Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Automatizovaná analýza medicínských dat

Plzeň 2013 Pavel Cihlář

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem. Veškerá literatura a další zdroje, z nichž jsem při zpracování čerpal, jsou uvedeny v seznamu použité literatury a v práci řádně citovány. Práce nebyla využita k získání jiného nebo stejného titulu.

V Plzni dne 26. června 2013

Pavel Cihlář

Abstract

Diploma thesis focuses on the descriptive characteristic and analysis of the medical data stored in the MRE data model project. This model is organized through a series of ontologies in the RDF repository. The SPARQL language is used to obtain the data for the analysis. The theoretical part describes the technologies and statistical methods commonly used with the issue. The practical part describes the design and solution implementation for the descriptive characteristic of RDF data using standard UNIX tools. The outcomes are Shell scripts. Moreover, there is also described the design and implementation of the basic statistical and dependency analysis using the tools of R and MATLAB in the thesis. This part's outcome is the r–scripts and m–scripts generator written in Java language. In the conclusion, the thesis also includes the achieved results' final evaluation.

Key words: RDF, SPARQL, ontology, statistical methods, p–value, logistic regression, dataset, N–triples, R, MATLAB.

Abstrakt

Diplomová práce se zabývá popisnou charakteristikou a analýzou medicínských dat, která jsou uložena v datovém modelu projektu MRE. Ten je v RDF úložišti a je organizován pomocí řady ontologií. K získání dat pro analýzu je použit dotazovací jazyk SPARQL. V teoretické části jsou popsané použité technologie a statistické metody. V praktické části, práce je popsán návrh a implementace řešení pro popisnou charakteristiku RDF dat s použitím základních Unixových nástrojů. Výstupem popisné charakteristiky jsou Shell skripty. Dále je v práci popsán návrh a implementace základní statistické analýzy a analýzy závislostí s využitím nástrojů R a MATLAB. Výstupem této části je generátor r–skriptů a m–skriptů napsaný v jazyce Java. Závěr práce obsahuje zhodnocení dosažených výsledků.

Klíčová slova: RDF, SPARQL, ontologie, statistické metody, p–hodnota, logistická regrese, dataset, N–triples, R, MATLAB.

Poděkování

Děkuji Ing. Petru Včelákovi za odborné vedení a pomoc při řešení této diplomové práce.

Obsah

Úvod

V současnosti se většina medicínských výzkumů neobejde bez statistiky. Během těchto výzkumů je nutné nasbírat velké množství dat, která jsou následně podrobeny pomocí statických metod důkladnému prozkoumání. Výsledkem těchto analýz může být odpověď na otázku, jaké okolnosti ovlivňují výsledek léčby pacienta, nebo jaké charakteristiky, bývají pevně spjaty s danou diagnózou. Výsledkem analýzy můžeme získat preventivní nástroje k boji s konkrétní nemocí.

Tato diplomová práce se věnuje právě tématu statistické analýzy medicínských dat. Cílem je navrhnout a implementovat řešení pro popisnou charakteristiku, základní statistickou analýzu a analýzu závislostí jednotlivých prvků medicínských dat. Podmínkou tohoto řešení je možnost jeho automatizace a dále jeho nasazení v operačním systému založeném na Linuxu.

Medicínská data jsou uložena v datovém modelu projektu MRE, který je v RDF úložišti a je organizován pomocí řady ontologií. K získání těchto dat pro statistické zpracování je použit dotazovací jazyk SPARQL. V úvodu práce jsou proto tyto technologie popsány spolu se statistickými metodami, které jsou vhodné k řešení této problematiky.

Dále je v práci popsána implementace výsledných analyzačních nástrojů a jejich použití. V samotném závěru práce je provedeno zhodnocení dosažených výsledů a porovnání výstupů i implementace nástrojů R a MATLAB.

1 Pozadí práce

1.1 Resource Description Framework

Resource Description Framework (RDF) je standardním modelem pro prezentaci, popis a výměnu webových informací [1].

RDF rozšiřuje strukturu webových odkazů použitím jedinečných identifikátorů (URI) pro pojmenování vztahů mezi dvěma odkazy [2]. Tím získáváme tři URI, která tvoří nejjednodušší model RDF, označovaný jako trojice. Struktura tohoto modelu vytváří orientovaný ohodnocený graf, kde hrany představují pojmenované propojení mezi dvěma uzly. Tyto uzly představují vrcholy grafu.

První pracovní návrh RDF se dostal na veřejnost v roce 1997, kdy jej zveřejnila standardizační skupina World Wide Web Consortium (W3C) [3]. Tato technologie slibovala snadnější kategorizování a organizování webových informací.

1.1.1 Cíle RDF

`W3C` vyvíjelo RDF za účelem splnit následující cíle:

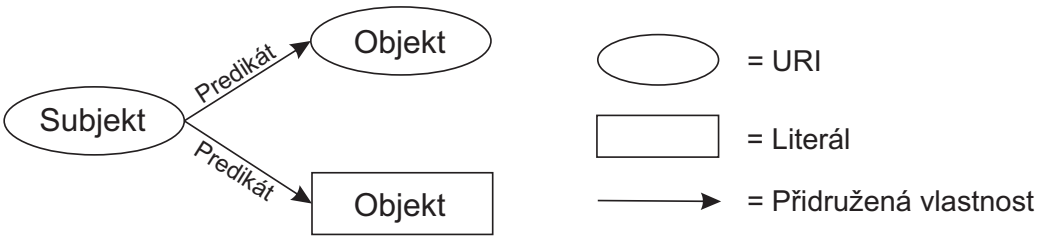
- mít jednoduchý datový model, který by aplikace snadno zpracovaly a s kterým by snadno manipulovaly,
- formální sémantiku, definující spolehlivá pravidla,
- syntaxi založenou na XML, která umožní zakódování datového modelu pro výměnu informací mezi jednotlivými aplikacemi,
- umožnit, aby kdokoliv mohl učinit tvrzení o jakémkoliv zdroji.

RDF umožňuje popis informací (např. titulek, autor, datum úpravy, obsah, atd.), tak aby mohlo být čteno a „pochopeno“ stroji a nezobrazovalo se uživateli.

1.1.2 Koncept RDF

Grafový datový model

Základní struktura každého výrazu RDF je soubor trojic (*angl. triples*), z nichž se každá skládá z předmětu (*angl. subject*), predikátu (*angl. predicate*, nebo *property*) a objektu (*angl. object*). Trojice je tvrzení (*angl. statement*) a tvoří RDF graf. Problematiku nám nejlépe osvětlí obrázek 1.1.



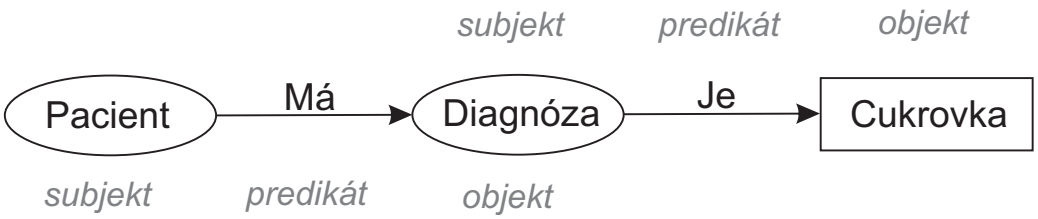
Obrázek 1.1: Ukázka obecné trojice

- Subjekt zde představuje daný datový zdroj (URI).
- Predikát představuje vlastnost subjektu a vyjadřuje tak vztah mezi subjektem a objektem.
- Objekt má nějakou hodnotu a často bývá odkazem na jiný subjekt (URI) – tím vznikají zřetězení. Objektem ale může být i tzv. literál – tj. např. jen řetězec znaků (`string`), číslo (`integer, double, decimal`), pravdivostní hodnota (`boolean - true, false`) atd.

RDF je datový model sémantického webu, který se dá nejlépe vyjádřit orientovaným grafem. Jednotlivé trojice tvoří jeden orientovaný graf, kde objekt jedné trojice může být subjetem druhé trojice viz obr. 1.2.

Datové typy

Datové typy jsou v RDF použity pro reprezentaci hodnot, jako jsou celá čísla, čísla s plovoucí čárkou a data [2]. Datový typ se skládá z lexikálního prostoru, z hodnotového prostoru a z mapovacích hodnot.



Obrázek 1.2: RDF graf

Například mapovací hodnota pro XML schéma datový typ je `xsd:boolean`, kde každý člen hodnotového prostoru (reprezentován T=true a F=false) má dvě lexikální reprezentace. To je patrné z tabulky 1.1.

Hodnotový prostor	{T, F}
Lexikální prostor	{"0", "1", "true", "false"}
Mapovací hodnoty	{<"true", T>, <"1", T>, <"0", F>, <"false", F>}

Tabulka 1.1: Příklad datového typu

RDF předdefinovává jen jeden datový typ a to `rdf:XMLLiteral`. Ten se používá pro vkládání XML v RDF. Není zde vestavěn žádný mechanismus pro číselné, datové a jiné společné hodnoty, ani pro jejich definování. RDF raději používá datové typy, které jsou definovány odděleně a identifikují se pomocí URI odkazů.

1.1.3 XML serializační syntaxe

Serializace je v oblasti informačních technologií mechanismus, který převádí určitý objekt do posloupnosti bitů [10]. Díky serializaci pak lze onu instanci bez problémů uložit do nějakého perzistentního úložiště (databáze, soubor, atd.), ale také i pomocí např. HTTP protokolu přenést na zcela jiný počítač. Všechny serializace RDF jsou vzájemně převoditelné. K tomu lze např. užít nástroje Any23, nebo Raptor RDF parser.

RDF serializační formáty jsou: N–Triples, N3 (nebo Notation3), Turtle, RDF/XML, RDFa (RDF in attributes)[11].

N–Triples

N–Triples je velice jednoduchá serializace, proto jí W3C Core Working Group používá k jednoznačnému vyjádření RDF datových modelů při vývoji aktualizací RDF specifikace [11]. Díky této jednoduchosti je také možné využít tento formát například při ručním psaní datasetů, určených k testování aplikací a ladění. Typická koncovka souboru pro formát N–Triples je `.nt`. V případě přenosu protokolem HTTP je použit mime typ `text/plain` [12].

Každý řádek ve formátu N–Triple představuje jednu trojici obsahující subjekt, predikát a objekt. Prvky trojice jsou od sebe odděleny mezerou a každá trojice je ukončena tečkou. Kromě prázdných uzlů a literálů jsou všechny prvky trojice URI odkazy uzavřené v lomených závorkách. Anonymní uzly jsou reprezentovány jako `_:name`, kde `name` je alfanumerický název uzlu, který začíná písmenem. Příklad formátu je na obrázku A.1.

N3

N–triples je koncepčně velmi jednoduchý formát, ale obsahuje větší množství redundantních informací. To má vliv na rychlost přenosu i zpracování. V případě malých dat se jedná o zanedbatelný problém. V případě práce s velkými objemy dat je vzhledem k menší náročnosti mnohdy výhodnější užít N3. N3 redukuje většinu opakujících se informací N–Triples formátu.

Počet prvků N-Triples formátu lze výrazně snížit tím, že použijeme krátké symboly, které budou reprezentovat opakující se uzly. Příklad formátu je na obrázku A.2.

RDF XML

Původní W3C doporučený formát na popis RDF byl XML datový model. Díky tomu se RDF často zaměňuje za RDF/XML serializaci [11].

Koncepčně je RDF/XML vytvořen z řady menších popisů, kde každý z těchto popisů sleduje cestu vedoucí přes RDF graf. Tyto cesty jsou popsány z hlediska uzlů (subjekty), hran (predikáty) a k nim připojených uzlů (objekty). Příklad formátu je na obrázku A.3.

1.2 Ontologie

Dle Borsta je ontologie formální specifikace sdílené konceptualizace určité oblasti [16]. Formální specifikace vyjadřuje znalosti pomocí určitého ontologického jazyka. Sdílená znamená, že ontologii užívají všichni členové určité komunity pro popis konceptů dané domény. Pojem konceptualizace znamená, že ontologie definuje koncepty domény na určité úrovni abstrakce, která odpovídá požadavkům na modelování domény.

Ontologie kromě definování vztahů mezi objekty umožňuje i objekty organizovat a umisťovat je do kategorií [17]. Mezi hlavní přínosy ontologií patří zabranění vícevýznamovosti. Podporují znovu využívání existujících struktur a dokáží sjednotit a propojit odlišnou terminologii.

Základní prvky ontologií jsou:

- Třída – základní stavební kámen ontologie. Tvoří uspořádanou stromovou hierarchii.
- Vlastnost – definuje vztah mezi jednotlivými třídami a individuály.
- Individuál – instance třídy.

1.2.1 Ontologie MRE

Ontologie MRE[1] (Medical Research & Education) vznikly na KIV. Vycházejí ze stejnojmenných následujících datových standardů:

- DASTA [20] – datový standard Ministerstva zdravotnictví ČR verze 3 i verze 4 (zkráceně DS3 nebo DS4 nebo obecně DASTA) slouží k předávání dat mezi zdravotnickými informačními systémy, je využíván v každodenní praxi a je zabudován do všech současných významných zdravotnických informačních systémů.
- DICOM [21] – datový standard společnosti NEMA pro distribuci, výměnu a zobrazování medicínských obrazových vyšetření nezávisle na jejich původu. Převzato z univerzity Stanford (DICOM Ontology Project).

[1] http://mre.kiv.zcu.cz

- SITS [22] – datový standard obsahující klinické a léčebné záznamy spojené s mrtvicí jako jsou terapeutické záznamy, popis příčin úmrtí a mnoho jiných.
- NIHSS [23] – je datový standard popisující hodnocení zdravotního stavu pacienta po mrtvici. Sleduje se, zda je pacient schopen mluvit, odpovídat na otázky, zda není narušena jeho orientace v prostředí nebo motorické schopnosti. Výsledkem je číselná hodnota udávající úroveň postižení. Čím více se hodnota blíží k nule, tím menší postižení u pacienta existuje.

Ukázka grafu ontologie DASTA, který vznikl v nástoji Protegé je v příloze B.1.

1.3 OWL

OWL (*angl. Web Ontology Language*) je ontologický jazyk [24]. Usnadňuje lepší strojovou interpretovatelnost webového obsahu, než je tomu u XML, RDF a RDFS. Poskytuje další slovní zásobu s formální sémantikou [25]. Vývoj jazyka probíhá pod hlavičkou W3C.

Volbu úrovně expresivity jazyka je nutné zvažovat s ohledem na výpočetní složitost, OWL poskytuje tři různé varianty jazyka. Jsou to OWL Lite, OWL DL a OWL Full.

1.3.1 Protégé

Protégé je volně šiřitelný ontologický editor a znalostně založený framework [26]. Umožňuje vytváření doménových modelů a znalostně založených aplikací s ontologiemi. Protégé implementuje bohatou sadu struktur znalostního modelování a činností pro podporu tvorby, vizualizace a manipulace s ontologiemi v různých reprezentačních formátech. Je napsán v jazyce Java.

2 SPARQL

SPARQL (Structured Protocol And RDF Query Language) je dotazovacím jazykem, který je určen k tvoření dotazů nad RDF grafy. Jeho syntaxe je do jisté míry podobná dotazovacímu jazyku SQL.

SPARQL obsahuje funkce pro dotazování povinných i nepovinných grafových vzorů a umožňuje jejich konjunkci i disjunkci. Dále umožňuje také určit, zda RDF graf obsahuje danou hodnotu. Výsledkem SPARQL dotazu může být i množina RDF grafů.

SPARQL vrací výsledky v mnoha různých formátech: např. XML, CSV/TSV ve formě jednoduché textové reprezentace (lehce použitelné tabulkovými editory), RDF, HTML, JSON [27].

SPARQL dotaz obsahuje [28]:

- Definici prefixů – pro zkrácení URI.
- Definici datasetu – definice dat, nad kterými bude SPARQL dotaz proveden.
- Definice klauzule (*angl. result clause*) – identifikuje druh dotazu (`select, construct, ask, describe`), resp. jaké informace budou dotazem vráceny.
- Vzor dotazu (*angl. query pattern*) – specifikuje podmínky, která musí vrácená data splňovat.
- Modifikátory dotazu – definice např. toho jak mají být data seřazena, nebo seskupena.

Obecný tvar dotazu je patrný z výpisu kódu 2.1:

```
# definice prefixu
PREFIX jmeno: <URI>
...
# definice datasetu
FROM ...
# definice klauzule
SELECT ...
# vzor dotazu
WHERE {
    ...
}
# modifikatory dotazu
ORDER BY ...
```

Výpis kódu 2.1: Obecný SPARQL dotaz

2.1 Namespaces

První částí SPARQL dotazu může být definice jmenných prostorů (*angl. namespaces*), které slouží k tomu, aby zkrátily URI jednotlivých prvků dotazu.

Subjekty, predikáty i objekty mohou být reprezentovány pomocí URI, proto mohou být zkráceny [28]. Jmenný prostor se definuje pomocí klíčového slova PREFIX, poté následuje název, dvojtečka a URI v lomených závorkách. Definice prefixu je zakončena tečkou. Příklad definice některých prefixů jsou ve výpisu kódu 2.2:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
PREFIX owl: <http://www.w3.org/2002/07/owl#> .
PREFIX mre: <http://mre.kiv.zcu.cz/> .
PREFIX sits: <http://mre...cz/ontology/2012/01/sits.owl#> .
PREFIX nihss: <http://mre...cz/ontology/2012/01/nihss.owl#> .
PREFIX dasta: <http://mre...cz/ontology/2012/01/dasta.owl#> .
PREFIX dicom: <http://bmir.stanford.edu/projects/dicom.owl#> .
```

Výpis kódu 2.2: SPARQL prefixy

2.2 Data

Vstupními daty může být jakýkoliv soubor dat (dataset) v libovolné RDF serializaci. Tento soubor může být lokální, nebo může být umístěn v úložišti na serveru.

SPARQL endpoint [30] umožňuje přístup k jednomu, nebo více datasetům. Jedná se v podstatě o webovou službu, která je identifikovaná pomocí URI a definuje komunikační protokol[1]. Do SPARQL endpointu mohou být posílány SPARQL dotazy, na které se poté vrací odpovědi.

2.3 Formy dotazů

SPARQL jazyk specifikuje čtyři různé varianty dotazů (*angl. result clause*) pro různé účely [31]:

- SELECT – nejběžnější forma dotazu. Používá se k extrahování hodnot. Výsledky jsou vráceny v tabulkovém formátu.
- CONSTRUCT vrací RDF graf vytvořený ze šablony, která je specifikovaná, jako součást samotného dotazu. CONSTRUCT lze použít i k transformování RDF dat například do jiné ontologie.
- ASK – vrací `true` a `false` výsledky v závislosti na tom, zda byl hledaný vzor v datasetu nalezen, nebo nikoliv.
- DESCRIBE – vrací RDF graf, který popisuje zdroj.

2.4 Vzor dotazu

SPARQL klauzule WHERE se poměrně liší od téže klauzule v SQL. V SQL se filtrují sloupce a řádky z tabulek, zatímco ve SPARQL se filtruje graf, v kterém je vše propojeno přes URI. Ve SPARQL je klauzule WHERE tvořena trojicemi.

[1]Seznam endpointů http://www.w3.org/TR/rdf-sparql-protocol/

Tyto trojice tvoří filtr, který je aplikován na RDF dataset a pouze vyhovující trojice (podgraf původního grafu) jsou vráceny jako výsledek. Jinými slovy tento filtr určuje podobu výsledného grafu, kde jsou všechna data také propojena.

Proměnné SPARQL dotazu jsou označeny prefixem „`?`“. Každý prvek trojice může být nahrazen `?` s libovolným jménem proměnné. Při zpracovávání SPARQL dotazu se vyhledávají soubory trojic, které odpovídají vzorům v klauzuli WHERE. Klauzule FROM specifikuje cílový graf dotazu.

2.5 Modifikátory a zkratky dotazu

SPARQL umožňuje použít řadu zkratek a modifikátorů. Zde je výčet těch nejdůležitějších z nich:

- SPARQL klíčové slovo „`a`“ je zkratkou pro častý predikát `rdf:type`, udávající třídu subjektu.
- Středník může být použit jako zkratka pro oddělení dvou trojic, které sdílí stejný subjekt.
- LIMIT je modifikátor, který určuje počet trojic vrácených dotazem.
- OFFSET se často používá ve spojení s modifikátorem LIMIT. Určuje, od jakého prvku se vrátí daný počet trojic.
- Modifikátor ORDER BY řadí výsledky dotazu podle hodnoty jedné, nebo více proměnných.
- Modifikátor GROUP BY seskupuje výsledky dotazů podle hodnoty jedné, nebo více proměnných.
- Modifikátor DISTINCT eliminuje duplicitní řádky vrácené dotazem.
- Omezení FILTER používá boolean podmínky k vyfiltrování nechtěných výsledků dotazu (např. `?age > 60`).
- OPTIONAL zkouší najít požadovaný vzor, ale neselže, pokud ho nenajde celý. Pokud najde částečné řešení, vrátí ho a nenalezené prvky vrátí jako nulové (`no value`).

- Klíčové slovo UNION slouží k oddělení vzorů dvou grafů. Výsledky obou stran příkazu UNION budou obsaženy ve výsledku.

2.6 SPARQL příklad

Ukázka SPARQL dotazu viz výpis kódu 2.3. Výsledek dotazu SELECT tvoří sloupce s ID pacienta (`patient_id`), datum zapsání diagnózy (`diag_date`), pořadí diagnózy u pacienta (`diag_order`), kód diagnózy (`diag_code`) a popis diagnózy (`diag_specification`). Data těchto sloupců podléhají vzorům v klauzuli WHERE, prvky trojice uvozené otazníkem jsou proměnné. Vzory uvozené klíčovým slovem OPTIONAL jsou ve výsledných datech buď doplněny příslušnou, nebo nulovou hodnotou. Nakonec je výsledek dotazu řazen abecedně podle sloupců uvedených v části ORDER BY.

```
SELECT
  ?patient_id
  ?diag_date ?diag_order ?diag_code ?diag_specification
WHERE {
  ?diagnosis_uri rdf:type ds:ActualDiagnosis .
  ?diagnosis_uri ds:hasPatient ?patient_uri .
  ?patient_uri ds:id_pac ?patient_id .

  OPTIONAL {?diagnosis_uri ds:dat_du ?diag_date .}
  OPTIONAL {?diagnosis_uri ds:poradi ?diag_order .}
  OPTIONAL {?diagnosis_uri ds:diag ?diag_code .}
  OPTIONAL {?diagnosis_uri ds:spec_dg ?diag_specification .}
}
ORDER BY ASC(xsd:int(?patient_id)) ASC(?diag_date) ASC(?
    diag_order)
```

Výpis kódu 2.3: SPARQL dotaz na aktuální diagnózu pacientů

3 Statistické metody

Vstupem statistické analýzy je CSV soubor, který vznikl SPARQL dotazem. Data v něm obsažená mohou obsahovat číselné i textové údaje. Tyto údaje mohou, ale nemusí být úplné, např. vyplnění data úmrtí u některých pacientů.

Vlastní statistická analýza těchto dat se skládá ze dvou bodů. Prvním bodem je analýza základní. Do základní statistické analýzy patří určení minima, maxima, průměrné hodnoty, mediánu, směrodatné odchylky a test na statistická rozdělení.

Druhým bodem je analýza závislostí jednotlivých prvků dat. Tato analýza často vychází z analýzy základní. Některé metody (parametrické) vyžadují předem známý typ statistického rozdělení. Úkolem analýzy závislostí je pokusit se prokázat na dané hladině významnosti, zda např. diagnóza X, je závislá na diagnóze Y.

V této kapitole je uveden výčet metod, které lze užít při statistickém zpracování získaných dat. K vytvoření základní popisné charakteristiky (počty trojic, nezávislých subjektů, predikátů, objektů apod.) statistické metody použity nebyly.

3.1 Testování hypotéz

Testování statistických hypotéz je důležitou součástí biostatistiky, zvláště pak v oblasti vyhodnocování experimentálních dat na poli biologického a medicínského výzkumu [34]. Na základě prokázání statistické hypotézy lze rozhodnout o platnosti určitého tvrzení na úrovni celé populace.

Statistická hypotéza je tedy určitý předpoklad o vlastnostech zkoumané veličiny [32]. Na základě realizace náhodného výběru (vstupní data) ověřujeme určitou hypotézu týkající se náhodné veličiny (např. diagnóza X) [33].

Statistické hypotézy mohou být dvojího druhu [32]:

- parametrické testy – určité tvrzení o parametrech (střední hodnota, medián, odchylka, apod.).

- neparametrické testy – jedná o tvrzení například o tvaru rozdělení.

Vzhledem k tomu, že rozdělení základního souboru (vstupních dat) ani jeho parametry většinou neznáme, nemůžeme se stoprocentní pravděpodobností určit, zda hypotéza platí. Postupujeme tedy tak, že prověříme náhodný výběr populace na námi zvolené hladině významnosti.

Testování statistických hypotéz je rozhodovací postup, u kterého se na základě charakteru vstupních dat rozhodneme buď pro testovanou (nulovou) hypotézu anebo pro alternativní hypotézu. Rozhodování probíhá na základě testovací statistiky (testového kritéria), což je vhodně vybraná funkce naměřených hodnot z výběru.

- Nulová, testovaná hypotéza, obvykle označovaná jako H_0, je tvrzení, že efekt zkoumané oblasti je nulový (Například, že diagnóza X není závislá na diagnóze Y).
- Alternativní hypotéza, obvykle označovaná jako H_1, nebo H_a, je tvrzení o opaku nulové hypotézy (nemusí však vždy jít o přesný logický opak).

V případě zamítnutí nulové hypotézy platí alternativní hypotéza. Pokud nulová hypotéza zamítnuta není, pak nám data neposkytla dostatek podkladů k podpoře alternativní hypotézy. Neznamená to tedy, že alternativní hypotéza neplatí. Při testování se můžeme dopustit chybných rozhodnutí. Tato chybná rozhodnutí se rozdělují na dva druhy chyb.

- Chyba I. druhu – spočívá v odmítnutí nulové hypotézy, ačkoliv je pravdivá. Pravděpodobnost výskytu této chyby nazýváme hladina významnosti (obvykle značíme α).
- Chyba II. druhu – spočívá v nezamítnutí nulové hypotézy, ačkoliv platí hypotéza alternativní. Pravděpodobnost výskytu této chyby značíme jako β. Hodnotu (1 - β) nazýváme síla testu. V praktických úlohách je chyba I. druhu závažnější, než chyba II. druhu.

Výsledkem testu je zamítnutí, nebo nezamítnutí nulové hypotézy na hladině významnosti α. Je ovšem nutné mít na paměti, že testovaný soubor dat musí být dostatečného rozsahu a musí obsahovat reprezentativní data.

3.1.1 Testování pomocí p–hodnoty

P–hodnota je častým výstupem počítačových programů na testování hypotéz. Udává mezní hladinu významnosti, při které bychom nulovou hypotézu ještě nezamítli. Nulovou hypotézu totiž zamítáme na hladině α právě v případě, kdy je p–hodnota menší než nula. V takovém případě je výsledek statisticky významný. Čím je p–hodnota menší, tím je pravděpodobnější, že výsledek je správný.

3.2 Neparametrické testy

Užití neparametrických testů je vhodné zejména v případech, kdy pracujeme s výběry poměrně malých rozsahů, nebo s výběry, které nepocházejí z normálního statistického rozdělení [47]. Dále také pro výběry, jejichž statistické rozdělení neznáme.

Výhody těchto testů spočívají nejen v nezávislosti na statistickém rozdělení, ale i v často jednodušším výpočtu [47].

3.2.1 Kolmogorovův–Smirnovův test

Autory této metody jsou Andrej Nikolajevič Kolmogorov a Vladimir Ivanovič Smirnov. Kolmogorovův–Smirnovův test (K–S test) je neparametrický test, který se často používá v případě, kdy potřebujeme rozhodnout o statistickém rozdělení prvků populace [35]. Tento test je založen na porovnávání distribuční funkce předpokládaného statistického rozdělení s výběrovou distribuční funkcí [36].

H_0: Náhodný výběr pochází z rozdělení se spojitou distr. funkcí $F_0(x)$.

H_A: Náhodný výběr nepochází z rozdělení se spojitou distr. funkcí $F_0(x)$.

Testovací statistika D je definována jako největší vzdálenost mezi hodnotami výběrové distribuční funkce $F_n(x)$ a teoretické distribuční funkce $F_0(x)$ rovnicí 3.1.

$$D = max \mid F_n(x) - F_0(x) \mid \tag{3.1}$$

Nulová hypotéza předpokládá, že testovaný výběr odpovídá vybranému teoretickému rozložení. Na rozdíl od metody χ^2 tato metoda pracuje přímo s jednotlivými naměřenými hodnotami náhodných veličin, nikoliv s četnostmi pro rozdělení do tříd.

Z toho důvodu je možné K–S test dobré shody použít i v případě malých náhodných výběrů.

Příklad volání příslušných funkcí:

```
R: ks.test(x, y)
Matlab: kstest(x,cdf,alpha,tail)
```

Výpis kódu 3.1: Metody statistického SW: KS–test

3.2.2 Test dobré shody

Test dobré shody (Pearsonův chí–kvadrát) se taktéž používá ke zjištění, zda sledovaná veličina má rozdělení pravděpodobnosti určitého typu [36]. V případě malých rozsahů vstupních dat tento test často ani nelze použít. V případě, kdy je tento test už na hranici použitelnosti, dává K–S test lepší výsledky. Teprve v případě dostatečně rozsáhlých vstupních dat je tento test výhodnější.

Základní myšlenka chí–kvadrát testu dobré shody spočívá v porovnání pozorovaných a očekávaných četností, to je patrné z rovnice 3.2. Nulovou hypotézou je typicky parametr rozložení.

H_0: Teoretické a empirické rozdělení se shoduje.

H_A: Teoretické a empirické rozdělení se neshoduje.

$$\chi^2 = \sum_{i=1}^{k} \frac{(x_i - E_i)^2}{E_i} \tag{3.2}$$

- x_i– pozorovaná frekvence.
- k– stupeň volnosti.
- E_i– očekávaná frekvence dle nulové hypotézy.

Příklad volání příslušných funkcí:

```
R: chisq.test(x, y, correct, p)
MATLAB: chi2gof(x)
```

Výpis kódu 3.2: Metody statistického SW: Test dobré shody

3.2.3 Test nezávislosti

Princip je do jisté míry podobný přechozímu testu dobré shody [36]. Tento test se používá k posouzení závislosti dvou kvalitativních veličin měřených na prvcích téhož výběru. Sledované statistické znaky se uspořádávají do kontingenční tabulky četností. Při chí-kvadrát testu nezávislosti tvrdí hypotéza H_0, že sledované znaky jsou nezávislé, alternativní hypotézou H_1 je pak hypotéza o jejich závislosti[1].

$$K = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - e_{ij})^2}{e_{ij}} \tag{3.3}$$

- r – značí počet řádků kontingenční tabulky.
- c – značí počet sloupců kontingenční tabulky.
- e_{ij}– jsou teorietické četnosti.

Test nezávislosti vede pouze k závěru, zda existuje závislost mezi sledovanými veličinami. Již neurčuje těsnost mezi těmito veličinami. To lze dále určit např. pomocí Cramerova, či Čuprova koeficientu kontingence.

Příklad volání příslušných funkcí:

```
R: chisq.test(table(znak1, znak2))
MATLAB: chi2test(x)
```

Výpis kódu 3.3: Metody statistického SW: Test nezávislosti

[1] viz STA 114: Statistics http://stat.duke.edu/courses/Fall11/sta114/chisq.pdf

3.2.4 Kruskal–Wallisův test

Kruskal–Wallisův test (K–W test) je neparametrickou obdobu níže popsané jednofaktorové analýzy rozptylu viz kapitola 3.3.4. Nejčastěji se tento test používá, když máme jednu sledovanou a jednu měřenou proměnnou [43]. Kruskal–Wallisův test porovnává střední hodnoty dvou a více vzorků, aby zjistil, zda vzorky pocházejí z různých populací.

Při použití toho testu se předpokládá, že nulová hypotéza je stanovena tak, že hodnoty v jednotlivých skupinách mají stejné mediány, dle vztahu 3.4 [44]. Testovací kritérium Q udává rozdílnost aritmetických průměrů ve skupinách.

$$H_0 : x_{0.5_1} = x_{0.5_2} = \cdots = x_{0.5_k} \tag{3.4}$$

$$Q = \left[\frac{12}{n \cdot (n-1)} \cdot \sum_{i=1}^{n} \left(\frac{(SR_i)^2}{n_i} \right) \right] - 3 \cdot (n+1) \tag{3.5}$$

Příklad volání příslušných funkcí:

```
R: kruskal.test(sloupec1 ~ sloupec2, data = jmeno)
Matlab: p = kruskalwallis(X,group)
```

Výpis kódu 3.4: Metody statistického SW: KW–test

3.2.5 Wilcoxonův párový test

Jedná se o neparametrickou obdobu níže popsaného T–testu, viz kapitola 3.3.3. Používá se při opakovaných měřeních stejných souborů, kdy sledovaná veličina neodpovídá normálnímu rozdělení [46]. Testuje nulovou hypotézu, která předpokládá rovnost distribučních funkcí.

Zatímco u párového T–testu se nulová hypotéza stanovuje na základě rozdílů aritmetických průměrů, zde se nulová hypotéza stanovuje na základě rozdílů středních hodnot (H_0 = medián rozdílů je nulový). Pokud je vypočtené testovací kritérium menší než tabulková hodnota, pak zamítáme nulovou hypotézu, v opačném případě jí zamítnout nemůžeme. Testovací kritérium je určeno menší hodnotou z výsledků rovnice 3.6 [45]:

$$H = W_+ + W_- = \frac{n \cdot (n+1)}{2} \tag{3.6}$$

- W_+ – označuje součet pořadí odpovídajících kladným rozdílům.
- W_- – označuje součet pořadí odpovídajících záporným rozdílům.
- n – počet prvků.

Příklad volání příslušných funkcí:

```
R: wilcox.test(sloupec1 ~ sloupec2, paired=TRUE)
Matlab: p = ranksum(x,y)
```

Výpis kódu 3.5: Metody statistického SW: Wilcoxonův test

3.3 Parametrické testy

Parametrické testy jsou založeny na jistých předpokladech [47]. Ve většině případů se předpokládá, že data, která chceme testovat, pochází z jistého předem známého statistického rozdělení. V případě velké části testů se předpokládá normální (Gaussovo) rozdělení. Když tato podmínka splněna není, je lepší využít neparametrických testů.

3.3.1 Dvouvýběrový test dobré shody rozptylů

Dvouvýběrový test dobré shody rozptylů (F–test) je zaměřen na testování shody rozptylů dvou nezávislých výběrů [35]. Tyto rozptyly mohou mít různý počet prvků (n_1 a n_2). Předpokladem je, že vstupní data musí mít normální rozdělení [37]. F–test je citlivý na porušení předpokladu normality. Nenaplnění této podmínky může značně zvýšit pravděpodobnost chyby I. druhu.

Test je založen na testování poměru rozptylů. V případě, kdy by se rozptyly nelišily, měl by být jejich poměr roven přibližně jedné. Nulovou hypotézu proto stanovujeme pomocí vztahu 3.7.

$$H_0 : \sigma_1^2 = \sigma_2^2 \tag{3.7}$$

Postup výpočtu je následující:

- Nejprve vypočteme výběrové rozptyly.
- Stanovíme počet stupňů volnosti u obou výběrů a vypočteme testovací kritérium (statistiku) F.
- Zvolíme hladinu významnosti a vyhledáme odpovídající kritickou hodnotu pro F–test.
- Vypočtenou statistiku F dle rovnice 3.8 porovnáme s tabulkovou kritickou hodnotou. Pokud je statistika F vypočtená větší než tabulková, zamítáme nulovou hypotézu. V opačném případě nulovou hypotézu zamítnout nemůžeme.

$$F = \frac{\textit{větší z rozptylů } (s_1^2, s_2^2)}{\textit{menší z rozptylů } (s_1^2, s_2^2)} \tag{3.8}$$

Příklad volání příslušných funkcí:

```
R: var.test(x, y, ratio=1,alternative, conf.level)
Matlab: vartest2(x,y,alpha,tail)
```

Výpis kódu 3.6: Metody statistického SW: Dvouvýběrový T–test

3.3.2 Jednovýběrový T–test

V případě, že vstupními daty je populace s normálním rozdělením a neznámou střední hodnotou μ a taktéž s neznámým rozptylem σ^2, lze použít jednovýběrový T–test k ověření předpokladu, že se střední hodnota μ rovná určité hodnotě μ_0. Nulová hypotéza je stanovena vztahem 3.9 [35]. Testové kritérium t je stanoveno dle rovnice 3.10. V případě, že data nemají normální rozdělení, je lepší využít mediánového testu, nebo Wilcoxonova testu.

$$H_0 : \mu = \mu_0 \tag{3.9}$$

$$t = \frac{\bar{X} - \mu}{S}\sqrt{n} \tag{3.10}$$

Příklad volání příslušných funkcí:

```
R: t.test(x, alternative, mu, var.equal, conf.level)
Matlab: ttest(x,alpha,tail)
```

Výpis kódu 3.7: Metody statistického SW: Jednovýběrový T–test

3.3.3 Párový T–test

Cílem tohoto testu je porovnat dvě populace, z nichž máme dva náhodné výběry, nebo porovnat dvě měření stejného výběru z jedné populace [35]. Porovnáváme střední hodnoty μ_1 a μ_2. Tento test lze využít např. v situaci, kdy měříme objekt vícekrát a chceme zjistit, zda nějaký faktor, měl vliv na měřený objekt [38]. Příkladem může být například měření krevního tlaku ráno, v poledne a večer.

Měřením dostaneme dvojice (x_1, y_1), (x_2,y_2), ... , (x_n, y_n). Označme je jako μ_1= E(X) a μ_2 = E(Y). Na hladině významnosti α pak můžeme testovat následující hypotézu dle rovnice 3.11.

$$H_0 : \mu_1 = \mu_2 \tag{3.11}$$

Postup výpočtu je následující:

- Nejprve spočítáme aritmetický průměr a rozptyl.
- Poté vypočteme testovací kritérium (statistiku) t dle rovnice 3.12.
- Porovnáme tabulkovou kritickou hodnotu t s hodnotou t vypočítanou pro příslušný stupeň volnosti.

$$t = \frac{\mid \bar{x} - \mu \mid}{\sqrt{\frac{s^2}{n}}} \tag{3.12}$$

Kde $\bar{x}$ je průměr výběrového souboru, μ je střední hodnota základního souboru, s^2 je rozptyl výběrového souboru a n je počet členů výběrového souboru.

Příklad volání příslušných funkcí:

```
R: t.test(x, y, alternative, mu, var.equal, conf.level)
Matlab: ttest(x,y,alpha,tail)
```

Výpis kódu 3.8: Metody statistického SW: Párový t–test

3.3.4 Jednofaktorová ANOVA

ANOVA, neboli analýza rozptylu (**AN**alysis **O**f **VA**riance) se zabývá posouzením vlivu jednoho a více faktorů na sledované proměnné [40]. Existuje řada metod analýzy rozptylu. Nejjednodušší je jednofaktorová, popř. jednostupňová ANOVA (*angl. one-way ANOVA*). Používá se tehdy, když testujeme jen jeden faktor, který nabývá více úrovní. Můžeme kupříkladu testovat, zda použitá medikace má vliv na krevní tlak pacientů.

Pokud chceme porovnat více než dva výběry, tak by zdánlivě stačilo utvořit všechny dvojice náhodných výběrů a provést nad nimi dvouvýběrový T–test. Těchto testů by ale bylo $\binom{k}{2} = \frac{k(k-1)}{2}$ [39]. Kdyby byl každý z nich proveden na hladině významnosti α, byla by výsledná hladina významnosti testu mnohem vyšší, než α. Tímto by byl test zcela znehodnocen. Proto v roce 1925 vytvořil sir R. A. Fisher metodu ANOVA.

Analýza rozptylu byla původně navržena pro stejný rozsah jednotlivých výběrů, což označujeme jako vyvážené třídění. Čím těsněji je toto pravidlo naplněno, tím přesnější jsou výsledky testu [39].

Analýza rozptylu ve své parametrické podobě podle předpokládá [39]:

- nezávislost výběrů,
- normalitu rozdělení,
- identické rozptyly.

Nesplnění nezávislosti výběru může vést k nepřesným, nebo i k zcela nesmyslným výsledkům. V případě, že všechny výběry mají vyšší rozsah, není

ANOVA na porušení normality příliš citlivá [39]. Celkový rozptyl sledované proměnné lze rozdělit na dvě složky:

- rozptyl uvnitř skupin – rozptyl mezi jednotlivými prvky skupiny kolem skupinového průměru,
- rozptyl mezi skupinami – rozptyl skupinových průměrů kolem celkového průměru všech skupin.

Testujeme nulovou hypotézu, která tvrdí, že střední hodnoty celé populace jsou shodné, oproti alternativní hypotéze, která říká, že alespoň jedna dvojice středních hodnot se liší [41]. Testovat lze např. pomocí F–testu [42] následujícím vztahem rovnice 3.13.

$$F = \frac{rozptyl\ mezi\ skupinami}{rozptyl\ uvnitř\ skupin} \tag{3.13}$$

Variabilita jednotlivých pozorování kolem celkového průměru SS_T je dána vztahem 3.14 a nazývá se celkový součet čtverců (*angl. total sum of squeres*). Celkový rozptyl MS_T (*angl. mean of squeres*) je dán vztahem 3.15 [39]. Tyto hodnoty vrací většina statistického SW.

$$SS_T = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X})^2 \tag{3.14}$$

$$MS_T = \frac{SS_T}{n-1} \tag{3.15}$$

Příklad volání příslušných funkcí:

```
R: A = aov(sloupec1~sloupec2,data)
Matlab:p = anova1(X,group)
```

Výpis kódu 3.9: Metody statistického SW: ANOVA

3.4 Korelační a regresní modely

Metody regresní a korelační analýzy slouží k popisu statistických závislostí na základě předem určených hypotéz. Vzhledem k tomu, že v praktických situacích často bývají jednotlivé prvky závislé a podmíněné více jevy, nevystačíme vždy pouze s metodami, které zjišťují závislosti jen u určitých, izolovaných prvků daného souboru dat.

3.4.1 Korelační modely

V případě korelačních modelů většinou vyšetřujeme vícerozměrný soubor, kde zjišťujeme míru postižení závislostí mezi jednotlivými prvky populace [48]. Závislost, či nezávislost daného prvku nám není předem známa. Model tohoto typu se nazývá korelační. Typickým příkladem může být např. vztah mezi věkem a pohlaví vzhledem k určité diagnóze.

Obecně lze korelační model maticově zapsat jako n x m rozměrné pole dat, kde m je počet sloupců matice X a n je počet hodnot každého sloupce matice 3.16.

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ \vdots & x_{i,j} & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix} \tag{3.16}$$

3.4.2 Regresní modely

Regresní modely se odlišují především tím, že nezávisle proměnná (vysvětlující proměnná) je předem stanovená [48]. Zadavatel (popř. uživatel) předem určí, které proměnné jsou závisle a které jsou nezávisle proměnné. Takové modely se nazývají regresní.

Jako příklad lze uvést věk pacienta v závislosti na jeho zdravotním stavu s věkem odstupňovaným po deseti letech, ve kterých měříme danou veličinu.

Regresní modely se dělí na dvě hlavní skupiny. Na lineární a nelineární regresi [48]:

- Lineární model je takový, který má parametry v lineárním postavení (např. přímkový model $y = a + b_x$).
- Nelineární model je takový, který nemá parametry ve vzájemném lineárním postavení (např. $y = ax^b$). Výpočet parametrů těchto modelů bývá obtížnější, než je tomu u lineárních modelů.

Není vždy možné korelační a regresní modely od sebe oddělovat, neboť možnost předem stanovit nezávisle proměnnou bývá v mnoha případech omezená, nebo nemožná. Vztahy používané pro řešení korelačních a regresních modelů jsou v podstatě shodné, liší se hlavně jejich význam a interpolace [48].

Poslední neméně důležitou věcí je, že prokázání statistické závislosti mezi danými proměnnými, nemusí nutně znamenat, že jsou příčinně závislé, neboť tato příčinnost může být způsobena jinými nezahrnutými faktory [48] . Poté vzniká tzv. zprostředkovaná korelace, jako je např. růst počtu televizí v závislosti na růstu chovanců psychiatrických léčeben apod. Proto je vždy nutné zvážit na základě znalostí studovaného problému, zda má příslušná korelace skutečné logické zdůvodnění.

3.4.3 Metoda lineární regrese

Jedná se o statistickou metodu, která umožňuje prozkoumat vztah mezi dvěma veličinami. Kde jedna z veličin, tzv. nezávisle proměnná X, má ovlivňovat druhou, tzv. závisle proměnnou Y [49]. Předpokládá se, že obě veličiny jsou spojité [50].

Drápela popisuje řešení takové úlohy v kapitole 10.5.1 (str.90) [48].

> „ *Při řešení se obvykle snažíme nahradit každou měřenou hodnotu závisle proměnné Y (vysvětlované), hodnotou teoretickou, která leží na spojité funkci (modelu) nezávisle proměnné X (vysvětlující).* “

Podstatou regresní analýzy je:

- Stanovit nejvhodnější tvar regresního modelu, tj. určit příslušnou rovnici, která bude popisovat závislost Y na X.

- Vypočítat parametry modelu.

Pokud jsou splněny podmínky lineárního regresního modelu, můžeme koeficienty regresní přímky odhadovat metodou nejmenších čtverců [39]. Obecný vztah je dán rovnicí 3.17.

$$\sum_{i=1}^{n}(y_i - y_i^{'})^2 = min \tag{3.17}$$

Smyslem metody je tedy nalézt takový tvar regresní funkce, který minimalizuje hodnotu součtu čtverců odchylek skutečných a modelem vypočtených hodnot závisle proměnné Y.

Příklad volání příslušných funkcí:

```
R: fit <- lm(y ~ x1 + x2 + x3, data=nazev)
Matlab: b = regress(y,X)
```

Výpis kódu 3.10: Metody statistického SW: Lineární regrese

3.4.4 Metoda logistické regrese

Logistická regrese je jednou z variant zobecněného lineárního modelu, kdy závislá proměnná má binomické (nebo multinomické rozdělení). Hlavní rozdíl mezi lineární a logistickou regresí je patrný z následující tabulky 3.1 [51].

Velký rozdíl, oproti ostatním metodám jsou vstupní data. Ta musí zahrnovat jeden, či více sloupců libovolných číselných hodnot (nezávisle proměnné) a jeden sloupec závisle proměnné s binárními daty (0;1), které představují výskyt daného prvku (např. textového řetězce). Základní model logistické regrese lze popsat pomocí rovnic 3.18 a 3.19 [51].

$$\ln\left[\frac{\pi}{1-\pi}\right] = \alpha + \beta x \tag{3.18}$$

$$\pi = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}} \tag{3.19}$$

Lineární regrese	*Logistická regrese*
Hodnoty Y nejsou omezeny.	Hodnoty Y (pravděpodobnosti) musí být v rozmezí 0-1.
Hodnoty Y mají normální rozdělení.	Hodnoty Y mají binomické (multinomické) rozdělení.
Hodnoty Y pro určitou hodnotu X mají stejnou střední hodnotu.	Pravděpodobnost nastoupení jevu je pro určitou hodnotu X konstantní.
Hodnoty Y pro všechny hodnoty X mají konstantní rozptyl.	Hodnoty Y pro všechny hodnoty X mají binomický rozptyl nπ(1-π), který závisí na X, pokud π závisí na X.

Tabulka 3.1: Porovnání lineární a logistické regrese

- Hodnoty α a β jsou regresní koeficienty, k jejichž nalezení je použita metoda nejmenších čtverců.
- X je náhodná proměnná binomického rozdělení s parametrem π.

Příklad volání příslušných funkcí:

```
R: glm(birth ~ death + sex, family=binomial("logit"))
Matlab: B = glmfit(X,Y,'distr')
```

Výpis kódu 3.11: Metody statistického SW: Logistická regrese

4 Analýza dat

Tato kapitola se věnuje praktickému využití výše popsaného. Vlastní praktická práce se skládá ze tří druhů analýz. Jednou z nich je analýza RDF datasetu, kde je hlavním úkolem vytvořit základní popisnou charakteristiku, jejímž výstupem má být např. počet unikátních tříd, subjektů, objektů a predikátů, celkový počet trojic obsažených v datasetu a tak podobně. Výsledný nástroj musí být funkční v operačním systému založeném na Linuxu. Dále má být spustitelný vícekrát a to bez přičinění uživatele (např. po spuštění operačního systému).

Další analýzou je základní statistický popis dat. Tedy již ne RDF datasetu samotného, ale dat která vznikla aplikací SPARQL dotazu nad tímto datasetem. Jedná se o vytvoření histogramů unikátních prvků a v případě číselného vstupu i o zjištění hodnot: minima, maxima, mediánu, průměrné hodnoty a směrodatné odchylky. Nad každým číselným sloupcem má být proveden test na povahu statistického rozdělení.

Poslední analýzou, která je taktéž prováděna nad fragmentem dat, je analýza závislostí. Úkolem této analýzy je především určit, zda jsou zadané sloupce na sobě závislé. Výsledný nástroj má produkovat skripty, které vytvoří interaktivně uživatel (tj. určí metody, předpoklady). Tyto skripty mají být, stejně jako v předchozím případě, funkční v operačním systému založeném na Linuxu a mají být spustitelné bez nutné interakce s uživatelem.

4.1 RDF Dataset

Vstupní RDF dataset, který má výsledný nástroj popsat, je serializován ve formátu N-Triples. K dispozici je ve třech vyhotoveních ve verzi minimal.nt, view.nt a full.nt. Všechny verze tohoto souboru obsahují prvky z ontologií DASTA, DICOM, SITS a NIHSS. Rozdíl jednotlivých verzí tkví hlavně v jejich velikosti.

Zatímco nejmenší verze minimal.nt má v komprimované formě 1,17 MB a v dekomprimované 36,9 MB s celkovým počtem 215 525 trojic, verze view.nt má v komprimované formě 426 MB a v dekomprimované 5,3 GB s celkovým

počtem 33 144 765 trojic. Dataset full.nt má v komprimované formě 3,39 GB a v dekomprimované 38 GB s celkovým počtem 221 498 874 trojic.

Z toho je již patrné, že pokud se bude soubor procházet během analýzy vícekrát, může se jednat o časově velice náročnou operaci. Na zrychlení takové operace by pak už nemusela stačit pouhá optimalizace kódu, ale mohlo by být nutné využít paralelního zpracovávání dat.

4.2 Základní popisná charakteristika RDF datasetu

Skutečnosti, které mají být popsané, jsou kromě již zmíněných komprimovaná i dekomprimovaná velikost vstupního RDF datasetu, datum analýzy, histogramy unikátních subjektů, objektů i predikátů a různé další četnosti specifických trojic.

Ačkoliv se tato analýza nemá provádět příliš často (jistě ne denně), nesmí to být operace, trvající v řádech dnů. Možností řešení se nabízí hned několik. Zjevnou možností je využít jazyku C, popř. Java. Použít knihoven Jena, nebo Sesame a jejich prostřednictvím následně jazyka SPARQL.

Vzhledem k tomu, že výsledný program má běžet v operačním systému založeném na Linuxu a k tomu, že u něj není zapotřebí grafického uživatelského prostředí, napadla mě další volba. Tou je linuxový Shell. Jeho výhody jsou v tomto případě vcelku zjevné. Jednoduchá rozšiřitelnost a editace prakticky bez jakéhokoliv vývojového nástroje. Jednoduchá možnost paralelizace bez nutnosti psaní vláken. Využití komprimátorů operačního systému atd. Využít jazyka SPARQL se zde nabízelo také a to pomocí nástroje MetaMed [53] [54]. Z těchto důvodů byl v tomto případě využit právě tento přístup.

4.2.1 Datový soubor N–Triples

Vzhledem ke skutečnosti, že k RDF datasetu lze přistupovat identicky jako k textovému souboru, mou první volbou je skript využívající textový přístup. To znamená v tomto případě to, že program prochází RDF dataset a hledá výskyt zadaného textového řetězce. V případě výskytu řetězce navýší hodnotu pomocné proměnné o jedničku a pokračuje dál ve své činnosti. V

implementaci jsou použity výhradně základní unixové nástroje (cut, uniq, sort, grep, wc apod.).

Pro vybrání právě těch dat, která jsou zrovna předmětem analýzy běžící úlohy, je nutné dataset předzpracovat. To například znamená, že v případě hledání unikátních predikátů (druhý prvek trojice) je nutné nejprve odstranit všechny subjekty a objekty a až poté počítat unikátní objekty.

Ačkoliv zde byla snaha o co nejmenší počet průchodů programu RDF daty, bylo po optimalizaci programu, vzhledem k časové náročnosti, nutné využít paralelního zpracování.

Jednotlivé na sobě nezávislé úlohy jsou rozděleny do funkcí, které jsou následně spuštěny paralelně. Poté program čeká na jejich dokončení. Po skončení všech fukcí jsou zapsány výsledky do příslušného textového souboru. Výsledky jsou také uloženy pomocí ontologie VoID [55]. Dalšími výstupy programu jsou histogramy subjektů, predikátů a objektů v samostatných textových souborech seřazených podle počtu výskytu daného prvku v pořadí od největšího počtu k nejmenšímu. Vývojový diagram je v příloze C.1.

Nejzávažnější chybou tohoto přístupu je však závislost na serializaci vstupního souboru. Program by musel být napsán zvlášť pro každý serializační formát, tj. musel by být připraven na každou eventualitu a na základě vstupního souboru spustit tu, či onu výkonnou část svého kódu.

V tuto chvíli skript funguje pouze pro serializaci N-Triples. Vzhledem k tomu, že všechny RDF serializace jsou vzájemně převoditelné, lze tuto nepříjemnost obejít převedením jiné vstupní serializace RDF dat na formát N-Triples.

Základní popisná charakteristika vytvořená tímto způsobem je v tabulce 4.1. Obsahuje datum analýzy, jméno vstupního RDF datasetu, dekomprimovanou a komrimovanou velikost datasetu v B a v MB, serializační formát, počet prvků z ontologií MRE, počet nezávislých subjektů, objektů a predikátů, celkový počet trojic, počet tříd a celkový počet entit.

4.2.2 RDF úložiště

Předchozí přístup získával vcelku bezproblémovým způsobem popisné charakteristiky, jako jsou velikosti souborů, serializace vstupního souboru, jméno

Document	Medical RDF Dataset
Date	27.01.2013
Code	Minimal.nt
Serialization format	N-triple
Uncompressed byte size	37984
Uncompressed size	38MB
Compressed byte size	2176
Compressed size	2,2MB
DASTA	15913
DICOM	160030
SITS	38461
All triples	215525
Distinct subjects	7660
Distinct objects	8023
Distinct classes	11
Distinct predicates	287
Total entities	2987

Tabulka 4.1: Příklad základní popisné charakteristiky

souboru a datum testování. Proto nebylo třeba tuto část skriptu jakkoliv měnit.

Kamenem úrazu předchozího přístupu byla hlavně jeho závislost na serializaci vstupních dat. Tuto skutečnost lze vyřešit pomocí jazyku SPARQL, který na RDF serializaci závislý není. Jazyk SPARQL byl použit prostřednictvím externí open–source aplikace MetaMed. Vývojový diagram skriptu je na obrázku D.1.

V průběhu analýzy skript po zjištění charakteristik přímo nesouvisejících s RDF daty, spustí program MetaMed, kterému předá parametry. Tyto parametry jsou název modelu, typ modelu (pouze v paměti, v úložišti apod.), cesta ke vstupnímu souboru, serializace souboru, adresář se SPARQL dotazy a výstupní adresář.

Po ukončení MetaMedu, na které skript čeká, jsou extrahovány data z MetaMedem nově vytvořených souborů. Tato data skript následně uloží stejně jako v případě předcházejícího skriptu do textových souborů a do souboru s použitím ontologie VoID.

Při odstranění SPARQL dotazů, které jsou vytvořené v adresáři „query“, skript nebude fungovat správně. Nicméně pokud se do tohoto adresáře přidají další soubory s dotazy, MetaMed automaticky vygeneruje příslušné soubory s požadovanými daty. Tento skript lze tedy velice snadno rozšiřovat, rozhodně snadněji, než jeho předchozí verzi. Základní popisná charakteristika vytvořená tímto přístupem je v tabulce v příloze D.1. Ukázka popisné charakteristiky uložené prostřednictvím ontologie VoID je na výpisu kódu v příloze D.1.

Vzhledem k tomu, že MetaMed umí bez problémů pracovat i se vzdálenými úložišti, bylo použití tohoto skriptu testováno i na RDF modelu v úložišti Oracle. MetaMed je nástroj pro výpis metadat a souborů v lékařských datových formátech. Podporuje proto medicínské formáty jako je DASTA, DICOM, SITS i HL7. Dále pak serializace RDF/XML, RDF/XML-ABBREV, TURTLE, N3 a N-TRIPLE. Po drobné úpravě skriptu (přepsání jednoho argumentu) by navíc bylo možné změnit výstupní formát dat na textový, csv, tsv, rdf, sse, json, xml anebo na xmlstring. Vzhledem k těmto výhodám se přístup s jeho využitím zdál nejvhodnější.

Nástroj Metamed je napsán v jazyce Java a tak je samozřejmým požadavkem nainstalovaná Java, Standard Edition (Java SE) JRE 1.6 a vyšší.

Fragment textového souboru s histogramem počtu objektů v datasetu minimal.nt je demonstrován tabulkou 4.2.

No	objects
435	http://mre.kiv.zcu.cz/ontology/2012/01/sits.owl#SITSReport
424	http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#ActualDiagnosis
324	http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#DASTA

Tabulka 4.2: Fragment histogramu popisné charakteristiky

Pro velký počet položekje zde uvedeno pouze prvních pár řádků souboru. Výsledné histogramy lze vizualizovat pomocí skriptu napsaného v jazyce MATLAB na výpisu kódu 4.1. Ten vrací grafy ve formátu `png` a `eps`.

```
close all;
clear all;
[numbers, names] = textread('h.txt', '%u %s');
bar(numbers,'grouped');
xlabel('Items');
ylabel('Count');
print('-dpng','histogram.png);
print('-depsc','histogram.eps);
```

Výpis kódu 4.1: Skript k visualizaci histogramu v programu Matlab

Na výpisu kódu 4.2 jsou některé SPARQL dotazy, které byly použity k analýze. Každý z těchto souborů je v samostatném textovém souboru v adresáři „query“. Metamed postupně zpracovává jeden dotaz po druhém a ukládá výsledky do csv souborů do adresáře „output“.

```
// Celkovy pocet trojic
SELECT (COUNT(*) AS ?no) { ?s ?p ?o  }
// Celkovy pocet nezavislych objektu
SELECT (COUNT(DISTINCT ?o ) AS ?no)
{  ?s ?p ?o  filter(!isLiteral(?o)) }
// Histogram objektu
SELECT (COUNT(?o) as ?no) ?o
WHERE { ?s ?p ?o filter(!isLiteral(?o)) }
GROUP BY ?o ORDER BY DESC(?no)
```

Výpis kódu 4.2: Příklady užitých SPARQL dotazů

4.3 Základní statistická analýza

Vstupem základní statistické analýzy je fragment dat RDF datasetu, který v tomto případě představuje CSV soubor. Výsledkem této analýzy má být soubor s příslušnými statistikami. Řešení má být neinteraktivní a znovu spustitelné v operačním systému založeném na Linuxu.

Vstupní CSV soubor nemá předem známou velikost, není znám počet jeho sloupců, ani typ dat, která tyto sloupce obsahují. Vzhledem k těmto skutečnostem bylo obtížné vybrat vhodnou technologii pro vývoj aplikace.

S přihlédnutím k požadavku na automatizaci lze využít skriptu. Komplikací ale je neznámý vstupní soubor resp. neznámý typ jeho dat. Také by

se dalo využít nějakého nižšího jazyku jako je např. C, nebo vyššího jako je např. Java. Ačkoliv získání základních statistických údajů by bylo po implementační stránce v těchto jazycích vcelku triviální, bylo pravděpodobné, že v případě analýzy závislostí mezi prvky souboru to již platit nebude. Nabízelo se sice využít nějakou z externích volně šiřitelných knihoven jako je např. Liblinear (Java), ale ani to se nezdálo jako elegantní řešení.

Z těchto důvodů byly použity nástroje, které jsou ke statistickým analýzám dat přímo určeny. Těmito nástroji byly MATLAB a R. Bylo zvažováno i využití MS Excelu, popř. Statistiky, ale pro vlastní řešení se tyto nástroje nezdály tak výhodné. Oba tyto nástroje byly zavrhnuty zejména z toho důvodu, že nejsou multiplatformní a nehodí se pro konstrukci neinteraktivních nástrojů. R i MATLAB navíc umí pracovat s více daty než MS Excel. Ten je omezený 256 sloupci a 65 tisíci řádky. To by vzhledem k objemu medicínských dat pro některé výpočty nemuselo stačit.

V případě nevyužití statistického softwaru by bylo zapotřebí všechny výše popsané statistické metody z kapitoly 3 implementovat a vyladit ručně. S použitím programů MATALAB a R v případě některých výpočtů v podstatě stačilo správným způsobem zavolat příslušnou funkci a předat jí správná data. Výstup v podobě m–skriptů a r–skriptů navíc splňoval požadavek automatizaci.

Nicméně i přes všechny zmíněné výhody bylo zapotřebí vytvořit prostředí, jehož prostřednictvím by uživatel tyto skripty tvořil. V případě analýzy závislostí totiž nelze udělat to, aby výsledný nástroj sám spouštěl všechny metody a kombinace jejich vstupních dat. Výsledných skriptů by bylo příliš mnoho. U vytváření skriptů je proto počítáno s uživatelem, který má základní vědomosti o statistice a o datech, která chce zkoumat.

Dalším důvodem pro vytvoření mezivrstvy byl částečně problematický vstup programu MATLAB. Z těchto důvodů byl volen jazyk pro předzpracování dat. Tím je v tomto případě myšlena především jejich identifikace. Určení v jakém sloupci jsou textová a v jakém číselná data.

Vzhledem k tomu, že nebylo vyloučeno, že program bude v budoucnu předělán (popř. nad ním bude vytvořena nadstavba) do webové formy, byl raději zvolen jazyk Java.

4.3.1 Uživatelské prostředí

Jako uživatelské prostředí byla zvolena konzolová aplikace. V případě základní statistické analýzy je úkolem uživatelského prostředí získat od uživatele jméno vstupního souboru a jeho umístění. Zadaná cesta může být absolutní i relativní a je jediným vstupním argumentem.

Samotný program projde data vstupního souboru a určí, která data jsou ve formátu textového řetězce a která jsou ve formátu desetinného čísla. Program byl připraven detekovat i celá čísla a časové údaje, ale v případě této analýzy se to ukázalo jako nepotřebné. Aplikaci blíže popíši v kapitole zabývající se analýzou závislostí.

Výstupem programu jsou dva skripty (R a MATLAB) určené k základní statistické analýze, které se vytvoří v adresáři „scripts“.

Na výpisu kódu 4.3 je fragment CSV souboru, z kterého byly následně vygenerovány následující skripty.

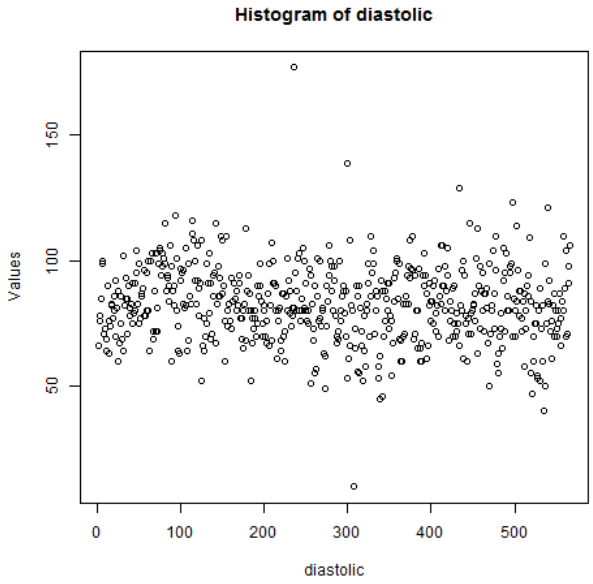
```
sits_id,timepoint,diastolic,systolic
CZUHP2005122701,After24H,66,117
CZUHP2005122701,After2H,76,156
CZUHP2005122701,Baseline,78,170
CZUHP2005122901,After24H,85,148
```

Výpis kódu 4.3: Fragment CSV souboru

4.3.2 Skript v R

Skript po načtení dat pomocí funkce `read.table` nejprve vytvoří pomocí funkce `summary` základní statistickou analýzu jednotlivých sloupců dat a pak i celého souboru. Následně se pro každý sloupec dat vytvoří grafický histogram ve formátu `png` a `eps`. Tento histogram nemusí vždy dávat smysl. Např. pokud je prvním sloupcem ID, je jasné, že provádět jakoukoliv statistiku, popř. vytvářet histogramy nad tímto sloupcem smysl nemá. Na obrázku 4.1 jsou naměřené hodnoty diastolického krevního tlaku.

Pro číselné sloupce se provede i analýza jejich statistického rozdělení. Jako nejlepší metoda se mi jevil K–S test popsaný v kapitole 3.2.1, který funguje téměř vždy i v případě malých rozsahů dat. Čím nižší p–hodnota, tím je



Obrázek 4.1: Diastolický krevní tlak – histogram

pravděpodobnější, že daný sloupec patří do příslušného rozdělení. Test se provádí na hladině významnosti α=0,05.

K–S test je prováděn pro normální (Gaussovo), binomické, poissonovo a gama rozdělení. Výsledky těchto testů jsou uloženy do jediného souboru.

Z výpisu kódu 4.4 je patrné, že výběr pravděpodobně pochází z poissonova rozdělení pravděpodobnosti. Dle p–hodnoty, která je v tomto případě velice malá, můžeme na hladině α=0,05 zamítnout nulovou hypotézu. Každý K–S test vrátí statistiku D, p–hodnotu, alternativní hypotézu, název testu a pořadí sloupce nad kterým byl tento test vykonán.

```
POISSON DISTRIBUTION:
Statistic: D = 0.154721535449846
P-value: 3.40583117264259e-12
Alternative hypothesis: two-sided
Method: One-sample Kolmogorov-Smirnov test
Data: data[, 4]
```

Výpis kódu 4.4: Část výstupu K–S testu v R

Na výpisu kódu 4.5 je výstup funkce `summary`, která je v tomto případě aplikována na celý vstupní soubor. Pro textové sloupce se automaticky vygenerují příslušné histogramy. Histogram `sits_id` je v tomto případě pro velký počet unikátních položek zkrácen. Je to dáno tím, že funkce `summary` bez dalších upřesňujících argumentů slouží zejména k získání základního přehledu o povaze dat souboru. Pro číselné sloupce funkce spočítá minimální, maximální hodnotu a medián. Dále aritmetický průměr (`Mean`), dolní kvantil (`1st Qu.`) a horní kvantil (`3rd Qu.`).

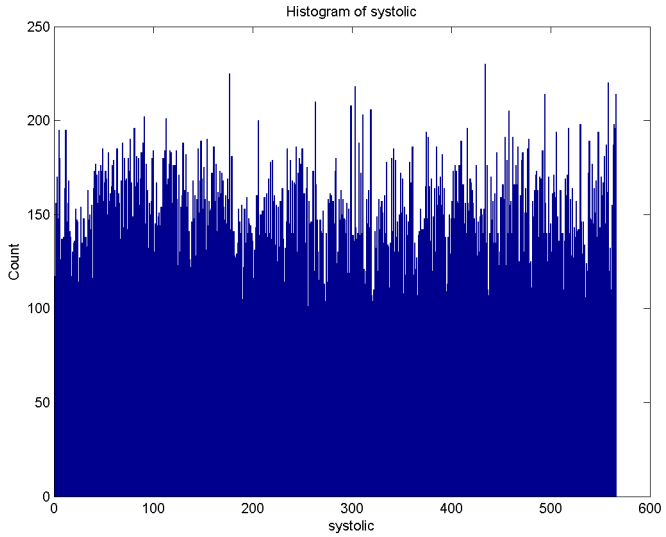
```
sits_id          timepoint      diastolic        systolic
CZUHP40901:  4   After24H:120   Min.   : 10.00   Min.   : 94
CZUHP02601:  4   After2H :121   1st Qu.: 73.00   1st Qu.:140
CZUHP10401:  4   After7D :103   Median : 82.00   Median :154
CZUHP21003:  4   Baseline:222   Mean   : 82.69   Mean   :154
CZUHP11004:  4                  3rd Qu.: 92.00   3rd Qu.:169
CZUHP11006:  4                  Max.   :177.00   Max.   :230
(Other)     :542
```

Výpis kódu 4.5: Výstup funkce `summary` pro celý soubor

4.3.3 Skript v MATLABu

Skript v MATLABu má tentýž účel jako skript v R. Po načtení dat prostřednictvím funkce `textscan` je prováděna základní statistická analýza na základě datového typu daného sloupce. V případě textových řetězců se vytváří pouze histogram ve formátu `png` a `eps`. Ten vzniká zavoláním funkce `hist`. Na obrázku 4.2 jsou naměřené hodnoty systolického krevního tlaku (číselný sloupec).

V případě číselných vstupů se kromě histogramů vytvářejí i soubory obsahující základní statistické veličiny. Na výpisu kódu 4.6 je výsledek výpočtu požadovaných statistik pro jeden číselný sloupec. Po názvu sloupce následuje maximální a minimální hodnota, dále medián, směrodatná odchylka a nakonec střední hodnota. Tyto hodnoty nevznikly zavoláním jedné funkce, jako tomu bylo v případě R. Vznikly zvlášť, konkrétně zavoláním funkcí `max`, `min`, `median`, `deviation` a `mean`. Všechny výstupy jednotlivých sloupců jsou v samostatných souborech.



Obrázek 4.2: Systolický krevní tlak – histogram

```
systolic
max: 230
min: 94
median: 154
deviation: 22.5272
mean: 153.951
```

Výpis kódu 4.6: Základní statistické hodnoty

Nad číselnými sloupci se provádí K–S test, jeden výsledek je ve výpisu kódu 4.7. V případě, že výsledek hypotézy (*Hypothesis test result*) je roven jedné, znamená to zamítnutí nulové hypotézy na hladině významnosti α=0,05.

Test vrací i p–hodnotu aby uživatel věděl, jak moc pravděpodobný je výsledek testu, resp. aby ho mohl lépe porovnat s ostatními výsledky na jiné typy statistických rozdělení. Dále test vrací testovací statistiku a kritickou hodnotu. Na konci souboru je stručný popis, který uživateli osvětluje jaká hodnota, co znamená.

```
BINOMICAL DISTRIBUTION:
Hypothesis test result: 1
P-value: 0
Test statiscic: 0.841345
Critical Value: 0.0567823
```

Výpis kódu 4.7: Část výstupu K–S testu v MATLABu

4.4 R

R [56] je jazyk a prostředí pro statistické výpočty a grafiku. Tento nástroj se od svého uvedení z poloviny 90. let minulého století [62] stal hojně využívaným nekomerčním prostředkem pro statistické analýzy dat.

Jedná se o volně šiřitelný a multiplatformní nástroj. Je podobný jazyku S, z kterého vychází. Ve standardní distribuci a v podpůrných balíčcích je implementované velké množství nejrůznějších statistických a matematických funkcí. Stejně jako MATLAB i R zvládá veškeré maticové výpočty a operace.

Kromě již zmíněného poskytuje i velmi propracovaný grafický výstup a to v podobě 2D i 3D grafů. Ty lze libovolně upravovat, přidávat popisky, měnit barvy i vzhled.

Samozřejmostí tohoto programu je možnost psát vlastní skripty a funkce. Nástroj je možné bezplatně stáhnout z webových stránek projektu pro příslušnou platformu. Tato stránka mimo jiné obsahuje i podrobné návody a dokumentaci programu. Příjemné je i grafické uživatelské prostředí. To je na obrázku v příloze E.1.

Volba vyvíjet právě v tomto nástroji byla podpořena právě díky jeho multiplatformnosti a možnosti automatizace vytvořených programů (skripty). Po zjištění, že program podporuje veškeré statistické metody popsané v kapitole 3 nebylo třeba s použitím R dále neváhat.

Za celou dobu používání R nebyl zpozorován jediný pád tohoto programu, a proto lze prohlásit, že je stabilní. Příkazy se prováděly okamžitě a vykreslování grafů bylo také velmi rychlé. Doba žádného z výpočtů nepřekročila jednu minutu. Tato doba je ale samozřejmně dána povahou výpočtu a velikostí vstupních dat.

4.5 MATLAB

MATLAB [57] je stejně jako R výkonný interaktivní nástroj pro matematické a statistické výpočty, který umožňuje vizualizaci dat. V příjemném grafickém uživatelském prostředí E.2 do sebe integruje velké množství nejrůznějších funkcí. Kromě vědeckotechnických výpočtů totiž umožňuje i modelování, simulace, paralelní výpočty, zpracování a měření signálů, návrhy řídících i komunikačních systémů a mnoho jiného.

Základním datovým prvkem MATLABu je matice. Umí ale pracovat s nejrůznějšími datovými typy. Stejně jako v případě jazyka R, i zde je možné psát vlastní skripty a funkce. Ty se ukládají do M-souborů. Syntaxe MATLABu je v mnohém podobná jazyku C. Obsahuje tak všechny nezbytné příkazy pro psaní programů, jako jsou podmíněné příkazy, větvicí příkazy, cykly a podobně.

MATLAB umožňuje rozšíření pomocí tzv. toolboxů. Jedná se o kolekci M–souborů, která je určena pro řešení jistých tříd problémů (např. `Statistic toolbox`).

Autorem MATLABu je firma The MathWorks, Inc., která svému produktu poskytuje značnou podporu. Kromě školení a prezentací vydává čtvrtletně časopis Newsletter, ve kterém informuje uživatele o nových produktech.

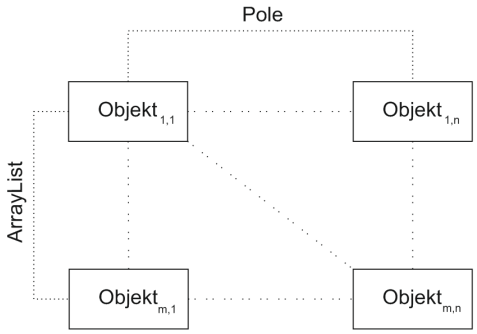
Na rozdíl od R, MATLAB není volně šiřitelný. Pro tuto práci byl vybrán ze stejných důvodů jako jazyk R.

4.6 Statistická analýza závislostí

Tvorba r–skriptů i m–skriptů se uskutečňuje prostřednictvím programu napsaného v Javě. Tento program umožňuje tvorbu skriptů jak pro základní statistickou analýzu, tak i pro analýzu závislostí. Jaký skript bude vytvořen, určuje uživatel na základě uživatelského vstupu přes příkazovou řádku (Linux i Windows).

Vstupem statistické analýzy závislostí je stejně jako v případě základní statistické analýzy CSV soubor s daty, která vznikly SPARQL dotazem na RDF úložiště. Konzolovou aplikací se prochází vstupní soubor a získaná data jsou ukládána do objektů. Tyto objekty mohou nabývat buď číselných, nebo

textových hodnot. Vstupní soubor se prochází pouze jednou. Řádky souboru jsou uloženy do pole (program předem spočítá počet prvků). Tato pole jsou následně uložena do ArrayListu (program předem nepočítá počet prvků, protože by musel projít celým souborem). Grafické znázornění je k dispozici na obrázku 4.3.



Obrázek 4.3: Struktura načtených dat

Po uložení dat program na základě vstupních argumentů spustí příslušnou metodu, které předá potřebné parametry. Původně program samotný počítal i základní statistiku, nicméně tato funkčnost byla pro nadbytečnost odstraněna.

Každá statistická metoda je implementována v samostatné třídě. Ta obsahuje metodu pro vytvoření m–skriptu i r–skriptu. Program zodpovídá za vytvoření souboru se skriptem a za jeho správné pojmenování. V případě chybného uživatelského vstupu informuje uživatele. Skripty se ukládají do adresáře „skripts“. Výstupy skriptů se ukládají do adresáře „output“.

Po provedení příslušné funkce se program ukončí. Výstupem programu je tentokrát pouze jeden skript, který se vytvoří v adresáři „scripts“. O tom, zda se vytvoří skript pro R, nebo pro MATLAB rozhoduje vstupní argument. V případě R je vstupním argumentem „r“, v případě MATLABU pak „m“. Vývojový diagram aplikace je ilustrován na obrázku F.1.

Uživatelský vstup generátoru skriptů

- cesta: absolutní, relativní.
- metody: ttest, ftest, anova1, linr, logr, kwtest, wilcox.

- rozdělení: norm, bin, gamma, poiss.
- parametry: v závislosti na druhu testu jsou to buď číselná označení sloupců, nebo tzv. formuly.

4.6.1 T–test

Vstupem T–testu mohou být jen číselné hodnoty. Výstupem skriptu (R, či MATLAB) je textový soubor, který obsahuje informace o výsledku testu. Možné vstupní argumenty jsou na výpisu kódu 4.8.

```
C:\Users\...\blood.csv ttest norm r 3 4
C:\Users\...\blood.csv ttest norm m 3
```

Výpis kódu 4.8: Spuštění funkce v programu

Prvním argumentem je cesta k datovému CSV souboru, druhým druh metody, třetím je statistické rozdělení (norm=normální), čtvrtým druh výstupního skriptu(m/r) a pak následují čísla sloupců, které se mají porovnávat. V případě jednoho zadaného čísla sloupce se udělá jednovýběrový t–test (výpis kódu 4.10) a v případě dvou zadaných čísel dvouvýběrový t–test (výpis kódu 4.9).

```
Welch Two Sample t-test
data:  data[, 3] and data[, 4]
t = -62.261, df = 994.775, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to
    0
95 percent confidence interval:
 -73.51116 -69.01888
sample estimates:
mean of x mean of y
 82.68551 153.95053
```

Výpis kódu 4.9: Douvýběrový T–test R

Na výpisu kódu 4.9 je výstup r–skriptu. První řádek popisuje název použité metody, kterou zde představuje dvouvýběrový t–test. Na druhém řádku je název dat, která byla k testu použita. V tomto případě jde o třetí a čtvrtý sloupec datového souboru. Dále je zde statistika t, stupeň volnosti (`df`) a p–hodnota. Nízká p–hodnota znamená zamítnutí nulové hypotézy a potvrzení hypotézy alternativní, která je popsána na dalším řádku výstupu. Ta v

tomto případě tvrdí, že skutečný rozdíl ve středních hodnotách je různý od nuly. To znamená, že testované výběry nepocházejí ze stejné populace. Dále výstup obsahuje 95% interval spolehlivosti a vzorek dvou středních hodnot z obou výběrů.

```
T-test:
Hypothesis test result: 1
P-value: 0
Confidence interval 81.4224
Confidence interval 83.9486
Test statistic: 128.577
Degrees of freedom of the test: 565
Estimated population standard deviation: 15.2994
```

Výpis kódu 4.10: Jednovýběrový T–test MATLAB

Na výpisu kódu 4.10 je výstup m–skriptu. První řádek popisuje název použité metody, na druhém řádku je výsledek testu, kde hodnota 1 znamená znamená zamítnutí nulové hypotézy na hladině α. Platnost alternativní hypotézy znamená, že se aritmetický průměr liší od střední hodnoty základního souboru. Poté následuje 95% interval spolehlivosti, hodnota testovací statistiky t, počet stupňů volnosti a odhadovaná standardní odchylka populace od střední hodnoty.

F–test má téměř totožné vstupní parametry jako T–test (až na jméno metody). Výstupy vytvořených skriptů jsou také takřka totožné, a proto zde nejsou popsány.

4.6.2 ANOVA

Vstup této metody se pro jednotlivé skripty liší, to je ostatně patrné z výpisu kódu 4.11. Zatímco pro MATLAB jsou argumenty stále čísla sloupců, v případě R je argumentem tzv. „formula“ složená z názvů sloupců (dle záhlaví) a speciálních znaků (`+,~,*,:,/`). V případě této metody už uživatel opravdu musí vědět, co je jeho záměrem analyzovat. V opačném případě mohou být na výstupu zcela nesmyslná data. Záleží na pořadí argumentů.

```
C:\Users\...\blood.csv anova1 norm r diastolic~systolic
C:\Users\...\blood.csv anova1 norm m 3 4
```

Výpis kódu 4.11: Spuštění funkce v programu

Vysvětlení zápisu formuly:

- $Z \sim X + Y$ znamená nasazení aditivního modelu $Z_i = X_i + Y_i$
- $Z \sim X*Y$ znamená nasazení modelu s interakcemi $Z_i = X_i + Y_i + X_iY_i$
- $Z \sim X/Y$ znamená model „vnořených“ interakcí $Zi = X_i + X_iY_i$
- $Z \sim X : Y$ znamená model pouze s interakcemi $Z_i = X_iY_i$

Výstupem r–skritpu metody ANOVA je výpis kódu 4.12. Dle nízké p–hodnoty (sloupec `Pr(>F)`) lze zamítnout nulovou hypotézu. Z toho plyne platnost alternativní hypotézy, konkrétně závislost testovaných výběrů.

```
             Df Sum Sq Mean Sq F value Pr(>F)
systolic      1  42372   42372   265.9 <2e-16 ***
Residuals   564  89878     159
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Výpis kódu 4.12: Tabulka ANOVA R

Výstupem MATLABu je ANOVA tabulka obr. 4.4 a graf obr. v příloze G.1. Nejdůležitější získanou hodnotou pro určení závislosti výběrů je taktéž p–hodnota (sloupec `Prob>F`). Vzhledem k tomu, že jde v tomto případě o velmi malou hodnotu, můžeme zamítnout nulovou hypotézu. Pak platí alternativní hypotéza. Sledované výběry jsou na sobě závislé.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	60491.9	104	581.653	3.74	1.73176e-022
Error	71758.2	461	155.658		
Total	132250	565			

Obrázek 4.4: Tabulka ANOVA MATLAB

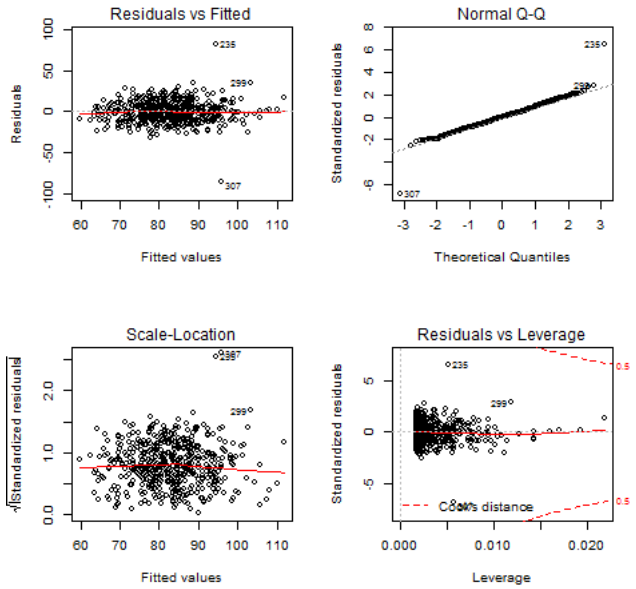
4.6.3 Lineární regrese

U této metody platí naprosto totéž co v přechozím případě. Skripty vrací grafy a soubory s vypočtenými aproximovanýmmi hodnotami. Demonstro-

váno na obrázku v příloze G.2. Na výpisu kódu 4.14 jsou vypočtené polynomiální koeficienty (MATLAB). Příklad spuštění metody v programu 4.13. Grafický výstup r–skriptu metody lineární regrese je na obrázku 4.5.

```
C:\Users\...\blood.csv linr norm r systolic~diastolic
C:\Users\...\blood.csv linr norm m 3 4 4
```

Výpis kódu 4.13: Spuštění funkce v programu



Obrázek 4.5: Závislost systolického tlaku na diastolickém

```
LINEAR REGRESSION
polynomial coefficients:
-4.44927e-008
2.09691e-005
-0.00376736
0.322576
-12.2186
280.629
```

Výpis kódu 4.14: Polynomiální koeficienty lineární regrese

4.6.4 Logistická regrese

Metoda logistické regrese je jedna z mála metod, na jejímž vstupu mohou být i textové hodnoty. Zde už je i podstatný výběr statistického rozdělení (norm=normální, bin=binomické, gamma=gamma, poiss=poissonovo). V případě R je vstupem formula v případě MATLABu sloupce, které chceme analyzovat.

Příklad spuštění příslušných funkcí 4.15.

```
C:\...\blood.csv logr bin r timepoint~diastolic+systolic
C:\...\blood.csv logr bin m 2 4
```

Výpis kódu 4.15: Spuštění funkce v programu

Výpis kódu 4.16 obsahuje výsledky logistické regrese r–skriptu po aplikaci funkce `summary`. Těmi jsou odhady residuálů, koeficienty s odhadem, standardní chybou, t–hodnotou a p–hodnotou $Pr(>|t|)$. Na základě nízké p–hodnoty zamítáme nulovou hypotézu o nezávislosti vstupních parametrů. Závěry plynoucí z výsledků platí pouze pro rozsah hodnot, pro které byl model vytvořen.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-85.775   -8.361    0.063    6.912   82.378

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.50316    3.66799   6.408 3.12e-10 ***
systolic     0.38442    0.02358  16.306  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 159.3573)

    Null deviance: 132250  on 565  degrees of freedom
Residual deviance:  89878  on 564  degrees of freedom
AIC: 4480.5

Number of Fisher Scoring iterations: 2
```

Výpis kódu 4.16: Logistická regrese R

5 Zhodnocení výsledků a diskuze

Výstupem praktické části práce jsou dva analyzační nástroje. Těmi jsou skripty určené k základní popisné charakteristice RDF souboru a generátor skriptů, který je určen k základní statistické analýze a taktéž k analýze závislostí.

V případě základní popisné charakteristiky byly vytvořeny dva znovu spustitelné Shell skripty. Jeden z nich využívá textového přístupu a druhý SPARQL dotazů. Po optimalizaci skriptu využívající textového přístupu byla doba výpočtu obou skriptů identická. Dataset minimal.nt se zpracoval již za 12 vteřin, dataset view.nt za 1 hodinu a 28 minut. V případě největšího datasetu full.nt potřeboval skript využívající SPARQL dotazy 2 hodiny a 46 minut, zatímco skript využívající textový přístup 3 hodniny a 5 minut. Je ale třeba mít na paměti, že tyto časové údaje jsou zcela orientační. Jsou závislé nejen na velikosti operační paměti, rychlosti procesoru a pevného disku, ale i na momentálně spuštěných úlohách. Dalším negativním faktorem může být např. fragmentace disku.

Testy byly prováděny na notebooku Asus K70IO, který měl k dispozici pouze 4 GB RAM DDR2, procesor Core2Duo s maximální pracovní frekvencí 2200Mhz a disk pouze s 5400 ot./min. Vše běželo na 64 bitové verzi operačního systému Ubuntu 12. Předpokládá se, že na výkonnějším HW by se časové údaje zásadně lišily. Za nejnáročnější operaci lze považovat komprimaci, popř. dekomprimaci datasetu. V případě vynechání příkazu pro komprimaci se doba výpočtu zkracuje více než o polovinu. Nejvíce je to patrné u větších datasetů. Pokud by se měl vybrat lepší z obou skriptů tak je to bezesporu skript využívající SPARQL dotazy. Díky MetaMedu je skript navíc schopen pracovat i se vzdálenými úložišti.

Výsledky, které podávají SPARQL dotazy jsou přesnější. Pokud porovnáme výsledky skriptů na RDF datasetu minimal.nt, nalezneme mnoho rozdílů. V tabulce 5.1 je uvedeno několik z nich. Nejmarkantnější rozdíl v tabulce je mezi nezávislými objekty. Zde se hledaný textový řetězec musel v datasetu vyskytovat několikrát, a proto skript pracující bez SPARQL dospěl k chybnému výsledku.

Nenalezl jsem žádná díla jiných autorů, jejichž úkolem by byla tvorba základní popisné charakteristiky RDF souboru.

Sledované prvky	Bez využití SPARQL	S využitím SPARQL
Nezávislé subjekty	7660	7660
Nezávislé objekty	8023	7074
Nezávislé třídy	11	13
Nezávislé predikáty	287	287
Celkový počet entit	2987	2305

Tabulka 5.1: Některé rozdíly v popisné charakteristice

Základní analýza i analýza závislostí je prováděna prostřednictvím jednoho programu – generátoru r–skriptů a m-skriptů. Tento nástroj, který má formu příkazové řádky, má za úkol pouze načíst informace od uživatele a vygenerovat příslušný skript. Vytvořené skripty se vždy upravují na základě vstupního souboru a vstupních argumentů. Nicméně, neznamená to, že pro každý výpočet se bude generovat nový skript. Předpokládá se, že se skripty budou používat opakovaně. V případě, že uživatel vytvoří statistiku nad určitými daty, kterých s každým dnem přibývá, může naplánovat spouštění skriptu např. na každý týden a po čase tak mít souvislou statistiku daného souboru dat. V případě, že se vnitřní struktura těchto dat změní, jednoduše si vygeneruje skript další. Na rozdíl od předchozí základní popisné charakteristiky, tento přístup je zcela platformě nezávislý. Je však závislý na použitých technologiích R a MATLAB.

Základní statistická analýza v R i v MATLABu dopadla téměř totožně. Největší rozdíl je patrný z grafů. Ty sice obsahují stejná data, ale vizuálně vypadají odlišně. Vytvoření skriptu v R bylo pohodlnější díky funkci `summary`. V případě MATLABu bylo sice třeba zavolat více než jen jednu funkci, ale výsledný skript také není příliš složitý. Výsledky K–S testu jsou v obou případech identické. Nejpodstatnější nepříjemností MATLABu byla skutečnost, že pro načtení dat bylo předem nutné znát datový typ všech sloupců, zatímco u R to nebylo nutné. Z těchto důvodů se jeví R jako pohodlnější. Samozřejmě by bylo možné do MATLABu stáhnout rozšiřující funkce (toolbox, M-file), které by tuto problematiku řešily, to ale během realizace nebylo považováno za podstatné. K realizaci postačily základní nástroje. Výsledky řešení základní statistické analýzy byly ověřeny vytvořením programu v jazyce Java.

Pokud by se měly hodnotit výsledky skriptů v R a skriptů v MATLABu v případě analýzy závislostí, tak opět nezbývá než konstatovat, že při stejných vstupních datech jsou téměř identické. Rozdíl je patrný zejména ve formátu výstupu, nikoliv v datech samotných. V praktické části je popsán vždy jeden

a ten samý soubor, jehož struktura je vcelku prostá. To je dáno tím, že v práci byl kladen důraz zejména na pochopení funkce programu a formátu výstupu. Složitější příklady jsou v příloze na CD. Prakticky ve všech úlohách je nejdůležitější nalezení p–hodnoty, na jejímž základě se buď dá, nebo nedá zamítnout nulová hypotéza. U testů závislostí je kladen velký důraz na uživatelský vstup. Je to právě uživatel, kdo vybírá vhodnou metodu a určuje, jaká data se budou porovnávat. Proto mohou být na výstupu často nesmyslné údaje. Jako nejvhodnější metoda pro určování závislostí se jeví metoda logistické regrese. Ostatní metody jsou použitelné pouze pro číselné údaje.

Pokud by měly být shrnuty výhody a nevýhody použitého statistického softwaru, tak vzhledem k téměř identickým výsledkům je nejpodstatnějším rozdílem licence těchto nástrojů. Zatímco R je volně šiřitelný, MATLAB nikoliv. To s sebou nese jistá omezení i rizika. Omezení ve smyslu vyšších počátečních nákladů, rizika ve smyslu možného ukončení vývoje. Pokud uživatel chce využívat pouze statistické funkce a nepotřebuje žádnou z integrovaných funkcí programu MATLAB, je R zjevnou volbou.

Bakalářská práce Zuzany Římské [59] se zabývá metodami analýz závislostí s využitím korelace a logistické regrese v prostředí R. V teoretickém úvodu své práce popisuje některé matematické metody, které byly použity i v této práci. Následně porovnává R s jinými nástroji (SAS, STATA). V praktické části vytváří několik skriptů v jazyce R, na nichž demonstruje použití metod. Data načítá taktéž prostřednictvím funkce `read.table` a následně vytváří generalizovaný lineární model pomocí funkce `glm` s binomickým rozdělením. Nad výsledky následně používá funkci `summary`. Prakticky identický přístup jsem použil v metodě logistické regrese. Autorka taktéž testuje model na hladině významnosti $\alpha = 0,05$. Kromě tohoto skriptu se v práci objevují už pouze dva další skripty, které se ale zabývají geostatistikou. V závěru práce je hodnoceno R jako výkonný statistický software.

Diplomová práce Pavlíny Kuráňové [60] se zabývá metodou logistické regrese jako nástroje pro diskriminaci v lékařských aplikacích. Mimo logistické regrese popisuje ve své práci i testování hypotéz a test dobré shody. Autorka zde ale hlavně popisuje matematický aparát. Nevytváří nástroj pro analýzu, nebo praktické použití logistické regrese, ačkoliv abstrakt práce sliboval implementaci v prostředí MATLAB. Součástí práce nejsou žádné výstupy ve formě zdrojového kódu. Jedinými výstupy jsou tabulky a grafy. Je možné, že zdrojové kódy byly pouze v příloze na CD, které není veřejně přístupné.

Diplomová práce Tomáše Fadrného [61] popisuje statistické zhodnocení dat. Teoretická část obsahuje výčet téměř všech metod, které jsou popsány i v této práci. Ke statistickému zpracování dat však autor užívá programu MS Excel a Minitab 14. Ani v tomto případě nebyly k dispozici datové soubory, či zdrojové kódy, a proto nemohly být výsledky těchto dvou prací porovnány. Avšak na základě obrázků a tabulek obsažených v práci lze říci, že autor v případě metody ANOVA a regresních modelů, sledoval tytéž veličiny, nad kterými na základě p–hodnoty činil totožné závěry.

Statistickou analýzou různých druhů dat se již zabývalo mnoho autorů. Nicméně málokterý vytvářel neinteraktivní nástroj, jehož prostřednictvím by se dala tato statistická analýza zautomatizovat. Mnoho autorů se také zabývalo technologiemi, jako jsou ontologie, RDF data a SPARQL. Málokterý však proto, aby RDF data analyzoval, nebo aby analyzoval fragmenty RDF dat vzniklých SPARQL dotazem.

Závěr

Cílem této práce bylo navrhnout a implementovat řešení pro popisnou charakteristiku, základní statistickou analýzu a analýzu závislostí jednotlivých prvků medicínských dat. Tato data byla uložena v datovém úložišti projektu MRE, který byl v RDF úložišti organizován pomocí řady ontologií. K získání těchto dat byl použit dotazovací jazyk SPARQL.

Nástroj pro vytvoření základní popisné charakteristiky byl implementován v dvojím provedení s využitím základních Unixových nástrojů. První provedení představuje Shell skript využívající textový přístup k získávání potřebných informací. Druhé provedení představuje Shell skript využívající volně šiřitelné externí aplikace MetaMed, jejímž prostřednictvím využívá SPARQL dotazů. Druhé provedení bylo prokazatelně rychlejší na větších datasetech a jeho výsledky byly přesnější.

Dále bylo vytvořeno interaktivní prostředí ve formě konzolové aplikace v jazyce Java. Tato aplikace na základě uživatelského vstupu produkuje r–skripty a m–skripty, které jsou vhodné jak k základní statistické analýze, tak k analýze jednotlivých prvků vstupních dat. Vstupní data v tomto případě představuje CSV soubor, který vznikl spuštěním SPARQL dotazu nad medicínskými daty. Dílčí výsledky jednotlivých metod a jejich interpretace jsou uvedeny přímo u jejich popisu v kapitole 4. Složitější ukázky jsou součástí přiloženého CD. Vzniklé skripty jsou multiplatformní a splňují požadavek na automatizaci.

V průběhu testování se ukázalo, že mezi nejvhodnější statistické metody patří metoda logistické regrese a to zejména proto, že nevyžaduje na svém vstupu pouze číselný vstup, ale umožňuje i vstup kategorizovaných dat. Nicméně je nutné, aby uživatel tvořící skript měl vždy na paměti, jaká data chce sledovat a podle toho určil vhodnou statistickou metodu. V opačném případě může být výstupem aplikace skript produkující zcela nesmyslné vý-

sledky. V případě dat malých rozsahů, nebo dat, která neobsahují reprezentativní prvky, mohou být získané údaje nepřesné, nebo taktéž zcela nesmyslné.

Při porovnání výsledků nástrojů R a MATLAB nelze než konstatovat, že všechny výsledky byly totožné. Lišila se především forma jejich výstupu, nikoliv charakter výsledných dat. Podpora zabudovaných funkcí byla u obou nástrojů obdobná. Pro statistickou analýzu dat bylo prostředí R pohodlnější. Nejzjevnější výhodou R je však to, že se jedná o svobodný software. V případě MATLABu tomu tak není.

Práce by se dala v budoucnu rozšířit o další statistické metody. Dále by se nad aplikací v budoucnu mohlo vytvořit webové prostředí, aby byl generátor skriptů vytvořený v této práci dostupný odkudkoliv.

Přínos je mimo jiné i v tom, že nástroj není omezen na medicínská data. Popisná charakteristika může být provedena nad jakýmkoliv RDF datasetem. Taktéž základní statistická analýza a analýza závislostí může být provedena nad jakýmikoliv daty. Byly splněny všechny cíle zadání této práce.

Seznam obrázků

Seznam tabulek

Seznam zdrojových kódů

Seznam zkratek

RDF	Resource Description Framework – nástroj pro prezentaci, popis a výměnu webových informací.
RDFS	Resource Description Framework Schema – sémantické rozšíření RDF.
URI	Uniform Resource Identificator – jedinečný identifikátor zdroje.
W3C	World Wide Web Consortium – standardizační skupina.
XML	Extensible Markup Language – obecný značkovací jazyk, který byl vyvinut a standardizován. konsorciem W3C.
N–Triples	Serializační formát RDF.
N3	Serializační formát RDF.
RDF/XML	Serializační formát RDF.
RDFa	Serializační formát RDF.
TURTLE	Serializační formát RDF.
DASTA	Ontologie MRE.
DICOM	Ontologie MRE.
SITS	Ontologie MRE.
NIHSS	Ontologie MRE.
ANOVA	Analysis of variance – analýza rozptylu.
CSV	Comma seperated values – hodnoty oddělené čárkou (soubor).

Literatura

[1] RDF WORKING GROUP. Resource Description Framework (RDF). [online]. 10. února 2004 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/RDF/>

[2] KLYNE, Graham a Jaremy J. CARROL. Resource Description Framework (RDF): Concepts and Abstract Syntax. [online]. [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/rdf-concepts/>

[3] LASH, Alex. W3C takes first step toward RDF spec. CNET.com. [online]. 10. března 1997 [cit. 26. června 2013]. Dostupné z: <http://news.cnet.com/2100-1001-203893.html>

[4] ANDREESSEN, Marc. Innovators of the Net: R.V. Guha and RDF. Netscape. [online]. 8. února 1999 [cit. 26. června 2013. Dostupné z: <http://web.archive.org/web/20020606203701/http://wp.netscape.com/columns/techvision/innovators_rg.html>

[5] BERNERS-LEE, Tim. Semantic Web Road map. [online]. 14. října 1998 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/DesignIssues/Semantic.html>

[6] BRICKLEY, Dan. Resource Description Framework (RDF). Mozilla.org [online]. 2007 [cit. 26. června 2013]. Dostupné z: <http://www-archive.mozilla.org/rdf/doc/>

[7] ADUNA. Sesame. ADUNA. OpenRDF.org [online]. 1997 [cit. 26. června 2013]. Dostupné z: <http://www.openrdf.org/>

[8] THE APACHE SOFTWARE FOUNDATION. Apache Jena. [online]. 2011 [cit. 26. června 2013]. Dostupné z: <http://jena.apache.org/>

[9] ŠTENCEK, J. Užití sémantických technologií ve značkovacích jazycích, Bakalářská práce [online]; Vysoká škola ekonomická v Praze, Fakulta

informatiky a statistiky, Katedra informačního a znalostního inženýrství: Nám. W. Churchilla 4 130 67 Praha 3, Prosinec 2009. Dostupné z <http://vse.stencek.com/semanticky-web/>

[10] CLINE, Marshall. C++ FAQ: What's this "serialization"thing all about?. Parashift.com [online]. 4. července 2012 [cit. 26. června 2013]. Dostupné z: <http://www.parashift.com/c++-faq-lite/serialize-overview.html>

[11] SEGARAN, Toby, Colin EVANS a Jamie TAYLOR. Programming the Semantic Web [online]. O'Reilly Media, Inc, July 14, 2009 [cit. 26. června 2013]. ISBN 978-0-596-15381-6. Dostupné z: <http://my.safaribooksonline.com/book/web-development/9780596802141>

[12] GRANT, Jan, Dave BECKETT a Brian MCBRIDE. RDF Test Cases. W3C.org [online]. 2004, 10. února 2004 [cit. 26. června 2013]]. Dostupné z: <http://www.w3.org/TR/rdf-testcases/#ntriples>

[13] BRICKLEY, Dan, R.V. GUHA a Brian MCBRIDE. RDF Vocabulary Description Language 1.0: RDF Schema. [online]. 10. února 2004 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/rdf-schema/>

[14] Semantic web. Ontology. [online]. 13. června 2012 [cit. 26. června 2013]. Dostupné z: <http://semanticweb.org/wiki/Ontology>

[15] GRUBER, Tom. What is an Ontology?. [online]. 1992 [cit. 26. června 2013]. Dostupné z: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

[16] BORST, Willem Nico. Construction of engineering ontologies for knowledge sharing and reuse [online]. Enschede: CTIT, Centre for Telematics and Information Technology, 1997 [cit. 26. června 2013]. ISBN 90-365-0988-2. Dostupné z: <http://doc.utwente.nl/17864/1/t0000004.pdf>

[17] MILLER, Libby. Ontologies and Metadata. [online]. 30. listopadu 2000 [cit. 26. června 2013]. Dostupné z: <http://ilrt.org/discovery/2000/11/lux/>

[18] HOFWEBER, Thomas. Logic and Ontology: 3. Ontology. The Stanford Encyclopedia of Philosophy [online]. 4. října 2004 [cit. 26. června 2013]. Dostupné z: <http://plato.stanford.edu/entries/logic-ontology/#Ont>

[19] VITVAR, Tomáš. Sémantický web: Semantic Web [online]. Praha: Nakladatelství ČVUT [cit. 26. června 2013]. Dostupné z: <http://cvut.cz/pracoviste/odbor-rozvoje/stranky/habilitace-a-inaugurace/habilitacni-prednasky/lecture.pdf>

[20] DASTA: Datový standard pro předávání dat mezi informačními systémy zdravotnických zařízení. [online]. 2012 [cit. 26. června 2013]. Dostupné z: <http://www.dastacr.cz/info.html>

[21] DICOM: The DICOM Standard. Medical imaging & technology alliance. [online]. 2011 [cit. 26. června 2013]. Dostupné z: <http://medical.nema.org/standard.html>

[22] SITS: Safe Implementation of Treatments in Stroke. Karolinska institutet in Sweden. [online]. 2000-2013 [cit. 26. června 2013]]. Dostupné z: <https://sitsinternational.org/>

[23] NIHSS Stroke Scale International. The international electronic education network. [online]. 1999-2010 [cit. 26. června 2013]. Dostupné z: <http://www.nihstrokescale.org/>

[24] MCGUINNESS, Deborah L. a Frank van HARMELEN. OWL Web Ontology Language: Overview. W3C. [online]. 10. února 2004 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/owl-features/>

[25] CONNOLLY, Dan, Frank van HARMELEN, Ian HORROCKS, Deborah L. MCGUINNESS, Peter F. PATEL-SCHNEIDER a Lynn Andrea STEIN. DAML+OIL (March 2001) Reference: Description. [online]. 18. prosince 2001 [cit. 26. června 2013].
Dostupné z: <http://www.w3.org/TR/daml+oil-reference>

[26] Protégé. Standford center for biomedical informatics research. [online]. 2013 [cit. 26. června 2013].
Dostupné z: <http://protege.stanford.edu/>

[27] PRUD'HOMMEAUX, Eric a Andy SEABORNE. SPARQL Query Language for RDF. W3C. [online]. 15. Ledna 2008 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/rdf-sparql-query/>

[28] PRUD'HOMMEAUX, Eric a Lee FEIGENBAUM. Cambridge semantics: SPARQL by Example. W3C. [online]. 2005 [cit. 26. června 2013]. Dostupné z: <http://www.cambridgesemantics.com/semantic-university/sparql-by-example#%281%29>

[29] W3C SPARQL WORKING GROUP. SPARQL 1.1 Overview. W3C. [online]. 21. března 2013 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>

[30] W3C SPARQL WORKING GROUP. SparqlEndpoints: Currently Alive SPARQL Endpoints. W3C. [online]. 16. ledna 2013 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>

[31] SEQUEDA, Juan. SPARQL 101. Cambridge semantics. [online]. 2013 [cit. 26. června 2013]. Dostupné z: <http://www.cambridgesemantics.com/cs/semantic-university/sparql-101>

[32] WYLLYS, Ronald E. Mathematical notes for lis 397.1 introduction to research in library and information science: Statistical hypotheses. The universitz of Texas at Austin School of Inrofmation [online]. 15. ledna 2003 [cit. 26. června 2013]. Dostupné z: <https://www.ischool.utexas.edu/~wyllys/IRLISMaterials/stathyp.pdf>

[33] Stat trek teach yourself statistics: What is Hypothesis Testing?. [online]. 2013 [cit. 26. června 2013]. Dostupné z: <http://stattrek.com/hypothesis-test/hypothesis-testing.aspx>

[34] RNDr. BEDÁŇOVÁ, Iveta Ph.D. a Prof. MVDr. Vladimír VEČEREK, CSc. Základy statistiky: pro studující veterinární medicíny a farmacie. Veterinární a farmaceutická univerzita Brno [online]. 2007 [cit. 2013-06-16]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Skripta.pdf>

[35] NIST/SEMATECH e-Handbook of Statistical Methods. [online]. 2012 [cit. 26. června 2013]. Dostupné z: <http://www.itl.nist.gov/div898/handbook/>

[36] KIRKMAN, T.W. Statistics to Use. [online]. 1996 [cit. 26. června 2013]. Dostupné z: <http://www.physics.csbsju.edu/stats/>

[37] BLACKWELL, Matt. Multiple Hypothesis Testing: The F-test. [online]. 3. prosince 2008 [cit. 26. června 2013]. Dostupné z: <http://www.mattblackwell.org/files/teaching/ftests.pdf>

[38] Parametrické testy – Studentův t-test: (Test rozdílu 2 středních hodnot). Veterinární a farmaceutická univerzita Brno [online]. 2007 [cit. 26. června 2013]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Teorie/Predn3/ttest.htm>

[39] LITSCHMANNOVÁ, Martina. Úvod do statistiky (interaktivní učební text). Vysoká škola báňská – Technická univerzita Ostrava, Západočeská univerzita v Plzni [online]. 2012 [cit. 2013-06-16]. Dostupné z: <http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/interaktivni_uvod_do_statistiky.pdf>

[40] SELTMAN, Howard J. Experimental Design and Analysis. Carnegie Mellon University, Department of Statistics. Pittsburgh. [online]. 10. června 2013 [cit. 26. června 2013]. Dostupné z: <http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>

[41] JONES, James. Stats: One-Way ANOVA. Richland Community College. Decatur. [online]. 2013 [cit. 26. června 2013]. Dostupné z: <http://people.richland.edu/james/lecture/m170/ch13-1wy.html>

[42] Analýza rozptylu (ANOVA): (testování rozdílu více středních hodnot). Veterinární a farmaceutická univerzita Brno [online]. 2007 [cit. 26. června 2013]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Teorie/Predn3/ANOVA.htm>

[43] MCDONALD, J.H. Handbook of Biological Statistics: Kruskal–Wallis test and Mann–Whitney U test. Sparky House Publishing, Baltimore, Maryland. [online]. 2009 [cit. 26. června 2013]. Dostupné z: <http://udel.edu/~mcdonald/statkruskalwallis.html>

[44] HENDL, Jan. Přehled statistických metod zpracování dat: analýza a metaanalýza dat. 1. vyd. Praha: Portál, 2004, 583 s. ISBN 80-717-8820-1.

[45] Wilcoxonův test. Veterinární a farmaceutická univerzita Brno [online]. 2007 [cit. 26. června 2013]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Teorie/Predn4/Wilcoxon.htm>

[46] MCDONALD, J.H. Handbook of Biological Statistics: Wilcoxon signed-rank test. Sparky House Publishing, Baltimore, Maryland. [online]. 2009 [cit. 26. června 2013]. Dostupné z: <http://udel.edu/~mcdonald/statsignedrank.html>

[47] DRÁPELA, Karel a Jan ZACH. Statistické metody I.: pro obory lesního, dřevařského a krajinného inženýrství. Vyd. 1. V Brně: Mendelova zemědělská a lesnická univerzita, 1999, 135, [16] s. ISBN 80-715-7416-3.

[48] DRÁPELA, Karel. Statistické metody II: pro obory lesního, dřevařského a krajinného inženýrství. Vyd. 1. V Brně: Mendelova zemědělská a lesnická univerzita, 2000, 144, [8] s. ISBN 80-715-7474-0.

[49] Linear regression. Yale University, Department of Statistic. [online]. 16. září 1997 [cit. 26. června 2013]. Dostupné z: <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>

[50] Stat trek teach yourself statistics: What is Linear Regression?. [online]. 2013 [cit. 26. června 2013]. Dostupné z: <http://stattrek.com/regression/linear-regression.aspx>

[51] DRÁPELA Logistická (logitová) regrese. Mendelova univerzita v Brně. [online]. 7. února 2008 [cit. 26. června 2013]. Dostupné z <http://user.mendelu.cz/drapela/Statisticke_metody/Prezentace/ostatni/folie_logit1.doc>

[52] Logistická regrese. [online]. 25. ledna 2012 [cit. 26. června 2013]. Dostupné z: <http://www.trilobyte.cz/downloadfree/qcemanual/logreg.pdf>

[53] VČELÁK, Petr a Jana KLEČKOVÁ. Extrakce metadat z medicínských dat: Medicical Meta Data Extraction and Manipulation Project. [online]. 2012 [cit. 26. června 2013].

[54] VČELÁK, Petr, Jana KLEČKOVÁ, Michal KRATOCHVÍL a Vladimír ROHAN. METAMED: Medical Meta Data Extraction and Manipulation Tool Used in the Semantically Interoperable Research Information System. In Biomedical Engineering and Informatics (BMEI). 2012, 5rd International Conference on 2012. (IEEE Catalog Number: CFP1293D-CDR)

[55] ALEXANDER, Keith, Michael HAUSENBLAS a Jun ZHAO. Describing Linked Datasets with the VoID Vocabulary. Yale University, Department of Statistic. [online]. 3. březen 2011 [cit. 26. června 2013]. Dostupné z: <http://www.w3.org/TR/void/>

[56] The R project for statistical computing. [online]. 16. května 2013 [cit. 26. června 2013]. Dostupné z: <http://www.r-project.org/>

[57] MATLAB: The Lenguage of Technical Computing. The Mathworks, Inc. [online]. 1994-2013, [cit. 26. června 2013]. Dostupné z: <http://www.mathworks.com/products/matlab/>

[58] ZEMSKÝ, Igor. Vizualizace RDF dat [online]. Brno, 2009 [cit. 26. června 2013]. Dostupné z: <http://is.muni.cz/th/60726/fi_m_a2/zemsky_dp.pdf>. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce RNDr. Tomáš Gregar.

[59] ŘÍMSKÁ, Zuzana. Metody analýzy závislosti s využitím korelace a logistické regrese v prostředí R[online]. Olomouc, 2010 [cit. 26. června 2013]. Dostupné z: <http://theses.cz/id/5as367/128859-818719391.pdf>. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta. Vedoucí práce Mgr. Pavel Tuček.

[60] KURÁŇOVÁ, Pavlína. Logistická regrese jako nástroj pro diskriminaci v lékařských aplikacích [online]. Ostrava, 2009 [cit. 26. června 2013]. Dostupné z: <http://www.am.vsb.cz/theses/mgr/2009/pdfs/kuranova.pdf>. DIPLOMOVÁ PRÁCE. VŠB – Technická univerzita Ostrava Fakulta elektrotechniky a informatiky Katedra aplikované matematiky. Vedoucí práce doc.Ing. Radim Briš, Csc.

[61] FADRNÝ, Tomáš. Statistické zhodnocení dat: Statistical data evaluation [online]. Brno, 2009 [cit. 26. června 2013]. Dostupné z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=16255>. Diplomová práce. Vyskoké Učení Technické v Brně, Fakulta strojního inženýrství. Vedoucí práce doc. RNDr. Bohumil MAROŠ, CSc.

[62] GENTLEMAN Robert and Ross IHAKA. Lexical scope and statistical computing. Journal of Computational and Graphical Statistics, 9:491-508, 2000.

A RDF Serializace

```
_:a <http://www.w3.org/1999/02/22-rdf-syntaxns#type>
<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#patient>.

<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#patient>
<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#firstname>
"Pavel" .

<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#patient>
<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#lastname>
"Cihlar" .

<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#patient>
 <http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#weight>
"105" .

<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#patient>
<http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#
    hasClinicalEvent>
<http://priklad.cz/> .
```

Výpis kódu A.1: Formát N–Triples

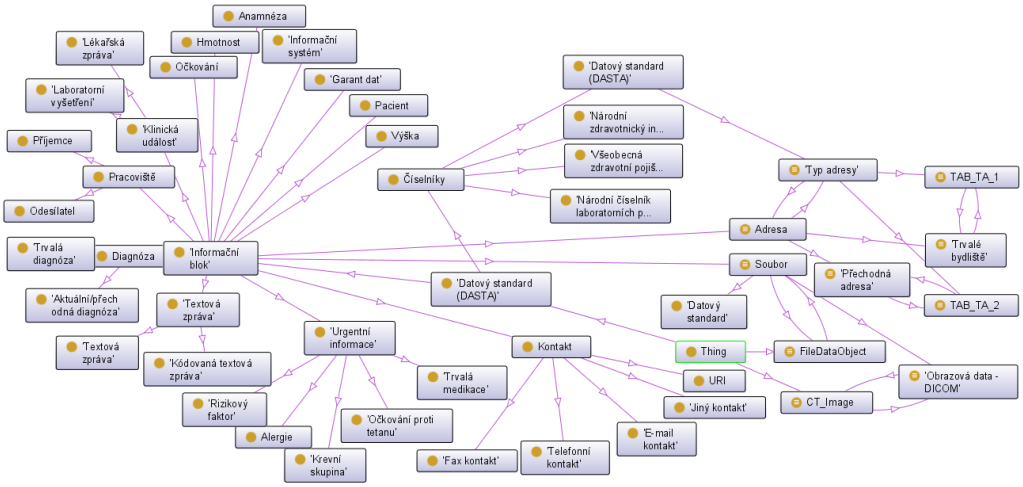
```
@prefix rdf:type: <http://www.w3.org/1999/02/22-rdf-syntaxns#
    type> .
@prefix dasta: <http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl
    > .
@prefix patient: <http://mre.kiv.zcu.cz/ontology/2012/01/dasta.
    owl#patient> .

[] a rdf:type:description;
      dasta:patient;
        patient:firstname "Pavel";
        patient:lastname "Cihlar";
        patient:weight "105";
        patient:hasClinicalEvent <http://priklad.cz/> .
```

Výpis kódu A.2: Formát N3

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntaxns#"
xmlns:dasta="http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#"
xmlns:patient="http://mre.kiv.zcu.cz/ontology/2012/01/dasta.owl#
    patient"
 <rdf:dasta>
   <dasta:patient>
      <patient:firstname>Pavel</v:name>
      <patient:lastname>Cihlar</v:role>
      <patient:weight>105</v:org>
      <patient:hasClinicalEvent rdf:resource="http://priklad.cz/
          "/>
   </dasta:patient>
 </rdf:dasta>
</rdf:RDF>
```

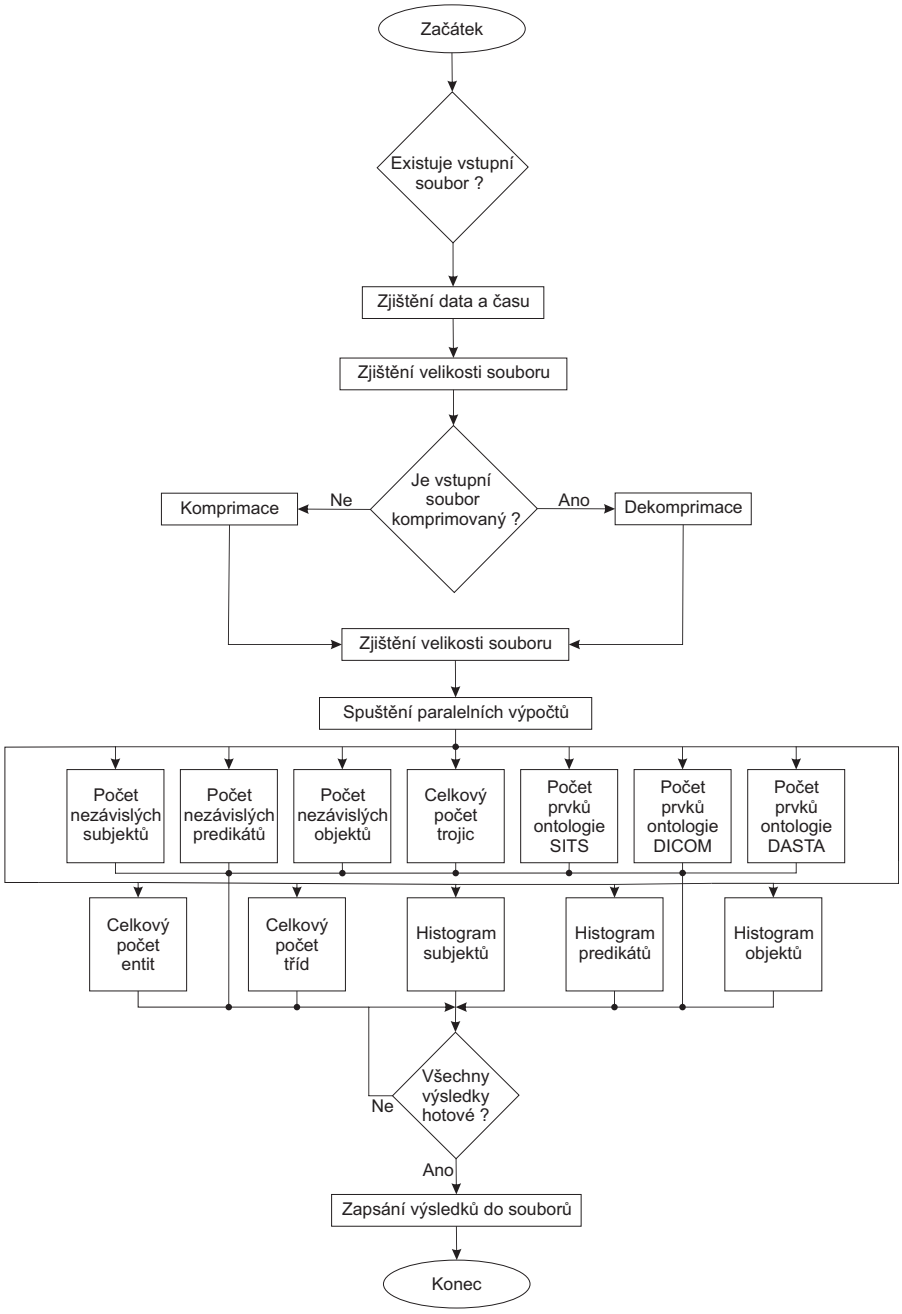
Výpis kódu A.3: Formát RDF/XML

B Graf ontologie



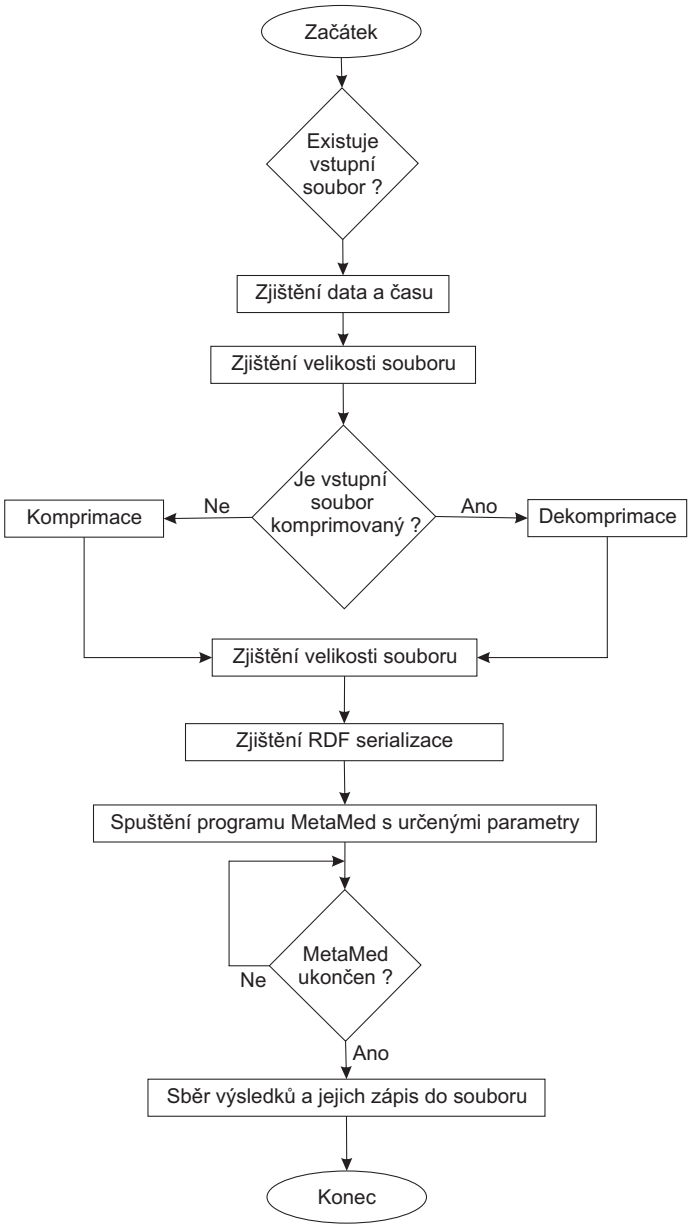
Obrázek B.1: Graf ontologie DASTA

C Datový soubor N–Triples



Obrázek C.1: Vývojový diagram Shell skriptu bez využití SPARQL

D RDF úložiště



Obrázek D.1: Vývojový diagram Shell skriptu s využitím SPARQL

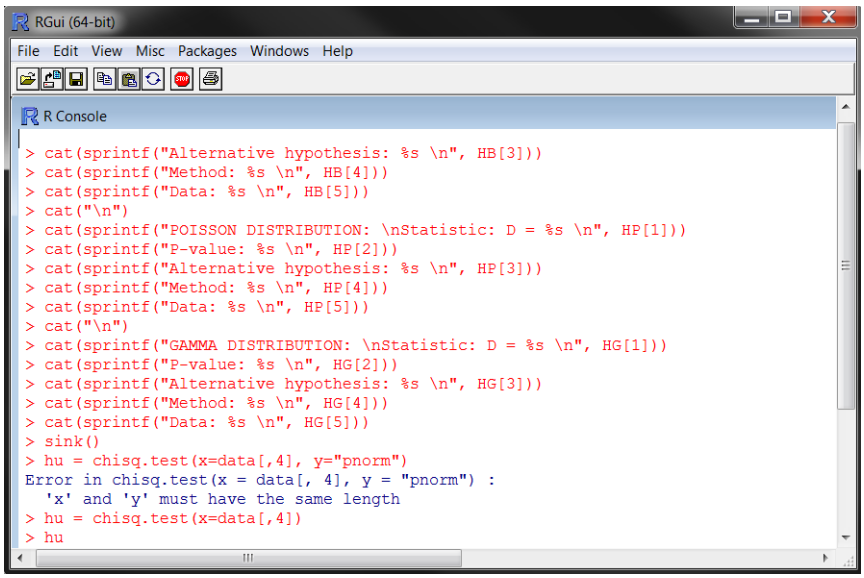
Document	Medical RDF Dataset
Date	27.01.2013
Code	Minimal.nt
Serialization format	N-triple
Uncompressed byte size	37984
Uncompressed size	38M
Compressed byte size	2176
Compressed size	2,2M
All triples	116720
Distinct subjects	7660
Distinct objects	7074
Distinct classes	13
Distinct predicates	287
Total entities	2305

Tabulka D.1: Příklad základní popisné charakteristiky

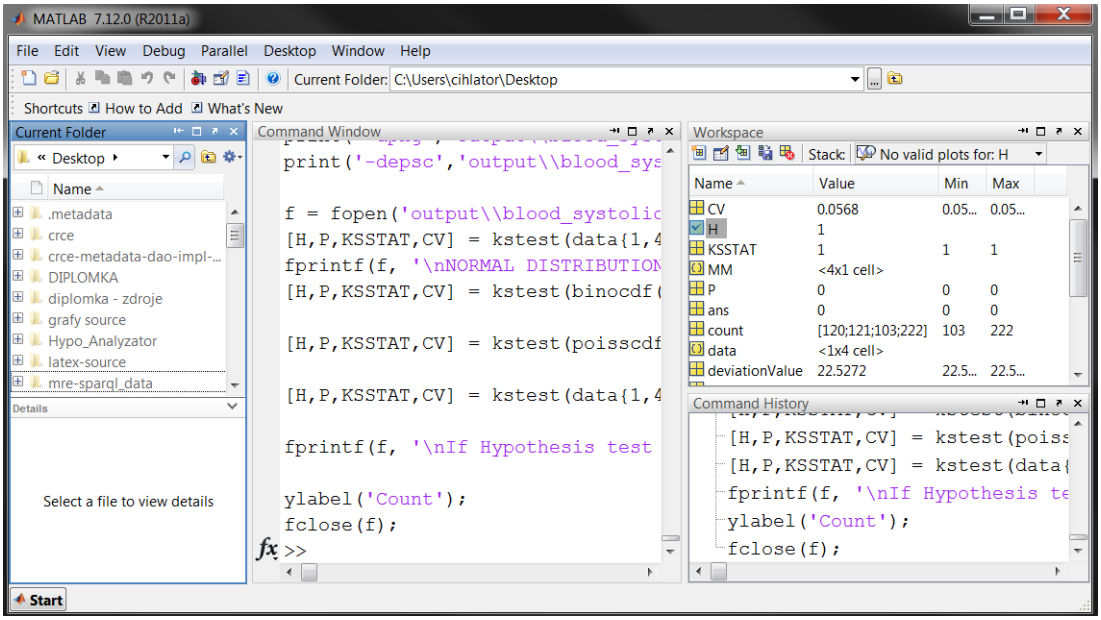
```
@prefix void: <http://rdfs.org/ns/void#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wv: <http://vocab.org/waiver/terms/norms> .
@prefix sd: <http://www.w3.org/ns/sparql-service-description#> .

:minimal a void:Dataset;
    dcterms:title "minimal";
    dcterms:description "anonymous medical dataset of all meta
       data";
    dcterms:source "minimal.nt";
    dcterms:created "2013-01-08"^^xsd:date;
    void:feature <http://www.w3.org/ns/formats/N-Triples>;
    void:triples 116720
;    void:classes 13
;    void:entities 2305
;    void:distinctSubjects 7660
;    void:distinctObjects 7074
;    void:properties 287 .
```

Výpis kódu D.1: Ukázka popisné charakteristiky ve formátu VoID

E GUI statistického SW

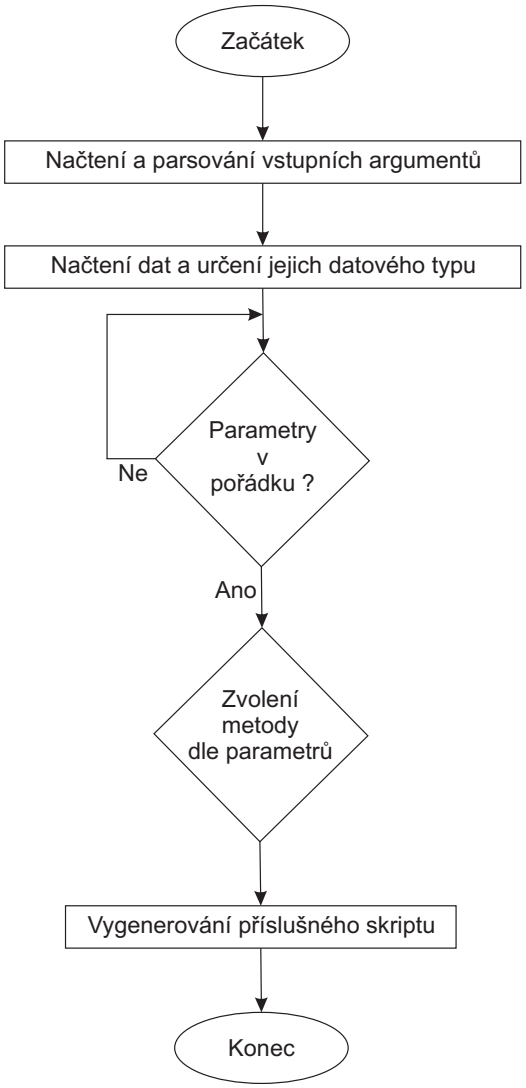


Obrázek E.1: Grafické uživatelské prostředí R



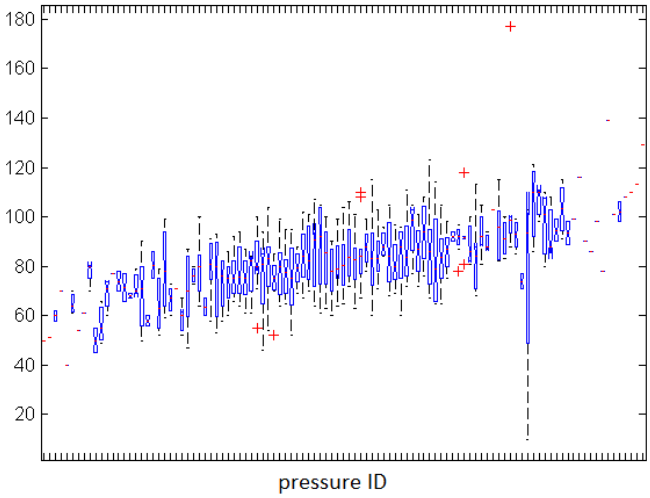
Obrázek E.2: Grafické uživatelské prostředí MATLAB

F Vývojový diagram aplikace

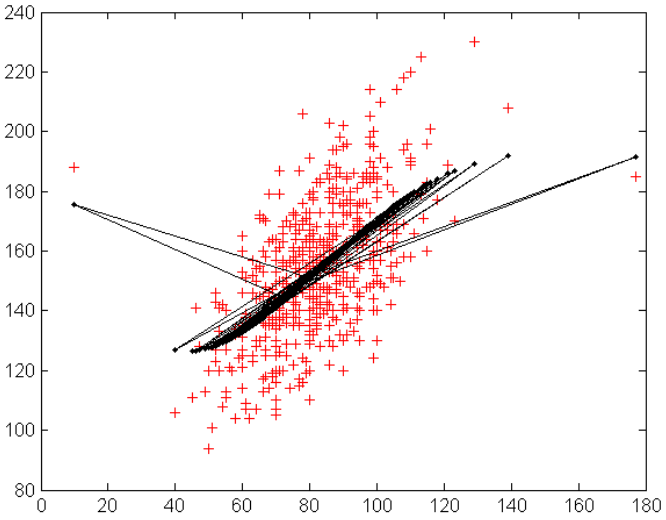


Obrázek F.1: Vývojový generátoru skriptů

G Výstupy statistických metod



Obrázek G.1: ANOVA graf MATLAB



Obrázek G.2: Graf lineární regrese v MATLABU