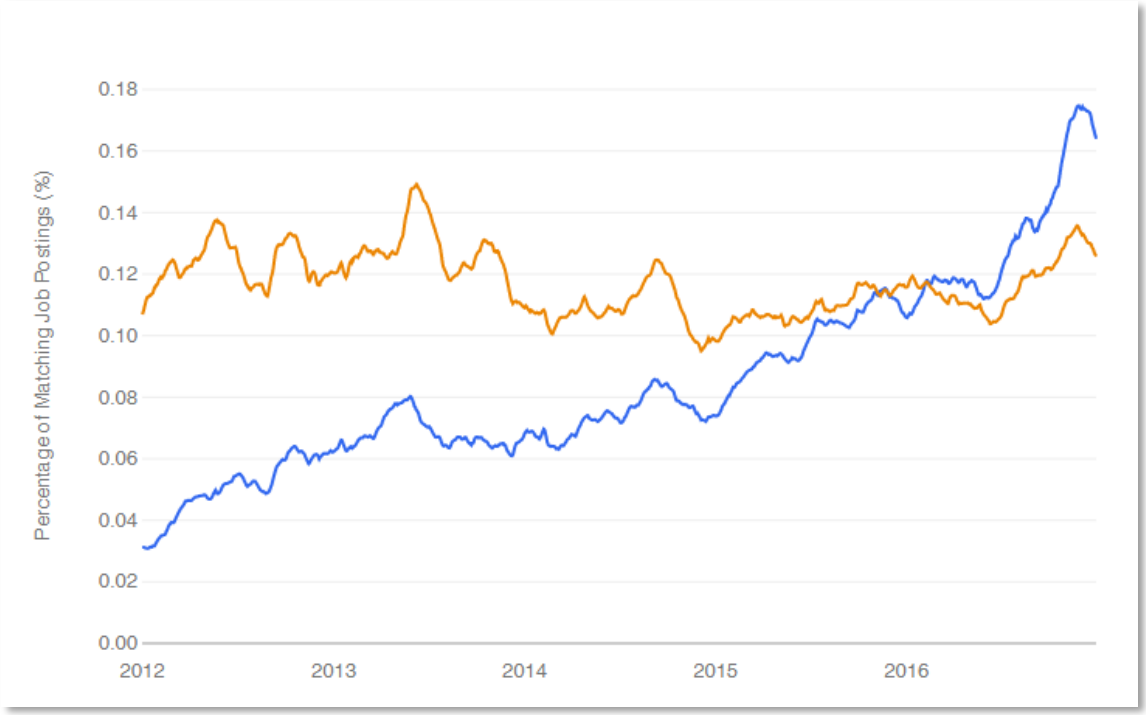
29. únor 2000 je datem oficiálního vydání první verze jazyka R 1.0.0. Tým je složen z necelých dvaceti vývojářů, kde několik z nich jsou široce známí statistici.

Nyní se R nalézá na CRANu ve verzi 3.3.3 pro operační systémy Windows, OS X (Mac) a Linux. Verze 3.0.0 byla uvedena v roce 2013 a do aktuální verze prošla již sedmnácti aktualizacemi.

V tuto chvíli zvládá standardní distribuce R (základ a doporučené knihovny) veškeré základní statistické operace (výpočet průměru, mediánu, rozptylu atd.), matematické operace, rozdělení pravděpodobnosti, strojové učení, Big Data analýzu a v neposlední řadě graficky vizualizovat statické a dynamické grafy či jiné formáty (JPEG, BMP, TIFF atd.) (22).

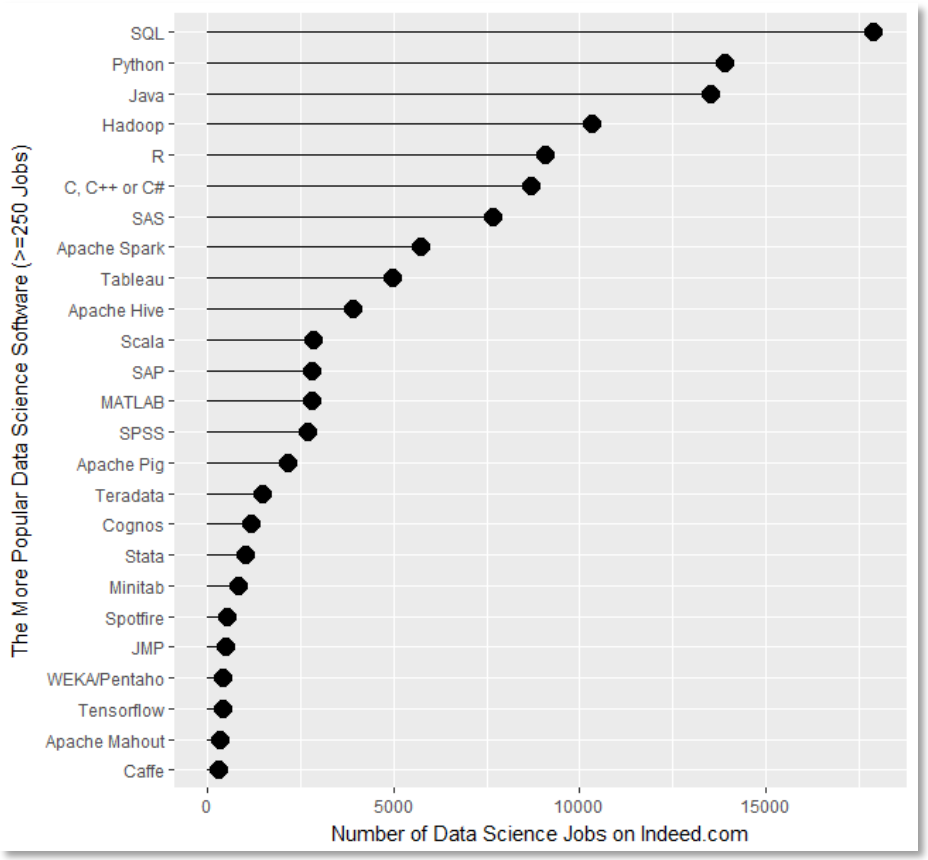
8.2 Proč používat R?

V posledních letech je Data Science často zmiňované hlavně v souvislosti se zaměstnáním. Ať už je na něj nahlíženo, jakkoliv často, je ve spojení s programovacím jazykem R. R je nástroj pro datovou analýzu a jeho popularita s časem pouze roste. Svědčí o tom fakta z blogu Roberta A. Meuchena, který zpracovává pravidelná srovnání nástrojů pro statistickou analýzu. Na jeho stránkách je možné se dočíst nespočet různých informací ohledně porovnání procent nabídek prací s různými nástroji pro datovou analýzu ale také je k vidění spousta grafů, které tato fakta potvrzují. Obrázek č.8.1 ukazuje že R (modrá křivka v grafu) předčilo zhruba v prvním čtvrtletí roku 2016 jazyk SAS (oranžová křivka v grafu) v procentech nabízených pracovních pozic s tímto nástrojem a jeho popularita se prozatím drží stále nad SAS, graf je aktuální ke dni 28.2.2017.



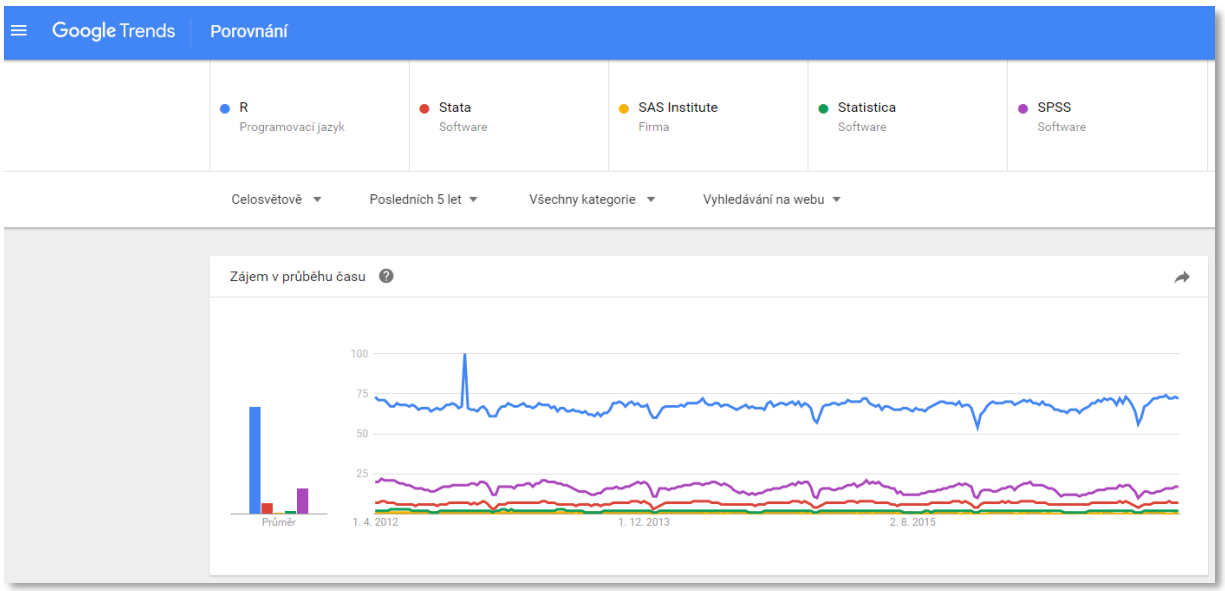
Obrázek 8.1 - Graf procentuálního porovnání nabídky pracovních pozic (Zdroj: http://r4stats.com/2017/02/28/r-passes-sas/)

Další zajímavou informaci je možné dohledat v obrázku č.8.2 (zdroj: http://r4stats.com/2017/02/28/r-passes-sas/). V oblasti pracovních pozic, konkrétně pro Data Science, překročilo R program SAS. Meuchen tvrdí že je to první report, kde se tato skutečnost stala, ale upozorňuje na ten fakt, že se to týká opravdu jen prací v oblasti Data Science, kdyby se prý jednalo o vypisování reportů o datasetech, je možné najít dvakrát tolik pracovních pozic pro SAS. Z grafu je taktéž patrné, že stále nejžádanější, je obecně známý databázový jazyk SQL, který tvoří nejfundamentálnější vrstvu přístupu k datům. Poté dále následuje Python, Java, ale i Hadoop.



Obrázek 8.2 - Popularita nástrojů pro Data Science (zdroj: http://r4stats.com/2017/02/28/r-passes-sas/)

Jako poslední důkaz pro podložení tvrzení o neklesající popularitě R, jsou zde trendy v hledání na Google, a to v období od roku 2010 do současnosti. Jedná se o porovnání počtu hledání programovacího jazyka R, Staty, SAS, nástroje Statistica a SPSS v celosvětovém měřítku. Zatímco ostatní statistické softwary a jazyky doznávají jednotek hledání týdně, či v případě SPSS je průměrný počet vyhledávání kolem dvaceti, u jazyka R je tento počet blížící se či překračující sedmdesát.



Obrázek 8.3 - Snímek z Google Trends pro porovnání statistických nástrojů (zdroj: https://trends.google.com/trends/)

Opodstatněním důležitosti těchto tvrzení je to, že pokud chce člověk hledat uplatnění v jakékoliv oblasti, je nutné, aby se naučil věcem, které mu budou nejvíce k užitku, a které pozvednou jeho cenu na trhu práce.
Shrnutí důvodů pro využívání tohoto programovacího jazyka je následující:

1. GPU Licence – opensource
2. Multiplatformní – Windows, Linux, MacOS
3. Popularita
4. Síla nástroje – zvládá mnoho druhů operací s mnoha různými typy dat
5. Podpora rozdělení na více clusterů či distribuci na více jader
6. Implementace mnoha druhů statistických implementací (regrese, Bayes atd.)
7. Široká podpora, jak vývojáři, tak komunitou (blogy, Q&A na StackOverflow)

Jedním z největších a nejhlavnějších důvodů jsou knihovny. Na CRANu je možné nalézt několik tisíc uživatelsky vytvořených knihoven (packages), kde některé z nich byly vytvořeny prominentními zástupci v daných oborech.

Dosud byla zmiňována pouze pozitiva, ale R má tak jako každý jazyk své určité nevýhody. Dle článku (23), kde figuruje jistý Matt Adams (Data Scientista na Code School) se píše, že R čelí velkým problémům se správou paměti i přesto že již na tento problém bylo a je vydáváno několik snah o nápravu. Tyto problémy mohou být způsobeny tím, že jsou všechny datové objekty nahrávány do RAM paměti a ta poté představuje svojí velikostí limit pro data, s jakými může R pracovat. Dále Roger Peng (zkušený veterán a programátor v oblasti R) poukazuje na fakt, že není možné R integrovat do webového či back-endového prostředí z důvodu chybějícího zabezpečení, na které se při vývoji tohoto programovacího jazyka v podstatě vůbec nedbalo. Tento problém však dneska z části řeší rozmanitá veličina cloudových řešení.

8.3 Proces učení se novému jazyku

R je stejně jako například Python nebo Javascript jazykem interpretovaným. Takový jazyk je nutné nechat projít *interpreterem* (tlumočníkem), který zpracuje kód do instrukcí pro počítač. Výhodou interpretovaných je, že nemusí projít procesem kompilace jako je tomu u kompilovaných jazyků, nebo také platformní nezávislost. Nevýhodou může být fakt, že je kód zpracován tak, jak lidově leží a běží, což může znamenat větší zabrání výpočetního výkonu a větší šanci pro zpracování chyb. Kompilované jazyky obsahují kompilátor, který kontroluje chyby již při kompilování kódu.

Softwarové prostředí jazyka je založeno na příkazové řádce. Ta umožňuje uživateli zadávat příkazy, jedná se jak o primitivní operace +, -, * či / tak složité jako je třeba lineární regrese a očekávat výsledky. Prostředí dále umožňuje práci s datasety, spojování více dat do jednoho dokumentu, evaluace dat z dokumentu a další. Samozřejmostí je i množnost vytváření vlastních funkcí. R je také jazykem procedurálním, což znamená, že vyžaduje přesnou posloupnost kroků pro vykonání nějaké úlohy. Hlavním rozdílem je to, že využívá operace pro práci nad daty, kdežto standardní OOP (Objektově orientované programování) spojuje jak data, tak operace jako součásti objektů.

Při porovnání s jazyky, které doteď znám (tj. C# a Java) se kód podstatně svojí syntaxí liší. Totiž, pokud v Javě nebo v C# je potřeba uložit a vypsat notoricky známou větu (řetězec) pro první seznámení s programem „Hello World“, jazyky vybízí k založení proměnné typu String s libovolným názvem, to označí fakt, že je proměnná řetězec a je naplněna slovy „Hello World“. V těchto jazycích musí každý objekt (místo v paměti které má svoji adresu) reprezentovat nějakou třídu. Jazyku R je však jedno, jaký to bude datový typ, a proto lze jednoduše založit proměnnou a rovnou jí naplnit potřebným výrazem, ať už je to číslo, text či jeden znak. Z pohledu začátečníka je toto nevynucené určování datového typu snazší na pochopení, ale pro někoho, kdo již předtím programoval, to může být nejednoznačné. Jako příklad:

Kód v jazyce Java

```
String retezec = "Hello World";
System.out.println(retezec);
```

VÝSTUP: Hello World

Kód v jazyce C#

```
string retezec = "Hello world";
Console.WriteLine(retezec);
```

VÝSTUP: Hello World

Kód v jazyce R

```
> retezec <- ("Hello World")
> print(retezec)
```

VÝSTUP: [1] "Hello World"

Tabulka 8.3.1

U R tedy může *retezec* (ve kterém je uloženo „Hello World“) znamenat jakýsi „název“ který reprezentuje skutečnost objektu. V takto dynamicky napsaném jazyce je možné, aby v jednu chvíli proměnná reprezentovala text a v druhé vektor čísel.

8.4 Základní příkazy, operace a datové typy

8.4.1 Základní příkazy

V poslední kapitole bylo zmíněno, že v R datové typy nejsou podstatné pro ukládání čehokoliv do proměnných. Toto tvrzení stále samozřejmě platí, ale ještě nebylo specifikováno, co je vše vůbec možné uložit a jakým způsobem. Proměnná představuje „název“ pomocí které je možné jí volat. Nejzákladnější příkaz představuje písmeno c (znamenající *combine*). Je to funkce sloužící pro spojení čísel do listu, v R se především používá název vektor. Pro přiřazení hodnoty do proměnné pomocí funkce c je ještě nutno použít symbol „<-“. Syntaxe je následující:

```
> promenna <- c(9, 8, 7, 6, 5)
> promenna
[1] 9 8 7 6 5
```

Byla založena proměnná s názvem *promenna* do které byl přiřazen vektor pěti čísel spojen funkcí c. Na jednotlivá čísla se lze odkazovat i pomocí indexů.

```
> promenna[3]
[1] 7
```

Pro shrnutí všech primitivních funkcí, které nabízí základní balíček jazyka R, je uvedena tabulka pokrývající tyto funkce a zároveň vysvětlující jejich použití.

sum() – sečte hodnoty vektoru/ů

```
> promenna1 <- c(1, 2, 3, 4, 5)
> sum(promenna)
[1] 35
> sum(promenna, promenna1)
[1] 50
```

data.frame() – udělá dataframe z vektorů

```
> ramec <- data.frame(promenna,
promenna1)
> ramec
  promenna promenna1
1        9         1
2        8         2
3        7         3
4        6         4
5        5         5
```

mean() – průměr

```
> mean(promenna)
[1] 7
```

range() – minimum a maximum

```
> range(promenna)
[1] 5 9
```

var() – variance/rozptyl

```
> var(promenna)
[1] 2.5
```

sd() – směrodatná odchylka

```
> sd(promenna)
[1] 1.581139
```

str() – struktura dataframu

Vysvětlivky: 5 pozorování 2 proměnných
Obe proměnné jsou numerické

```
> str(ramec)
'data.frame':   5 obs. of  2
variables:
 $ promenna : num  9 8 7 6 5
 $ promenna1: num  1 2 3 4 5
```

summary() – statistický popis dataframu

Vysvětlivky: minima a maxima
hodnota prvního a třetího kvartilu
průměr a rozptyl

```
> summary(ramec)
    promenna   promenna1
 Min.   :5   Min.   :1
 1st Qu.:6   1st Qu.:2
 Median :7   Median :3
 Mean   :7   Mean   :3
 3rd Qu.:8   3rd Qu.:4
 Max.   :9   Max.   :5
```

quantile() – hodnoty kvantilů v matici	**lentgth()** – počet prvků ve vektoru

`> quantile(ramec, probs=c(0.25,0.50,0.75))`
`  25%  50%  75%`
`3.25 5.00 6.75`
	`> length(promenna)`
	`[1] 5`

Tabulka 8.4.1

8.4.2 Datové typy

V jazyce R datové typy nepodléhají tolik vynucené syntaxi, jak to může být u ostatních programovacích jazyků. Dalo by se říci, že je v tomto dosti benevolentní a příkazy typu:

```
> a <- c("Hello", "World")
> a[1]
[1] "Hello"
> a[0]
character(0)
> a <- c(33, 44)
> a[1]
[1] 33
> a[0]
numeric(0)
```

jsou ekvivalentní a správně zapsané. Jediný rozdíl, kterého si lze všimnout je následující: pokud se indexuje na nultý prvek vektoru a[0], tak ten říká jak jsou data uložena. Základními datovými typy se tedy dají označit řetězce a čísla.

Dalšími možnými způsoby uložení jsou faktory, data framy (ve kterém byla prováděna předchozí demonstrace), dále jsou samozřejmostí logické operátory (TRUE/FALSE) či tabulky nebo matice.

8.4.3 Základní operace

Jakmile jsou definovány základní datové typy a je instancován nějaký vektor čísel, je nad ním možné vykonávat základní matematické operace.

```
> a <- c(1, 2, 3, 4)
> a
[1] 1 2 3 4
> a + 10
[1] 11 12 13 14
> a - 15
[1] -14 -13 -12 -11
> a * 4
[1]  4  8 12 16
> a / 5
[1] 0.2 0.4 0.6 0.8
```

Funkce pro mocniny, logaritmy či exponenciály jsou v R přítomny také.

```
> sqrt(a)
[1] 1.000000 1.414214 1.732051 2.000000
> exp(a)
```

```
[1]  2.718282  7.389056 20.085537 54.598150
> log(a)
[1] 0.0000000 0.6931472 1.0986123 1.3862944
```

Oba typy operací je poté možné skládat.

```
> (a+7)/(sqrt(10-b)*6-1/4)
[1] 1.3913043 1.0928588 0.9859692 0.9361702
```

Jelikož jsou veškeré operace v R prováděny na bázi jednotlivých elementů, tak pokud jsou sčítány vektory, je nutné, aby byly stejné délky.

```
> b <- c(9, 8, 7, 6)
> a + b
[1] 10 10 10 10
```

V neposlední řadě má v sobě R i třídící funkce.

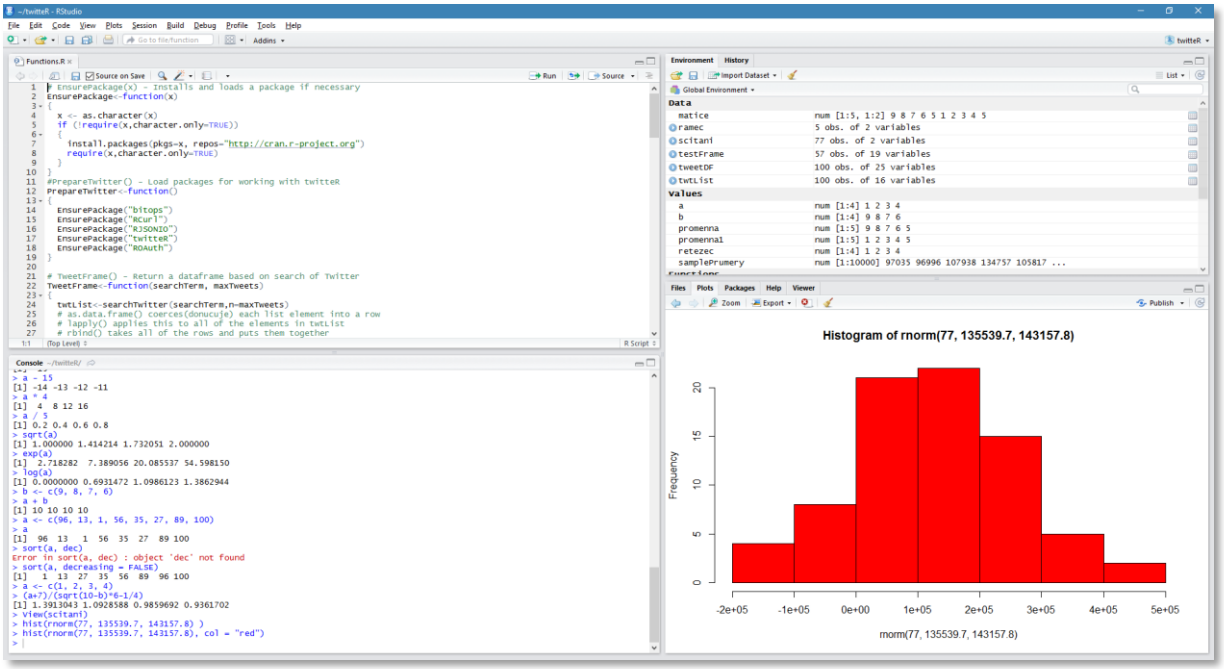
```
> a <- c(96, 13, 1, 56, 35, 27, 89, 100)
> a
[1]  96  13   1  56  35  27  89 100
> sort(a, decreasing = FALSE)
[1]   1  13  27  35  56  89  96 100
```

Základní balíček funkcí obsahuje dále i pravděpodobnostní rozdělení jako jsou: Normální rozdělení, Binomické rozdělení, chí-kvadrát a další.

Tuto kapitolu nejlépe shrnuje přísloví „účel světí prostředky“. Prostředkem pro účely Data Science je programovací jazyk R dostačující. R je jen příkazovou řádkou a do nynější chvíle s ním tak bylo i nakládáno, ale to se časem může stát komplikací z hlediska usnadnění práce, dokumentace nebo správy kódu.

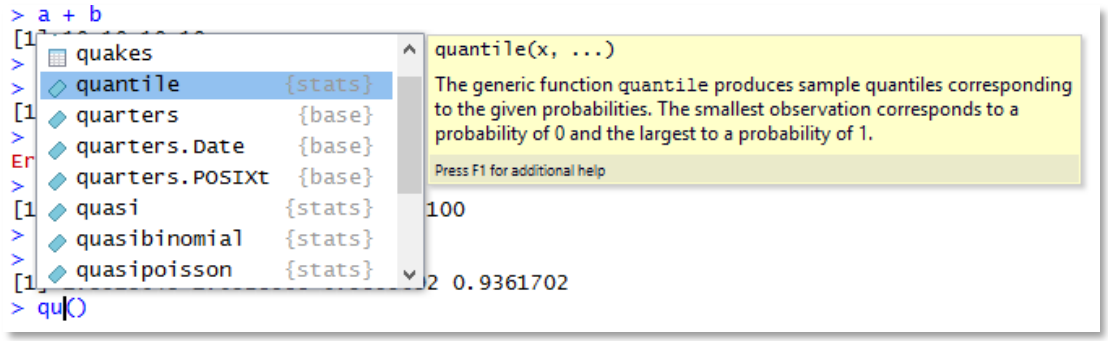
8.5 RStudio

Existuje velké množství IDE pro jazyk R. IDE je zkratkou pro Integrated Development Environment. IDE slouží jako vylepšené vývojové prostředí, které vývojářovi usnadňuje či automatizuje mnoho operací, které by jinak musel dělat ručně. Pro potřeby aplikace jsem se rozhodl využít software RStudio. Po prvním otevření RStudia je možné vidět 4 okna (obr. č. 8.4).



Obrázek 8.4 - Prostředí RStudia (zdroj: Vlastní)

Levý horní roh slouží pro vytváření a editaci nových skriptů v R či pro zobrazení datasetů. Vlevo dole je klasická příkazová řádka, kterou obsahuje R jako takové. Zde se zadávají veškeré příkazy a na rozdíl od samotného R, zde je software RStudio schopný i napovídat (obr. č. 8.5). Funkce napovídání neboli dokončování kódu, je velice přínosná hlavně ve směru urychlení programování či popsání metody, kterou je v plánu využít.



Obrázek 8.5 - Code Completion – napovídání k dokončení kódu RStudia (zdroj: Vlastní)

Vpravo nahoře pak lze nalézt okno Environment, kde se nalézají veškeré založené proměnné v pracovním prostředí a záložku History, která obsahuje historii prováděných příkazů. Vpravo dole je záložek hned několik, ale nejzajímavější je záložka Plots, kde se zobrazují vizualizační prvky jako histogramy atp. Dále záložka Packages, která je jakousi knihovnou stažených balíčků a rozšíření připravených k použití. Prostřednictvím této záložky se dají instalovat další balíčky pro rozšíření funkcí R. Celkové prostředí programu RStudio je intuitivní a jednoduché.

8.5.1 Vytvoření funkce a nového skriptu

Základní funkce již není nutno zmiňovat, ale může nastat situace, kdy potřebnou funkci program R nenabízí, jako například funkce s modem. Založení obyčejného R skriptu a naprogramování vlastní funkce summary bude obsahem této kapitoly.

```
MojeSummary <- function(vstup)
{
  prumer <- mean(vstup)
  minimum <- min(vstup)
  maximum <- max(vstup)
  prvnikv <- quantile(vstup, probs = 0.25)
  tretikv <- quantile(vstup, probs = 0.75)
  rozptyl <- var(vstup)
  smerod <- sd(vstup)
  hodnoty <- c("Prumer", "Min", "Max", "25%", "75%", "Var", "SDO")
  vysledky <- c(prumer, minimum, maximum, prvnikv, tretikv, rozptyl,
smerod)
  df <- data.frame(hodnoty, vysledky)
  return(print(df))
}
```

Postupně byly založeny proměnné a naplněné hodnotami. Vektor *nadpisy* reprezentuje textový popis pro jednotlivé hodnoty a vektor *vysledky* tyto hodnoty spojuje. Poté je vytvořen dataframe *df* do kterého jsou nadpisy a výsledky spojeny pro přehledné zobrazení v tabulce. Následně je už volána pouze metoda return pro navrácení výsledků, což je příkaz print(df). Nyní v porovnání s originální funkcí summary.

```
> a
[1] 1 2 3 4
> MojeSummary(a)
  hodnoty vysledky
1  Prumer 2.500000
2     Min 1.000000
3     Max 4.000000
4     25% 1.750000
5     75% 3.250000
6     Var 1.666667
7     SDO 1.290994
> summary(a)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.75    2.50    2.50    3.25    4.00
```

Výsledky jsou s funkcí summary identické, ve vlastní funkci MojeSummary je pouze výhodou, že může poskytnout výsledky v přesnější formě a obsahuje navíc

rozptyl a směrodatnou odchylku. Konstrukce vlastní funkce není nijak složitá a skript může obsahovat podobných funkcí hned několik. Opět se tento parametr bude odvíjet od potřeb na implementaci.

8.5.2 Knihovny

Asi se nedá říci, že by programovací jazyk R bez knihoven nemohl existovat, ale určitě je možné tvrdit, že by bez nich nebyl tím, čím je dnes. Sami zakladatelé tvrdí, že knihovny jsou jedním z klíčových faktorů pro jeho úspěch (24). Díky licenci, pod kterou R funguje, je možné, aby si jakýkoliv uživatel napsal svojí vlastní knihovnu v tomto jazyce a poté ji dal kolovat mezi ostatní uživatele. Package, jak je jejich originálním názvem označují zakladatelé lehce upravovatelnou kolekci funkcí a datasetů, jejichž standardy spravuje CRAN. CRAN zároveň umožňuje i sdílení jednotlivých knihoven privátně, tedy například se spolupracovníky nebo pouze pro vlastní potřebu.

Přidání a instalace knihovny není nic složitého. Příkaz install.packages zajistí stažení knihovny

```
> install.packages("[NÁZEV PACKAGE]")
```

a příkaz library poté zařadí knihovnu mezi používané v aktuální instanci programu RStudia.

```
> library("[NÁZEV PACKAGE]")
```

Nejpoužívanější knihovny reprezentuje vždy pár nejznámějších a nejvíce používaných knihoven dle směru, co je potřeba v R udělat. Jejich seznam je (25):

- Načtení dat
  - RODBC, RMySQL, RSQLite – pro čtení dat z databází
  - XLConnect, xlsx – čtení a zápis do formátu Windows Excel
- Manipulace s daty
  - dplyr – sumarizace, reorganizace a spojování datasetů
  - stringr – práce a operace s textem
- Vizualizace dat
  - ggplot2 – velmi slavná knihovna pro stylizaci grafů
- Modelování dat
  - zde záleží na typu algoritmu – caret pro klasifikační modely, randomForest pro strojové učení, mgcv pro aditivní modely regrese
- Report výsledků
  - shiny, R Markdown, xtable
- Prostorová data
  - sp, maptools, maps, ggmap
- A další…

Díky knihovnám je tedy možné R využít mnoha směry. Není důvodem je nevyužívat, snad jen pouze upozornit na fakt, že díky tomu, že jsou knihovny tvořené uživateli samotnými, se může lehce stát, že jsou nekompletní či nemají jistou záruku správného výstupu. Proto je výhodnější si knihovny vybírat dle popularity a zkušeností od ostatních uživatelů. Nejvíce populární knihovny jsou průběžně i často aktualizovány a rozšiřovány o nové funkce.

9 Aplikace

9.1 Příklad I

Pro první příklad jsem se rozhodl jako ukázku syntaxe a vlastností programovacího jazyka R demonstrovat na běžném příkladu se sčítáním obyvatel v ČR. Ze stránek ČSÚ (Český statistický úřad) jsem si stáhl veřejně dostupná data sčítání lidu z roku 2011, jedná se o soubor s 11 základními údaji (dalo by se říci atributy) z České Republiky, konkrétně jde o celkové počty obyvatel, počty mužů a žen, dělení obyvatelstva dle věku atd. Řádky tvoří jednotlivé kraje, okresy a poté i obce. Z tohoto dokumentu jsem vyfiltroval pouze údaje o okresech a zanechal sloupec s celkovým počtem obyvatel, pro minimalizaci množství informací hned na úvod. Následovat bude už pouze jednoduchá příprava dat a základní statistická deskripce datového souboru.

V první řadě bylo nutné v XLS dokumentu ze statistického úřadu upravit sloupce pro snadnější následující orientaci v datasetu. Také bylo nutné pomocí jednoduchého skriptu v MS Excelu 2016 odstranit diakritiku, jelikož s tou si R neumí poradit. Následně je vše připraveno a je možné data nahrát do R.

```
scitani <- read.DIF("clipboard", transpose = TRUE)
```

Příkazem read.DIF byla data nahrána ze schránky a parametr transpose = TRUE zajišťuje správné přečtení formátovaného excelového dokumentu. Náhled na data v R:

```
> scitani
        V1             V2
1       okres Benesov  95459
2       okres Beroun   86160
3       okres Kladno   158799
```

a tak dále…

Připadalo mi, jako by ČSU počítal s následnou analýzou jejich dat, jelikož v porovnání například s americkými daty pro sčítání obyvatel, ČSU neodděluje řádově statisíce tečkou či čárkou. Z toho důvodu není nutné sloupec s údaji o obyvatelích nijak formátovat. Nyní však k určení průměru, mediánu a modu, tedy středních hodnot datového souboru a poté výpočet rozptylu a směrodatné odchylky.

```
> mean(scitani$V2)
[1] 135539.7
> median(scitani$V2)
[1] 110522
//Zde bylo nutno využití vlastní funkce, R v základu modus nepodporuje
> MyMode(scitani$V2)
[1] 95459
> var(scitani$V2)
[1] 20494161957
> sd(scitani$V2)
[1] 143157.8
```

Jak je možno vidět z výsledků poskytnutých jazykem R, průměrný počet žijících obyvatel, je po zaokrouhlení 135540 obyvatel v jednom kraji se směrodatnou odchylkou

143157,8. Další věc, v čem R exceluje, je zobrazování dat. Pro kohokoliv s nedostatečnou představivostí, to může být aspekt, který lze ocenit ze všeho nejvíce. Pokud není jasné, proč je vlastně směrodatná odchylka 143 tisíc je tady funkce hist().

```
> hist(scitani$V2, breaks = 100, xlab = "Počet obyvatel", ylab =
"Počet okresů", main = "Histogram sčítání obyvatel", col = "green")
```

VÝSTUP:

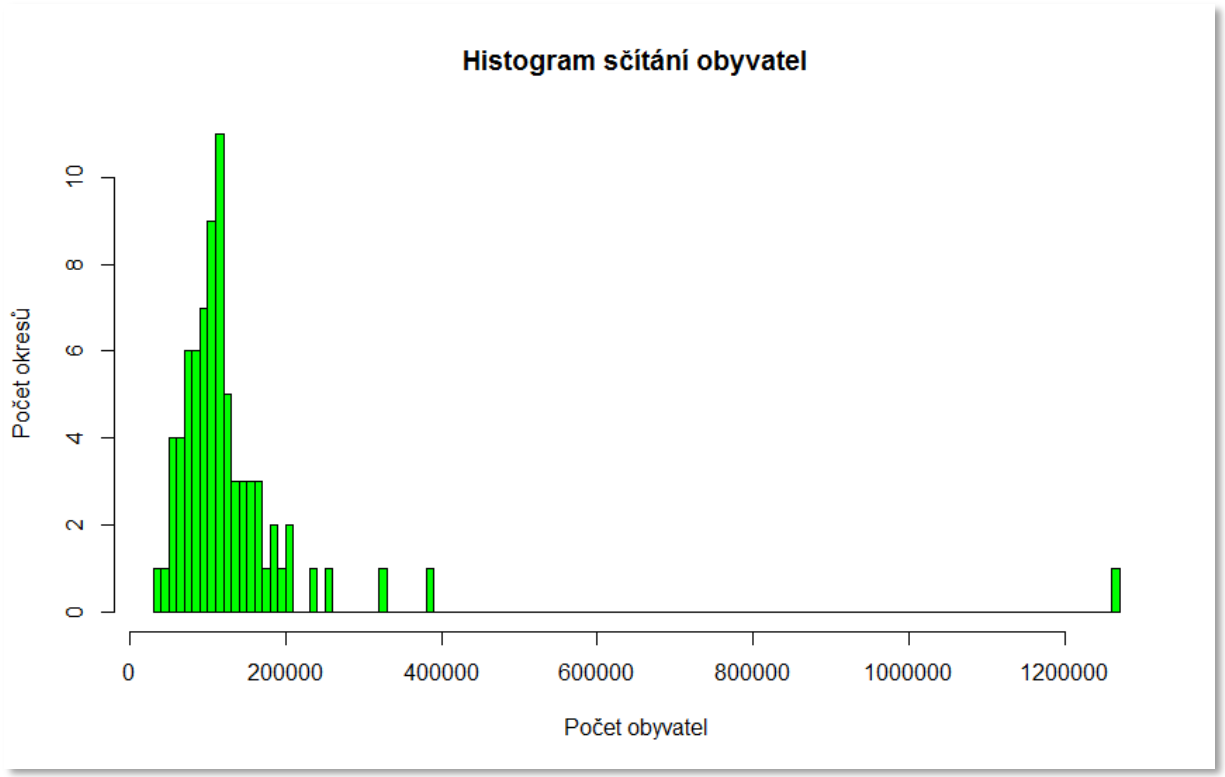


Diagram 9.1.1 - histogram sčítání obyvatel, vygenerován pomocí funkce hist() v R

Parametr breaks v příkazu zajistil detailnější rozvržení v histogramu jednotlivých okresů, dalšími parametry už byly pouze pojmenovány osy a název histogramu nebo změněna jeho barva. Každopádně histogram pomohl s představou rozdělení okresů a nyní je jasně vidět, že nejvíce krajů čítá kolem 150 tisíc obyvatel a outlierem neboli odlehlou hodnotou je okres Praha.

Na datech bude proveden ještě jeden experiment (15. s. 54), a to pomocí funkce sample(), která umožňuje vybrání libovolného počtu vzorků z datasetu, bude provedeno 5 výběrů a navíc tento celý příkaz bude zaobalovat funkce průměru (mean), aby byla zjištěna průměrná hodnota těchto 5 výběrů. Následně funkce replicate() zajistí zopakovaní funkce dle zadaného počtu opakování (v tomto případě bude počet opakování třeba 10000).

```
> samplePrumery <- replicate(10000, mean(sample(scitani$V2,size = 5,
replace = TRUE)),simplify = TRUE)
```

Pomocí tohoto příkazu byla založena proměnná *samplePrumery* s požadovanými výběry, replace = TRUE u funkce sample zajistí, aby se vzorky nahrazovaly a poté simplify = TRUE u funkce replicate znamená, že bude na výstupu figurovat jednoduchý vektor s čísly.

```
> length(samplePrumery)
[1] 10000
```

Je možné ověřit že se v proměnné *samplePrumery* opravdu nachází 10000 průměrů.

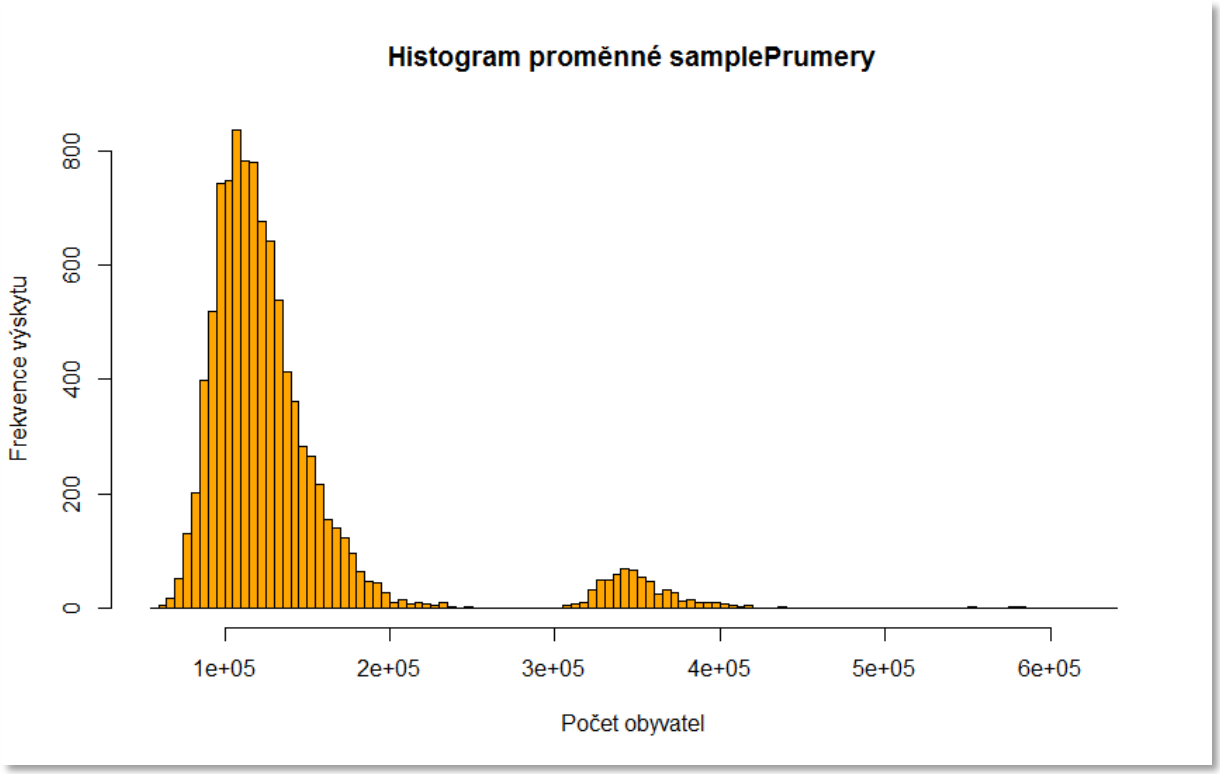


Diagram 9.1.2 - histogram proměnné samplePrumery, vygenerován pomocí funkce hist() v R

Jak lze vidět zprvu na histogramu nejvíce průměrů vybraných z datasetu bylo opět kolem 120ti tisíc obyvatel, přičemž při velkém množství opakování se našlo několik výjimek a byly vybrány i průměry vyšší. Nyní ke kvantilům.

```
> summary(samplePrumery)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  58830  102800  118500  135300  140300  638700
```

Funkce summary poskytla zajímavé informace. Již bylo řečeno, že maxima ve výši 638700 bylo dosaženo pomocí opakovaného tažení okresu Praha do vzorků, který několikrát navýšil běžný průměr. Součástí metody vzorkování tedy leží v opakování počtů pokusů na tolikrát, že se stane extrém. Kvartil 75 % napověděl, že pouze 25 % vzorků je vyšších než 140300. Opačně kvartil 25 % říká, že je šance 1 ku 4em, že bude nový tažený vzorek menší jak 102800.

```
> quantile(samplePrumery, probs=c(0.25,0.50,0.75))
     25%      50%      75%
102761.1 118548.9 140288.8
```

Funkce quantile(), má jemně odlišné výsledky, což je způsobeno funkcí summary(). Tady si lze udělat přesnou představu nad tím, kde jsou hranice kvantilů. Není však nutno se zde limitovat pouze na nejznámější kvartily a medián, dalšími známými kvantily jsou například hranice 2.5% a 97,5%.

```
> quantile(samplePrumery, probs=c(0.025, 0.975))
     2.5%     97.5%
 81369.78 352560.47
```

Z výstupu vyplývá, že je 2.5 % šance že bude další vzorek menší jak 81369,78 nebo větší než 352560,47. Doteď nebyla vypočítána směrodatné odchylka. Jejím výpočtem bude zároveň ukončen i experiment.

```
> sd(samplePrumery)
[1] 63056.44
```

Směrodatná odchylka neboli střední chyba průměru určuje, jak moc jsou hodnoty odchýleny od průměru, její hodnota je 63056,44.

Tato demonstrace si kladla za cíl ukázat alespoň základní sílu jazyka R nad výběrem údajů z datasetu a jejich hodnocením. Co může znamenat další nevýhodu pro R je fakt, že bez elementárních statistických znalostí je mnohem náročnější se s podobnou analýzou utkat.

9.2 Příklad II

Ve druhém příkladu bude snaha o hlubší prozkoumání datového souboru. Bude analyzován pomocí statistických prvků a budou u něj pokládány další výzkumné otázky.

V první řadě je nutné získat vhodná data, která budou dále zpracovávána. Z repositáře pro strojové učení UCI Machine Learning Repository jsem vybral aktuálně druhý nejpopulárnější dataset. Jedná se o Adult Dataset (26). Dle charakteristik na UCI je dataset multivariační (konkrétně 15 atributů), obsahuje celkově 48842 instancí/pozorování, z čehož budou použity pouze instance testovací, kterých je 32561 a nachází se v něm i několik chybějících hodnot. Datum přispění datasetu do repositáře je 1.5.1996, tedy nejedná se o nejaktuálnější data.

Následuje přidání dat do R, k čemuž využiji funkci read.table.

```
> adult <- read.table("C:/Users/Brzda/Desktop/BakPráce/R-
Studio/Adult/adult.data", header = FALSE, col.names = c("AGE",
"WRKCLASS", "FNLWGT", "EDU", "EDUNUM", "MARISTAT", "OCCUP", "RELAT",
"RACE", "SEX", "CAPGAIN", "CAPLOSS", "HPW", "NATCOUNT", "INCOME"))
```

Funkce read.table zajistí vyčtení tabulkových dat, kde parametr header = FALSE říká, že data neobsahují popisnou hlavičku. Proto volím pojmenování jednotlivých sloupců parametrem col.names. Jelikož jsou původní data ale dělena čárkou, tak budou s čárkou i přečtena, proto následuje jednoduché čištění dat.

```
> adult <- lapply(adultCopy, function(x){gsub(",", "", x)})
```

Příkaz lapply aplikuje vybranou funkci na celý dataframe – jedná se o funkci gsub, která dokáže nahradit vybraný znak za libovolný jiný znak. V tomto případě je nahrazena čárka řetězcem, jež neobsahuje nic, tedy ničím. Nyní je nutné zkontrolovat datový typ jednotlivých sloupců příkazem str.

```
> str(adult)
'data.frame':  32561 obs. of  15 variables:
 $ AGE      : Factor w/ 73 levels "17","18","19",..: 23 34 22 37 ...
 $ WRKCLASS: Factor w/ 9 levels "?","Federal-gov",..: 8 7 5 5 5  ...
 $ FNLWGT   : Factor w/ 21648 levels "100009","100029",..: 20430  ...
 $ EDU      : Factor w/ 16 levels "10th","11th",..: 10 10 12 2 10 ...
 $ EDUNUM   : Factor w/ 16 levels "1","10","11",..: 5 5 16 14 5 ...
 $ MARISTAT: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5
...
 $ OCCUP    : Factor w/ 15 levels "?","Adm-clerical",..: 2 5 7 7 ...
 $ RELAT    : Factor w/ 6 levels "Husband","Not-in-family",..: 2  ...
 $ RACE     : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 3 ...
 $ SEX      : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 1 2 ...
 $ CAPGAIN  : Factor w/ 119 levels "0","10520",..: 34 1 1 1 1 1 1 ...
 $ CAPLOSS  : Factor w/ 92 levels "0","1092","1138",..: 1 1 1 1 ...
 $ HPW      : Factor w/ 94 levels "1","10","11",..: 35 5 35 35 ...
 $ NATCOUNT: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 40 ...
 $ INCOME   : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 ...
```

V datasetu se opravdu nachází 15 atributů a konkrétně jsou to tyto:

- AGE (věk)
- WRKCLASS (pracovní pozice)
- FNLWGT (dle dokumentace se jedná o speciální vážený průměr – nebude využito)
- EDU (dosažené vzdělání)
- EDUNUM (číselná reprezentace vzdělání – nebude využito)
- MARISTAT (rodinný stav)
- OCCUP (povolání)
- RELAT (druh rodinného stavu)
- RACE (rasa)
- SEX (pohlaví)
- CAPGAIN (kapitálový zisk)
- CAPLOSS (kapitálová ztráta)
- HPW (odpracovaných hodin týdně)
- NATCOUNT (původní národnost)
- INCOME (příjem)

Nyní budou ze souboru odstraněny sloupce dat, které nebudou využity. Jedná se opět o jednoduché příkazy se specifikací sloupce v hranatých závorkách a přiřazení hodnoty NULL.

```
> adult[, 3] <- NULL
> adult[, 4] <- NULL
```

Z dostupných dat funkce přečetla správně veškeré atributy, ale zcela jistě některé z nich není potřeba ukládat do datového typu faktor, a proto bude výhodnější z atributů jako AGE, FNLWGT, CAPGAIN, CAPLOSS a HPW udělat obyčejné integery pomocí krátké vlastní funkce.

```
naCislo <- function(vstup)
{
   vstup <- as.vector(vstup)
   vstup <- as.integer(vstup)
   return(vstup)
}
> attach(adult)
> adult$AGE <- naCislo(AGE)
> adult$CAPGAIN <- naCislo(CAPGAIN)
> adult$CAPLOSS <- naCislo(CAPLOSS)
> adult$HPW <- naCislo(HPW)
```

Příkaz attach zařídil připnutí datasetu do prostředí což znamená, že není nutné při každém volání atributu z datasetu vypisovat jeho jméno před atribut. Pro převedení hodnot na integery bylo nutné vytvořit vlastní funkci naCislo, z důvodu ulehčení opakování stejných příkazů dokola.

Nyní jsou hodnoty, alespoň tam kde je to třeba, převedeny do lépe uchopitelných datových typů a je možné začít se statistickou analýzou.

Dokumentace datasetu tvrdí, že byl dataset tvořen dle podmínky (AGE > 16), neboli věk je větší než 16 let. Funkce vytvořená v kapitole 8.5.1 pomůže ke zjištění hodnot všech pozorování.

```
> MojeSummary(AGE)
  hodnoty  vysledky
1  Prumer  38.58165
2     Min  17.00000
3     Max  90.00000
4     25%  28.00000
5     75%  48.00000
6     Var 186.06140
7     SDO  13.64043
```

Minimální věk v datasetu je 17 let a maximální 90 let, přičemž průměrný věk je 38,6. Rozptyl je ve výšce 186, což signalizuje, že se jedná o data napříč všemi věkovými kategoriemi, ale také, že průměr nebude v tomto datasetu nijak extrémně zavádějící hodnotou.

```
> mfv(AGE)
[1] 36
> sum(AGE==36)
[1] 898
> sum(AGE<mean(AGE))
[1] 17508
> 100*(17508/length(AGE))
[1] 53.76985
> sort(unique(AGE))
 [1] 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
[40] 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79 80 81 82 83 84 85 86 87 88 90
> length(unique(AGE))
[1] 73
```

Modem, tedy nejčastějším věkem je věk 36 let, v počtu 898 výskytů. Tato hodnota je jen malý kus pod průměrem. Celkově se pod průměrným věkem nachází 17508 pozorování tvořících zhruba 53,8 % celkového počtu. Funkce unique, zaobalená ve funkci sort pro lepší přehled vypsala dostupné věky v souboru a length provedl jejich součet, kde výsledek je 73. Níže lze vidět na histogramu, že má věk určitou klesající tendenci a je nakloněn více vlevo.

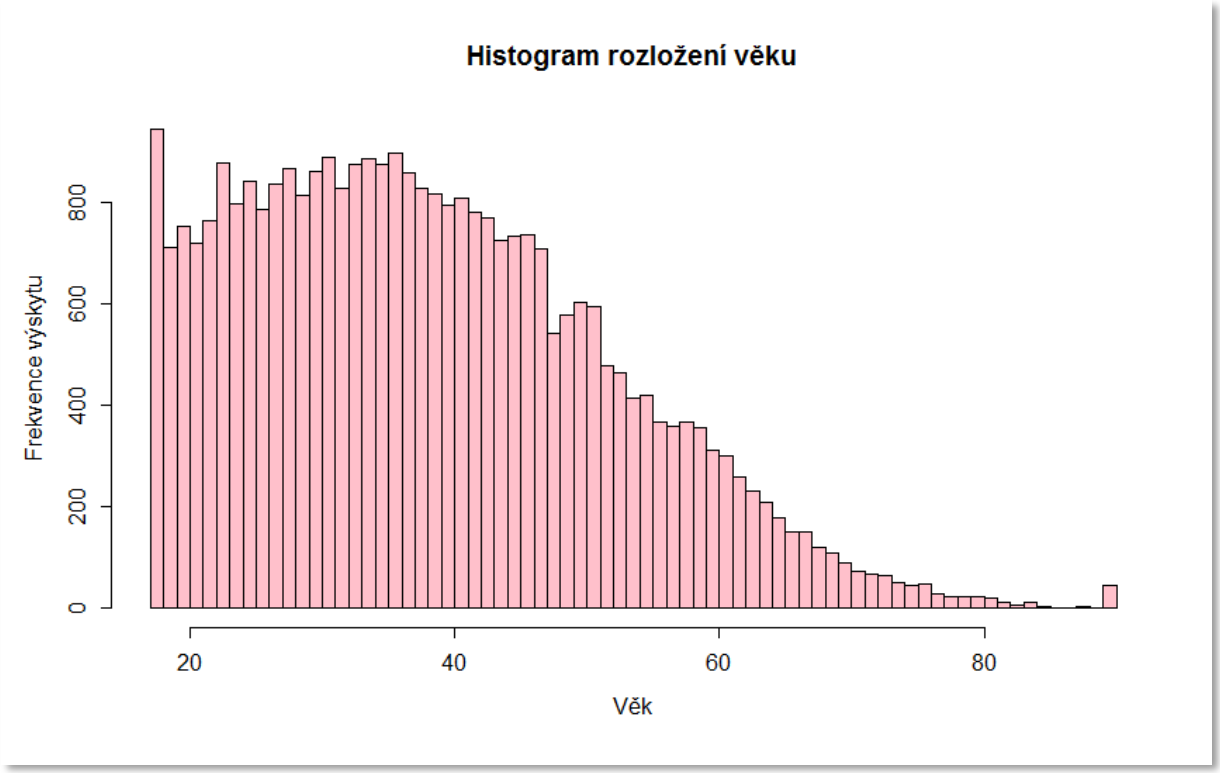


Diagram 9.2.1 - histogram rozložení věku, vygenerován pomocí funkce hist() v R

Navazuje atribut pracovní pozice, kde bude využita funkce summary.

```
> data.frame(summary(WRKCLASS))
                  summary.WRKCLASS.
?                              1836
Federal-gov                     960
Local-gov                      2093
Never-worked                      7
Private                       22696
Self-emp-inc                   1116
Self-emp-not-inc               2541
State-gov                      1298
Without-pay                      14
```

Pro přehlednější výpis byla funkce zaobalena do datové struktury data.frame. Z výstupu lze vidět že je zde 1836 neznámých hodnot které reprezentuje otazník. Jednoznačně zde dominuje privátní sektor 22696 výskyty s tím, že zde můžeme najít i jedince, kteří nikdy nepracovali, či pracovali bez nároku na finanční odměnu.

```
> zam <- summary(WRKCLASS)
> zam <- data.frame(zam)
> zam$zam <- 100*(zam$zam/length(WRKCLASS))
> colnames(zam)[1] <- "procenta"
> zam
                   procenta
?                5.63864746
Federal-gov      2.94831240
Local-gov        6.42793526
```

```
Never-worked      0.02149811
Private          69.70301895
Self-emp-inc      3.42741316
Self-emp-not-inc  7.80381438
State-gov         3.98636406
Without-pay       0.04299622
```

Do proměnné *zam* byl vložen obsah funkce summary a jeho datový typ změněn na data.frame. Záhy je možné ve sloupci hodnot sledovat převedená procenta (nativně R nepodporuje, pouze obsahem knihoven) ze kterých vyplyne, že privátní sektor pokrývá skoro 70 procent všech pozorování. Pro usnadnění práce u přehledu ostatních atributů bude vytvořena funkce, vylepšená o malé drobnosti.

```
procenta <- function(atribut)
{    pom <- summary(atribut)
     pom <- data.frame(pom)
     pom$pom <- paste(round(100*(pom$pom/length(atribut)),digits = 2),
"%")
     colnames(pom)[1] <- "procenta"
     return(pom)
}
```

Funkce paste zajistí spojení výpočtu procent a znaku %. Round způsobuje, aby se výsledky nezobrazovali v tak velkých desetinných místech, limituji je tímto na dvě desetinná místa. Nyní už je možné funkci použít pro výpočty dalších atributů.

```
> procenta(EDU) //Vzdělání
             procenta
10th           2.87 %
11th           3.61 %
12th           1.33 %
1st-4th        0.52 %
5th-6th        1.02 %
7th-8th        1.98 %
9th            1.58 %
Assoc-acdm     3.28 %
Assoc-voc      4.24 %
Bachelors     16.45 %
Doctorate      1.27 %
HS-grad       32.25 %
Masters        5.29 %
Preschool      0.16 %
Prof-school    1.77 %
Some-college  22.39 %
```

```
> procenta(OCCUP) //Povolání
                  procenta
?                   5.66 %
Adm-clerical       11.58 %
Armed-Forces        0.03 %
Craft-repair       12.59 %
Exec-managerial    12.49 %
Farming-fishing     3.05 %
Handlers-cleaners   4.21 %
Machine-op-inspct   6.15 %
Other-service      10.12 %
Priv-house-serv     0.46 %
Prof-specialty     12.71 %
Protective-serv     1.99 %
Sales              11.21 %
Tech-support        2.85 %
Transport-moving     4.9 %
```

```
> procenta(MARISTAT) //Rodinný stav
                      procenta
Divorced              13.65 %
Married-AF-spouse      0.07 %
Married-civ-spouse    45.99 %
Married-spouse-absent  1.28 %
Never-married         32.81 %
Separated              3.15 %
Widowed                3.05 %
```

```
> procenta(RELAT) //Druh rodi.
stavu
               procenta
Husband         40.52 %
Not-in-family   25.51 %
Other-relative   3.01 %
Own-child       15.56 %
Unmarried       10.58 %
Wife             4.82 %
```

```
> procenta(NATCOUNT) //Původní nár.
                            procenta
?                             1.79 %
Cambodia                      0.06 %
Canada                        0.37 %
Columbia                      0.18 %
Cuba                          0.29 %
Dominican-Republic            0.21 %
Ecuador                       0.09 %
El-Salvador                   0.33 %
England                       0.28 %
France                        0.09 %
Germany                       0.42 %
Greece                        0.09 %
Guatemala                      0.2 %
Haiti                         0.14 %
Holand-Netherlands               0 %
Honduras                      0.04 %
Hong                          0.06 %
Hungary                       0.04 %
China                         0.23 %
India                         0.31 %
Iran                          0.13 %
Ireland                       0.07 %
Italy                         0.22 %
Jamaica                       0.25 %
Japan                         0.19 %
Laos                          0.06 %
Mexico                        1.97 %
Nicaragua                      0.1 %
Outlying-US(Guam-USVI-etc)    0.04 %
Peru                           0.1 %
Philippines                   0.61 %
Poland                        0.18 %
Portugal                      0.11 %
Puerto-Rico                   0.35 %
Scotland                      0.04 %
South                         0.25 %
Taiwan                        0.16 %
Thailand                      0.06 %
Trinadad&Tobago               0.06 %
United-States                89.59 %
Vietnam                       0.21 %
Yugoslavia                    0.05 %
```

```
> procenta(SEX) //Pohlaví
        procenta
Female   33.08 %
Male     66.92 %
```

Tabulka 9.2.1

Mezí zajímavé údaje by se dal vybrat například i ten fakt, že celou třetinu maximálního dosaženého vzdělání pokrývá vzdělání středoškolské, dále také, že 90 % zkoumaných je původních obyvatel, a že téměř dvě třetiny tvoří muži, konkrétně 66 %. S další grafickou reprezentací dat napomůže funkce boxplot, kdy proti sobě je možné položit 2 smysluplné údaje a graficky je porovnat.

```
> boxplot(AGE ~ INCOME, main = "Rozložení věků na příjmy", xlab =
"Příjmy", ylab = "Věk", col = "red")
```

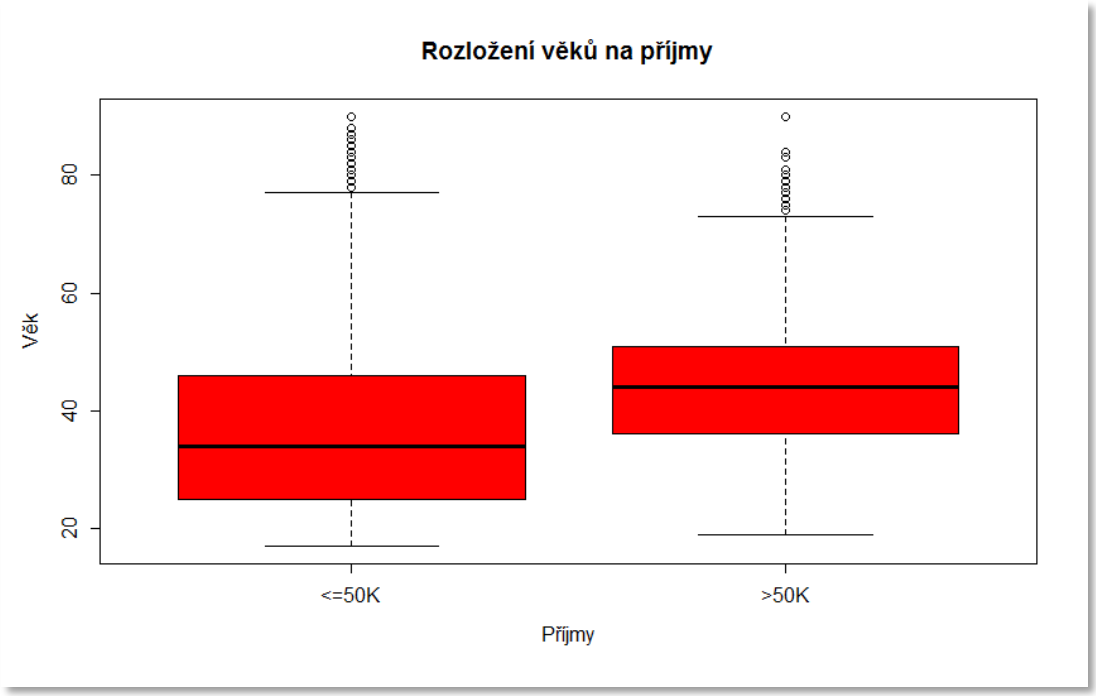


Diagram 9.2.2 - boxplot rozložení věků na příjmy, vygenerován pomocí funkce boxplot() v R

```
> boxplot(HPW ~ INCOME, main = "Rozložení hodin týdně strávených v
práci na příjmy", xlab = "Příjmy", ylab = "Hodiny v práci", col =
"brown")
```

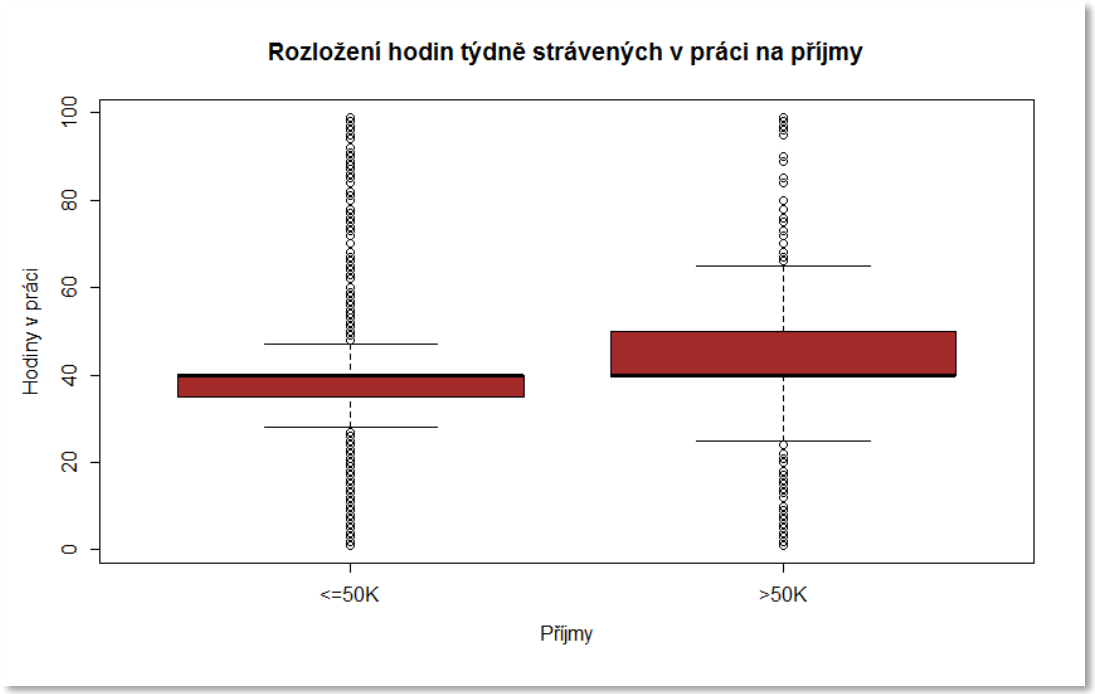


Diagram 9.2.3 - boxplot rozložení hodin týdně strávených v práci na příjmy, vygenerován pomocí funkce boxplot() v R

Boxplot je speciálním druhem grafu a další způsob vizualizace dat. Horní hranice prostředního obdélníku představuje třetí kvartil a spodní hranice první kvartil. Čarou v obdélníku je zastoupen medián souboru a úsečkou nad a pod ním je zobrazeno maximum a minimum. Jednotlivé body zobrazují odlehlé hodnoty souboru. Z druhého grafu je opět patrné, že výdělek nad 50 tisíc dolarů ročně nemusí nutně znamenat trávit více hodin v práci. Stejně tak první graf znázorňuje, že více mladších lidí dosahuje příjmů nad 50 tisíc.

Nyní budou porovnány základní funkce vizualizace R, konkrétně funkce plot a vizualizační knihovna ggplot2 (27).

```
> plot(INCOME, xlab = "PŘÍJEM", ylab = "počet")
```

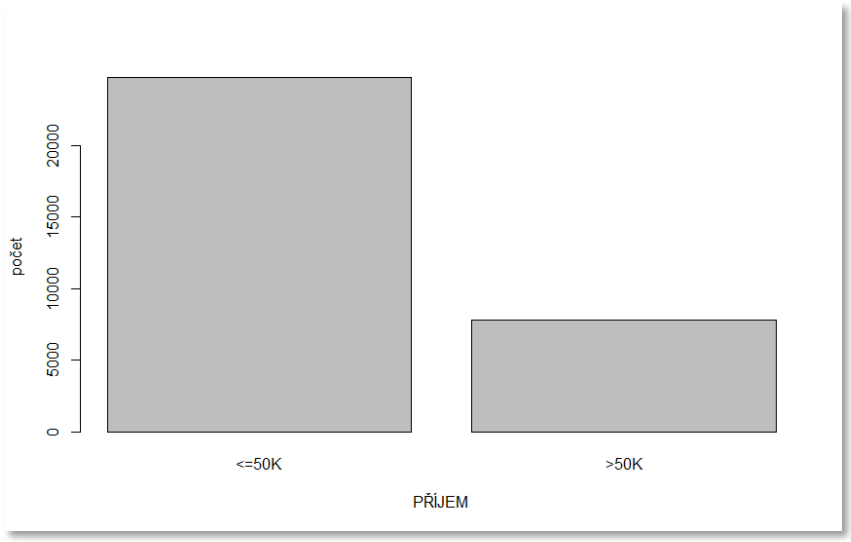


Diagram 9.2.4 - plot() v R

```
> qplot(INCOME, xlab = "PŘÍJEM", ylab = "počet")
```

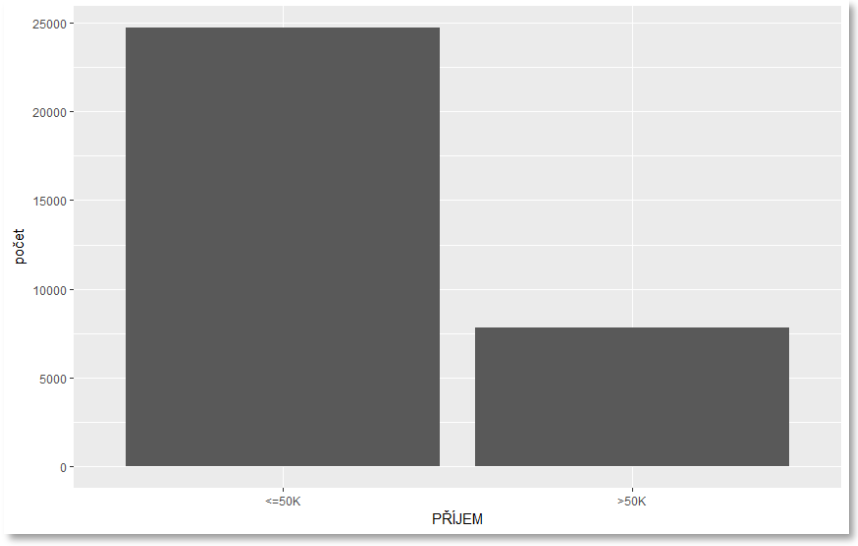


Diagram 9.2.5 - qplot() jako součástí package ggplot2 v R

Qplot zastává quick plot, tedy slouží k podobným účelům jako plot v základním R. Výhodou ggplot2 je však bezesporu jeho obratnost v mnohem barvitější a přehlednější vizualizace dat. Využiji jej proto nyní, k zobrazení příjmu ve spojení s pracovní pozicí.

```
> qplot(INCOME, fill = WRKCLASS) + facet_grid(. ~ adult$WRKCLASS)
```

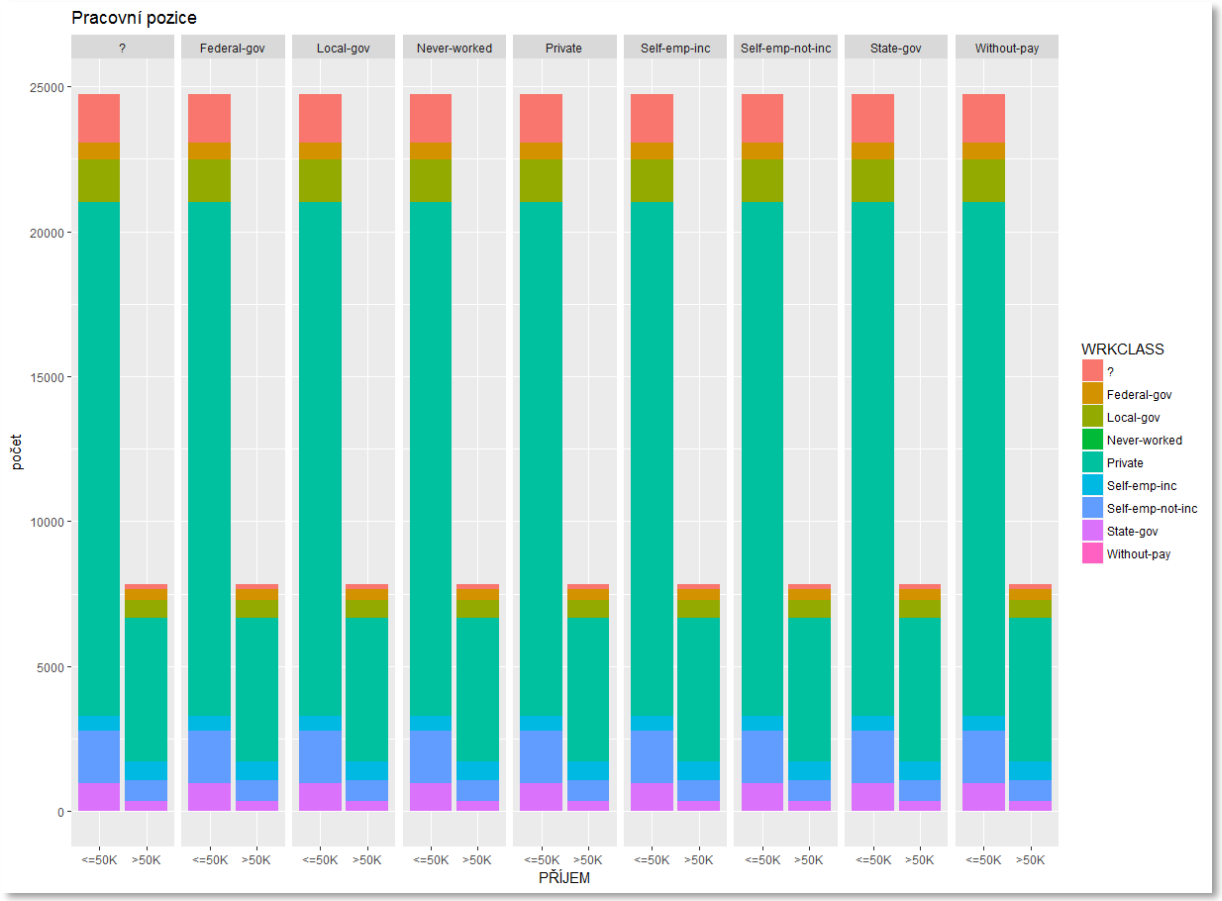


Diagram 9.2.6 - reprezentace pracovních pozic pomocí qplot() s funkcí face_grid() v R

Spojením příkazu qplot (s parametrem fill pro vyplnění atributu WRKCLASS) a příkazu facet_grid došlo k úhlednému rozdělení dat dle pracovní pozice a příjmu. Facet_grid slouží jako dělič dat pro diagram. Parametry příkazu zajišťují zobrazení ve vertikálním pořadí (kdyby byla tečka po parametru, WRKCLASS data by se zobrazily horizontálně). Legenda se u funkce qplot vypisuje automaticky. Co by ještě mohlo být zajímavě pro vizualizaci, je porovnání vzdělání a příjmů, diagram bude uveden v přílohách kvůli zachování kvality a čitelnosti dat.

10 Shrnutí a závěr

Na základě rešerše odborné literatury a dostupných online zdrojů zkoumajících cílovou oblast byla v této práci stručně popsána problematika Data Science. Odbornou literaturu tvořila zejména zahraniční literatura psaná v anglickém jazyce, což mi velmi vyhovovalo a považuji to také za velkou výhodu tématu Data Science. Téma této práce pro mě představovalo neprobádanou oblast informatiky, a proto jsem byl přesvědčen, že touto prací dokážu shrnout její podstatné základy.

V teoretické části byla okrajově dotčena historie a vznik vědy o datech, popsány její principy a její hlavní náplň datová analýza. Byly vysvětleny pojmy jako Big Data, struktura dat nebo metody uložení dat. Byl popsán programovací jazyk R a následně rozšířen o R-Studio což je integrované vývojové prostředí.

V praktické části byl kladen důraz na deskriptivní statistiku a její využití na reálných datech. Reálná data byla v obou případech získána z online zdrojů, kdy se v prvním případě jednalo o data z Českého statistického úřadu a v druhém případě o data z UCI datového repositáře. Z výsledků aplikace je možno získat přehled o rozložení obyvatel napříč okresy v České Republice nebo o tom, jaká byla v roce 1996 mezi obyvateli v USA situace v oblasti příjmů, zaměstnaní, rodinném stavu atp. Aplikační část pro mě znamenala vynaložení velkého úsilí, z důvodu zjištění nedostatečného přehledu v základních statistických pojmech. Takto mi umožnila si vyjasnit statistické definice a jejich využití.

Tato práce by měla sloužit hlavně jako základní přehled a shrnutí principů z oblasti Data Science. Využijí ho převážně Ti, co se chtějí s touto oblastí seznámit, nahlédnout na podprocesy této vědy nebo zjistit informace o programovacím jazyku R. Po přečtení této práce by měly být jasné pojmy jako rozměr dat, struktura dat nebo proces datové analýzy.

Potencionální rozšíření této práce vidím především v praktické části, kdy bylo pouze statistickou deskripcí nahlédnuto na dataset, který však byl primárně určen k účelům predikování, zdali příjem osoby přesáhne určenou částku. Teoretická část jistě snese rozšíření v detailnějším popisu datové analýzy a jejích následných forem (formy mohou být reprezentovány na ukázkových souborech) ale i v kapitole o samotné oblasti Data Science (rozšíření historie a podrobnější porovnání s ostatními vědami).

11 Seznam použité literatury

11.1 Tištěné zdroje

[1] ZAKI, Mohammed J. a Wagner MEIRA. Data mining and analysis: fundamental concepts and algorithms. ISBN 9780521766333.

[2] MAREŠ, Petr, Ladislav RABUŠIC a Petr SOUKUP. Analýza sociálněvědních dat (nejen) v SPSS. Brno: Masarykova univerzita, 2015. ISBN 978-80-210-6362-4.

[3] YU, Lei; LIU, Huan. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: ICML. 2003. p. 856-863.

[4] DIETRICH, David, Barry HELLER a Beibei YANG. Data science & big data analytics: discovering, analyzing, visualizing and presenting data. ISBN 978-1-118-87613-8.

[5] Data science: create teams that ask the right questions and deliver real value. ISBN 978-1-4842-2252-2.

[6] JANERT, Philipp K. Data analysis with open source tools. Beijing: O'Reilly, c2011. ISBN 978-0-596-80235-6.

[7] SCHUTT, Rachel a Cathy O'NEIL. Doing data science. ISBN 978-1-449-35865-5

[8] TUKEY, John W. Exploratory data analysis. Addison-Wesley Series in Behavioral Science: Quantitative Methods, Reading, Mass.: Addison-Wesley, 1977, 1977.

[9] MAZZA, Riccardo. Introduction to information visualization. London: Springer, c2009. ISBN 978-1-84800-218-0.

11.2 Elektronické zdroje

[10] PRESS, Gil. A Very Short History Of Data Science. Forbes [online]. Poslední změna 28.5.2013 09:09, [cit. 10.4.2017]. Dostupné z: https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/2/#5224cd03699e

[11] CASTROUNIS, Alex. What is Data Science, and What Does a Data Scientist Do? [online]. Poslední změna 7.3.2017, [cit. 31.3.2017]. Dostupné z: http://www.kdnuggets.com/2017/03/data-science-data-scientist-do.html

[12] DOLÁK, Ondřej. Big data, Nové způsoby zpracování a analýzy velkých objemů dat [online]. Poslední změna 2011, [cit. 2.2.2017]. Dostupné z: http://www.systemonline.cz/clanky/big-data.htm

[13] CLAVERIE-BERGE, Isabelle. Solutions Big Data IBM [online]. Poslední změna 13.3.2012, [cit. 2.2.2017]. Dostupné z: http://www-05.ibm.com/fr/events/netezzaDM_2012/Solutions_Big_Data.pdf

[14] KOMOROWSKI, Matthew. a history of storage cost (update) [online]. Poslední změna 9.3.2014, [cit. 2.2.2017]. Dostupné z: http://www.mkomo.com/cost-per-gigabyte-update

[15] STANTON, Jeffrey, Version 3: An introduction to Data Science [online]. Poslední změna 2012, [cit. 2.2.2017]. Dostupné z: http://surface.syr.edu/cgi/viewcontent.cgi?article=1165&context=istpub

[16] KALINA, Jan a TEBBENS, Jurjen Duintjer. Metody pro redukci dimenze v mnohorozměrné statistice a jejich výpočet [online]. s. 2. Poslední změna 16.8.2016 13:36, [cit. 2.2.2017]. Dostupné z: http://www.cs.cas.cz/duintjertebbens/pubs/KalinaDT.pdf

[17] NIST/SEMATECH e-Handbook of Statistical Methods, What is EDA? [online]. Poslední změna duben 2012, [cit. 5.3.2017]. Dostupné z: http://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm

[18]AMAR, Robert, EAGAN, James a STASKO, John. Low-level components of analytic activity in information visualization [online]. Poslední změna říjen 2005, [cit. 5.3.2017]. INFOVIS 2005. IEEE Symposium on. IEEE, 2005. s. 111-117. Dostupné z: http://www.cc.gatech.edu/~stasko/papers/infovis05.pdf

[19] TEAM, R. Core. R language definition. Vienna, Austria: R foundation for statistical computing, 2000 [online]. Poslední změna 17.4.2017, [cit. 11.3.2017]. Dostupné z: https://cran.r-project.org/doc/manuals/r-patched/R-lang.pdf

[20] "What is GNU?". The GNU Operating System. Free Software Foundation [online]. Poslední změna 4.9.2009, [cit. 11.3.2017]. Dostupné z: https://www.gnu.org/

[21] IHAKA, Ross. The R Project: A Brief History and Thoughts About the Future [online]. Poslední změna 21.10.2009 23:30, [cit. 11.3.2017]. Dostupné z: https://www.stat.auckland.ac.nz/~ihaka/downloads/Massey.pdf

[22] SMITH, David. A big list of the things R can do [online]. Poslední změna 2.7.2012, [cit. 11.3.2017]. Dostupné z: http://blog.revolutionanalytics.com/2012/07/a-big-list-of-the-things-r-can-do.html

[23] KRILL, Paul. Why R? The pros and cons of the R language [online]. Poslední změna 30.6.2015, [cit. 11.3.2017]. Dostupné z: http://www.infoworld.com/article/2940864/application-development/r-programming-language-statistical-data-analysis.html

[24] Friedrich Leisch. Creating R packages: A tutorial [online]. In Paula Brito, editor, Compstat 2008- Proceedings in Computational Statistics, Heidelberg, Germany, 2008. Physica Verlag. Poslední změna 14.9.2009, [cit. 11.3.2017]. Dostupné z: https://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf

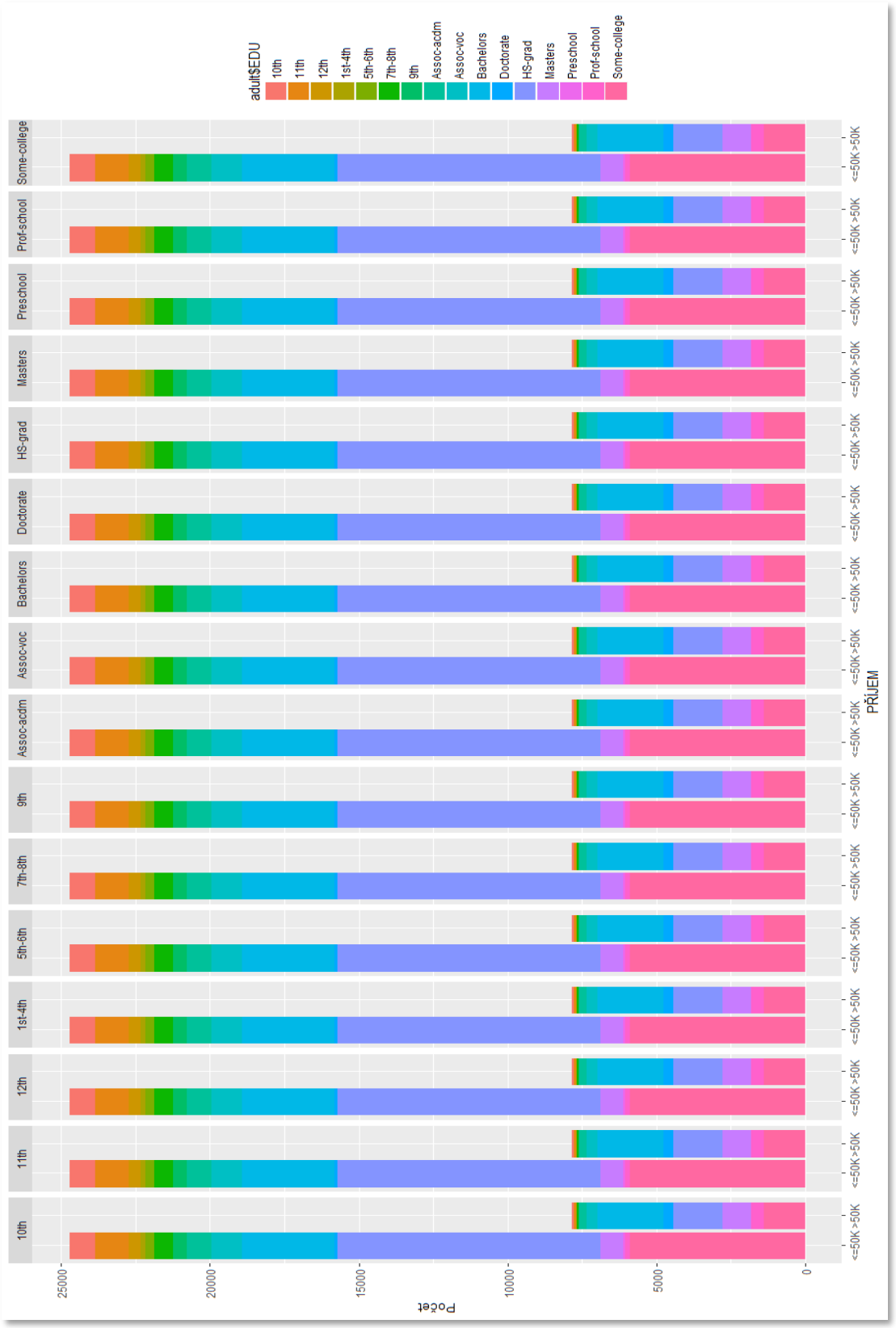
[25] Garret Grolemund. Quick list of useful R packages [online]. Poslední změna 13.1.2017, [cit. 11.3.2017]. Dostupné z: https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages

[26] Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Poslední změna 1.5.1996, [cit. 25.3.2017]. Dostupné z: https://archive.ics.uci.edu/ml/datasets/Adult

[27] H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, Poslední změna 2013, [cit. 25.3.2017]. Dostupné z: http://ggplot2.org/

12 PŘÍLOHY

1)


adult$EDU
10th
11th
12th
1st-4th
5th-6th
7th-8th
9th
Assoc-acdm
Assoc-voc
Bachelors
Doctorate
HS-grad
Masters
Preschool
Prof-school
Some-college
Počet
0
5000
10000
15000
20000
25000
PŘÍJEM
<=50K
>50K

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Brzek Tomáš	Pospíšilova 698/17, Hradec Králové	I14065

TÉMA ČESKY:

Data Science: Principy, technologie a znalosti.

TÉMA ANGLICKY:

Data Science. Principles, technologies and knowledge.

VEDOUCÍ PRÁCE:

prof. RNDr. Hana Skalská, CSc. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl: Sestavení přehledu a porovnání definic, metod a technologií pro tuto oblast. Pokusit se detekovat prospektivní nezbytné znalosti. Ideálně pohledem studenta AI a jeho současného obsahu vzdělávání.

1. Úvod a cíl práce
2. Přístupy k analýze dat
3. Popis jedno a vícerozměrných dat (redukce dimenze)
4. Data & Big Data
5. R-Studio (Základy, syntaxe, poznatky)
6. Aplikace
7. Shrnutí, výsledky a závěr

SEZNAM DOPORUČENÉ LITERATURY:

SEZNAM DOPORUČENÉ LITERATURY:

Stanton, J.M. An Introduction to Data Science.
ZAKI, Mohammed J. a Wagner MEIRA. Data mining and analysis: fundamental concepts and algorithms.

Podpis studenta: ..... **Datum:** .....

Podpis vedoucího práce: ..... **Datum:** .....