Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Bakalářská práce

Webová demonstrace základních statistických výpočtů s využitím matematického software R a SAGE

Jana Ernekerová

Vedoucí práce: Ing. Daniel Vašata

12. května 2015

Poděkování

V první řadě bych chtěla poděkovat mému vedoucímu Ing. Danielu Vašatovi za skvělé vedení mé práce, cenné rady, nekonečnou trpělivost a ochotu odpovídat na všechny moje otázky. Dále bych ráda poděkovala své rodině za podporu během celého mého studia a svým přátelům, kteří jsou mi oporou a motivují mě v mém studiu dál pokračovat.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 12. května 2015 .....

Odkaz na tuto práci

Ernekerová, Jana. *Webová demonstrace základních statistických výpočtů s využitím matematického software R a SAGE*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá možnostmi integrace volně dostupných matematických algebraických systémů R a Sage do webové aplikace. Napojení statistického softwaru R do webové aplikace bylo provedeno s využitím API poskytovaného projektem OpenCPU, napojení matematického softwaru Sage za pomoci služby Sage Cell Server. Oba zvolené matematické systémy se podařilo úspěšně využít ve webové aplikaci postavené na jazyce PHP. Výsledkem je jednoduchá webová aplikace pro základní statistické výpočty. Hlavním přínosem práce je rozbor možností využití systémů R a Sage ve webové aplikaci a jejich porovnání z hlediska jednoduchosti integrace, efektivity a praktické použitelnosti.

Klíčová slova Webová aplikace, Základní statistické výpočty, Počítačový algebraický systém, Matematický software R, SageMath, PHP, OpenCPU, Sage Cell Server

Abstract

This thesis deals with options of the integration open source calculating algebraic systems R and Sage into the web application. The connection of R software into the web application was done using API provided by OpenCPU project, the connection of Sage was done with Sage Cell Server service. Both

selected algebraic systems were successfully used in the web application built on PHP language. The result is simple web application for basic statistical calculations. The main contribution of this thesis is the analysis of the possibility of using software R and Sage in the web applications and their comparison in terms of ease of integration, effectivity and practical applicability.

Keywords Web application, Basic statistical calculations, Computer Algebra System, Mathematical software R, SageMath, PHP, OpenCPU, Sage Cell Server

Obsah

Úvod **1**

1 Analýza: R **3**
1.1 Úvod do R 3
1.2 Webové rozhraní pro R 4
1.3 OpenCPU 7
1.4 Jak používat OpenCPU 9

2 Analýza: Sage **19**
2.1 Seznámení se Sage 19
2.2 Statistické výpočty v Sage 20
2.3 Sage Cell Server 22

3 Vybrané statistické metody **25**
3.1 Základní pojmy teorie pravděpodobnosti 25
3.2 Vybraná spojitá rozdělení 27
3.3 Bodové odhady 28
3.4 Odhady tvaru rozdělení 32
3.5 Testování hypotéz 33

4 Tvorba testovací aplikace **39**
4.1 OpenCPU 40
4.2 Sage Cell Server 43
4.3 Srovnání systémů a vyhodnocení 45

Závěr **49**

Literatura **51**

A Seznam použitých zkratek **57**

B Obsah přiloženého CD **59**

Seznam obrázků

1.1 Ukázka popisu balíčku R [25] 10
1.2 Nastavení webhook na GitHub pro umístění R balíčku na veřejném serveru [27] 14

2.1 Jednoduché použití příkazu `interact` pro výpočet faktoriálu. . . . 21

4.1 Formátování výstupu kódu R v buňce Sage Cell Server. 43

Seznam tabulek

3.1 Jednovýběrový t-test o střední hodnotě [54, tab. 9.3.1] 35
3.2 Dvouvýběrový t-test rovnosti středních hodnot pro $\sigma_1^2 = \sigma_2^2$ [54, tab. 9.3.5] 36
3.3 Dvouvýběrový t-test rovnosti středních hodnot pro $\sigma_1^2 \neq \sigma_2^2$ [54, tab. 9.3.6] 36
3.4 Párový t-test rovnosti středních hodnot [54, tab. 9.3.7] 37

Úvod

Matematická statistika a web

Matematická statistika jako vědní disciplína se od počátku svého vývoje, na začátku 18.století [1], rozšířila téměř do všech empirických vědních disciplín. Počátky jejího vývoje jsou nejvíce spojovány s přírodními vědami, jako je biologie, astronomie a psychologie, dnes má významnou roli také v oblastech teorie měření, statistické fyziky, matematické lingvistiky, demografie a pojistné matematiky, ale také epidemiologie, robotiky a mnoha dalších. Tato oblast matematiky má široké využití nejen v akademickém světě, ale také v běžném životě. Hlavní úlohou statistiky je analýza dat obsahujících prvek nahodilosti. S takovým vyhodnocením dat, tedy statistikou, se denně setkává každý, aniž si to mnohdy uvědomuje.

Počítačových algebraických systémů (anglicky Computer Algebra System - CAS) je již v dnešní době velké množství. Jde zpravidla o komplexní programy, které bývají účinným nástrojem k počítání náročných úloh z různých vědních disciplín. Některé z těchto programů začínají pronikat také do oblasti webových technologií a nabízejí různé způsoby online prezentace výpočtů i matematických demonstrací. Tyto systémy najdeme jak komerční, tak také Open Source. Tato bakalářská práce se zaměřovala výhradně na Open Source systémy a to konkrétně software Sage a statistický software R.

Web má v dnešní společnosti obrovský význam, nároky na webové aplikace se čím dál více zvyšují a vyžadují složité analýzy online. Počítačové algebraické systémy byly k takovým složitým analýzám vytvořeny, a tak je výhodné se pokusit je integrovat do webových aplikací a využít tak jejich vysoký analytický potenciál.

Autorčinou motivací k výběru tohoto tématu pro její bakalářskou práci byl hlavně velký zájem o matematiku a touha spojit matematiku a informatiku do jednoho uceleného projektu. Znalosti získané během tvorby této práce ji tak jistě budou nápomocné jak v jejím dalším studiu, tak i v budoucím zaměstnání.

Cíle práce

Prvním cílem této bakalářské práce bylo prozkoumat možnosti integrace volně dostupných matematických systémů R a Sage do webových aplikací.

Z tohoto důvodu se práce nejprve zabývá možnostmi napojení statistického software R do webových prezentací s využitím API poskytovaného projektem OpenCPU. V další části pak obdobně možnostmi napojení matematického software Sage za pomoci služby Sage Cell Server. Cílem analytické části práce také bylo prozkoumat závislost těchto řešení na konkrétních webových technologiích.

Dalším cílem a také cílem praktické části práce bylo navrhnout a vytvořit jednoduchou webovou aplikaci testující možnosti obou těchto systémů na několika základních statistických úlohách.

Posledním cílem práce poté bylo obě řešení otestovat a provézt jejich porovnání z hlediska jednoduchosti integrace, efektivity a praktické použitelnosti.

Struktura práce

První část práce se věnuje analýze možností integrace R do webových aplikací. Rozebírá možnosti, které pro integraci můžeme využít a poté se zaměřuje na analýzu API projektu OpenCPU. Analyzuje jeho přednosti a poté popisuje jak lze tento systém využívat.

V druhé části obdobně analyzuje možnosti integrace Sage do webových aplikací, zaměřuje se pak na službu Sage Cell Server. Také rozebírá možnosti použití Sage ke statistickým výpočtům.

Další kapitola je poté věnována teorii k vybraným metodám z matematické statistiky, které jsou předmětem výpočtu v testovací aplikaci.

V poslední části se práce zabývá samotnou tvorbou aplikace, návrhu jednotlivých částí, integrací R a Sage do webové aplikace a také implementací jednotlivých výpočtů. Na závěr provedeme srovnání obou systémů.

Kapitola 1

Analýza R a možnosti jeho použití ve webové aplikaci - OpenCPU

1.1 Úvod do R

R je softwarové prostředí pro statistické výpočty a grafiku. Při dodržení podmínek GNU General Public License nadace Free Software Foundation je volně šířitelný a nevylučuje komerční využití programu. Jedná se o projekt podobný jazyku a prostředí S vyvinutému v Bell Laboratories (dříve AT&T, nyní Lucent Technologies) Johnem Chambersem a jeho kolegy a lze ho považovat za odlišnou implementaci jazyka S. Jsou mezi nimi některé významné rozdíly, ale většina kódu napsaného pro S bude zároveň fungovat i v R [2].

Je multiplatformní - je dostupné pro širokou škálu UNIXových prostředí, pro Windows i MacOS X [3]. Zdrojové kódy a předkompilované verze programu R pro nejběžnější operační systémy jsou dostupné na stránkách nadace R Foundation for Statistical Computing (CRAN).

R není jen prostředí, ale jde zároveň také o jednoduchý a efektivní programovací jazyk obsahující podmínky, cykly, uživatelem definované rekurzivní funkce, prostředky pro vstup a výstup. Kromě velkého množství statistických a grafických technik, poskytuje prostředky pro manipulaci a ukládání dat, sadu operátorů pro výpočty na polích a maticích, rozsáhlé, konzistentní a integrované prostředky pro analýzu dat, grafické prostředky pro analýzu a zobrazování dat a to ať již na obrazovce nebo v tištěné podobě. Pomocí balíčků (*packages*) je uživatelem velice snadno rozšiřitelné o další metody. Přibližně osm balíčků je součástí oficiální distribuce R. Mnohé další balíčky pokrývající velmi rozsáhlou oblast moderní statistiky jsou dostupné přes CRAN.

Snadnost, s kterou lze vytvářet dobře navrhnuté obrázky a grafy patří k největším přednostem tohoto prostředí. Do grafů lze v případě potřeby

snadno vkládat matematické symboly a vzorce a uživateli je ponechána plná kontrola nad výsledným vzhledem grafu.

Další výhodou je možnost naprogramování výpočetně náročných postupů v C, C++ nebo Fortranu a jejich následného připojení k R a volání za běhu [2].

R má také svůj vlastní, LaTeXu podobný, formát pro tvorbu dokumentace, jenž je používán při poskytování dokumentace on-line, či v tištěné podobě a to hned v několika formátech.

1.2 Webové rozhraní pro R

Pro R již bylo vyvinuto mnoho webových rozhraní. Několik možností lze nalézt na stránkách samotného projektu R [4]. Informace na těchto stránkách ale nejsou aktuální, a tak se situace již značně změnila.

Rweb Tento projekt byl vyvinut Jeffem Banfieldem, z Montana State University, Montana, USA v letech 1997-99 především pro potřeby studentů statistiky na této univerzitě [5]. *Rweb* je sada webových stránek a Perl skriptů, které poskytují webové rozhraní založené na R a které jeho autor nechává volně k dispozici. Webové stránky tohoto projektu [5] a článek autora tohoto projektu vydaný v roce 1999 v internetovém periodiku Journal of Statistical Software [6] informují uživatele o třech dostupných verzích tohoto rozhraní: základní Rweb, jenž od svých uživatelů požaduje znalost jazyka R a který má běžet ve většině prohlížečů, poté sofistikovanější verzi rozhraní v jazyce Javascript a Rweb moduly. Všechny tři verze mají zároveň možnost načítání dat uživatele. Ty jsou webovým rozhraním načteny pomocí funkce R *read.table*.

Příklady užití v článku autor dokládá obrázky a výstupy z programu, poslední aktualizace stránek ale proběhla v červnu 1999 a většina odkazů na návody a dokumentaci nefunguje. Lze najít ještě stránky Jeana Thioulouse z roku 2004 [7], které jsou celé postavené na tomto webovém rozhraní, ale jejich zdrojové kódy jsou v této době pravděpodobně jedinou dostupnou „dokumentací“ k tomuto projektu, což ho v zásadě činí nepoužitelným.

R-Online Webové rozhraní, které vytvořil Ulf Bartel. Jde ale o podobný příklad jako předchozí projekt. Poslední aktualizace na stránce proběhla v roce 2005 a ačkoliv je v dokumentaci (pouze v němčině) napsáno o mnoha funkcích, jediný odkaz na příklad hlásí chybu na serveru [8].

Web Decomp, E-Decomp *Web Decomp* je statistický plugin pro MS Excel a *E-Decomp* webové rozhraní pro R. Stránky projektu však nenabízí žádný funkční příklad použití a dokumentace k webovému rozhraní je pouze v japonštině [9], což značně zužuje okruh potenciálních uživatelů.

Rserve, FastRWeb Jde o projekt Simona Urbanka. Jedná se o TCP/IP server, který umožňuje využít prostředí R z různých programovacích jazyků bez nutnosti jeho inicializace nebo odkazů na R knihovnu. Každé připojení má zvlášť oddělený pracovní prostor a adresář. Implementace strany klienta je k dispozici pro rozšířené jazyky jako je C/C++, PHP nebo Java. *Rserve* podporuje vzdálené připojení k serveru, autentizaci a přenos souborů. Typické použití je integrování backendu R pro výpočet statistických modelů, grafů a dalších funkcí v různých aplikacích [10].

Verze *Rserve* pro použití na webu se jmenuje *FastRWeb*, má svoje vlastní webové stránky a dokumentaci, i když ne tak bohaté jako samotný Rserve. FastRWeb je infrastruktura, která umožňuje jakémukoliv webserveru použít R scripty pro generování obsahu za běhu, jako jsou webové stránky a grafika. URL jsou mapovány do skriptů a mohou mít volitelné argumenty, které jsou poslány funkcím R běžícím ve skriptu. Například volání

```
http://my.server/cgi-bin/R/foo.png?n=100
```

by způsobilo, že FastRWeb zavolá funkci ve skriptu foo.png.R s parametrem n:

```
fce_v_foo.png.R(n="100").
```

FastRWeb může běžet na jakémkoliv webserveru, který podporuje CGI (Common Gateway Interface) nebo PHP [11].

Stránky Rserve byly naposledy aktualizovány v roce 2013 a stránky FastRWeb v roce 2012. Je možné zde najít několik odkazů na aplikace využívající Rserve a psaných v jazycích C/C++ nebo Javě, a také webové aplikace. Dokumentace, i když stručná nám poskytuje kompletní návod, jak tento projekt využít.

CGIwithR Jedno z webových rozhraní pro R, o kterých bylo publikováno v elektronickém časopise Journal of Statistical Software. David Firth, autor projektu, v článku *CGIwithR: Facilities for processing web forms using R* [12] seznamuje s tímto balíčkem pro použití se statistickým výpočetním prostředím R. Balíček má usnadňovat zpracovávání informací z webových formulářů a zobrazování výsledků v HTML přes CGI protokol. To je protokol určený pro komunikaci mezi webovým serverem a programem generujícím dynamický obsah. CGIwithR umožňuje jednoduché použití R jako skriptovacího jazyka CGI, čímž uživateli zpřístupňuje rozsáhlé statistické zázemí prostředí R v CGI skriptech, aniž by se musel nejprve učit jiné programovací jazyky, například Perl, který je pro CGI skripty velice populární.

Stránky projektu neodkazují na žádné online příklady, ale je zde několik ukázkových souborů, které jsou volně ke stažení [13]. A ačkoliv článek o projektu v Journal of Statistical Software je již z roku 2003, poslední aktualizace na webu projektu proběhla v loňském roce (2014) a zdrojové kódy jsou k dispozici ke stažení přímo na jeho stránkách nebo na GitHub.com.

Webbioc *Webbioc* byl součástí projektu *Bioconductor* a poskytoval webové rozhraní pro použití balíčků R vytvořených v rámci tohoto projektu. Bioconductor je open source softwarový projekt poskytující nástroje pro analýzu a porozumění genomickým datům (data spojená s analýzou dědičné informace organismu - genomu [14]) a je založen především na programovacím jazyce R. Na webu projektu Bioconductor už ale nelze najít žádné informace o tomto webovém rozhraní. Ačkoliv projekt Bioconductor je stále aktivní, poskytuje pouze R balíčky pro výpočty spojené s oblastí genetiky.

Rook, RApache, brew Nástroje pro R vytvořené Jeffreym Hornerem. Prvním z nich je R balíček *Rook*. Rook je webové rozhraní a zároveň softwarový balíček pro R, který definuje rozhraní mezi webovým serverem a R aplikací. Aplikace Rook je doslova funkce R, která přijímá jako vstup prostředí R a vrací list příslušných HTTP položek (status, HTTP hlavičku a tělo HTML dokumentu) jako výstup [15]. Dalším autorovým počinem je *RApache*, projekt podporující vývoj webových aplikací za použití prostředí a jazyka R a webového serveru Apache [16]. A třetím, také nezanedbatelným projektem je balíček *brew* pro tvorbu zpráv, v nichž může uživatel kombinovat souhrnné tabulky a grafy z R s rozsáhlými texty [17].

Rwui On-line aplikace, která se používá k vytvoření webového rozhraní pro běh R skriptu. Kód je generovaný automaticky, takže rozhraní pro R skript může být stáhnuto a rozběhnuto během několika minut a uživatel nemusí mít žádné znalosti z oblasti programování webových aplikací [18]. Autor skriptu vytvoří webovou aplikaci vyplněním informací v sekvenci několika po sobě jdoucích webových formulářů, poté je mu vygenerovám kód aplikace, kterou si stáhne a nainstaluje na vlastní server.

R-php Projekt rozvíjený na University of Palermo v Itálii. Jeho autory jsou Angelo M. Mineo a Alfredo Pontillo a jeho cílem je vytvořit webově orientovaný statistický software. *R-php* je vytvářen v PHP a MySQL, je to volně šířitelný software s všeobecnou veřejnou licencí GNU. Implementuje dva moduly: První umožňuje jednoduché vkládání kódu R a zobrazování výsledků (analýz a grafů) na jiné stránce. Druhý modul provádí statistické analýzy za použití grafického uživatelského rozhraní. Projekt vznikl v roce 2005 a není jasné, zda na něm autoři stále pracují.

Na webu projektu najdeme jak příklady využití, tak kompletní dokumentaci v italštině, a také v angličtině [19]. Využití R-php je však vhodné pouze v aplikacích založených na jazyce PHP.

R PHP ONLINE CGI webové rozhraní pro chod R programů online, včetně grafického výstupu. Autorem je Steve Chen z TKU, Taipei, Taiwan. Projekt je vydán pod veřejnou licencí a může být použit jak ve Windows, tak v linuxovém prostředí. Jde o čisté CGI rozhraní, kód R tedy není volán přes PHP knihovnu. Funguje tak, že nejprve deleguje program v R do příslušného R balíčku a poté zobrazuje výstupy online [20]. Poslední novinka k projektu je na stránkách datována rokem 2003 a velká část webu je v čínském jazyce.

Shiny Framework pro tvorbu webových aplikací v R. Balíček shiny byl na CRAN přidán teprve 20.února letošního roku (2015), jde tedy o poměrně nový projekt. Umožňuje velice snadné vytváření interaktivních webových aplikací s R. Jeho součástí je mnoho ovládacích prvků pro manipulaci s daty (zaškrtávací a přepínací políčka, posunovače, tlačítka, výběrová pole, formuláře pro textové i numerické vstupy a formuláře pro vložení souborů) [21]. Interaktivita, kterou tento framework nabízí jinak statickému prostředí R a zároveň fakt, že uživatel nepotřebuje znát HTML, CSS, ani JavaScript patří k jeho velkým přednostem. Naopak se ale uživatel musí seznámit se strukturou a jazykem shiny a samozřejmě znát jazyk R. Uživatel má možnost využít pro své shiny aplikace online cloud. Tato služba je k dispozici buď s omezenými možnostmi zdarma, nebo v různých úrovních placená. Na placeném účtu má pak uživatel prostor pro více aplikací, e-mailovou podporu, možnost autentifikace uživatelů při vstupu do jeho aplikací a hlavně možnost více hodin jejich aktivního užívání. Stejně tak si může uživatel vybrat ze dvou verzí serveru. Zdarma má k dispozici ke stažení tzv. Open Source Edition verzi balíčku pro instalaci serveru, placená je potom verze Professional Edition. Ta má samozřejmě zase více možností a vychytávek. Projekt není vhodný pro použití z externích webových aplikací.

1.3 OpenCPU

1.3.1 Proč OpenCPU

Je zřejmé, že používání R ve webových aplikacích se v dnešní době těší vysoké oblibě, a uživatelům se tak již nabízí široké možnosti. Pro integraci statistického softwaru R do webové aplikace je v této bakalářské práci použit *OpenCPU* systém a to hned z několika důvodů.

Tento projekt nabízí HTTP API pro použití jazyka R ve webové aplikaci. Jde o čisté spojení R a webu. OpenCPU je kompatibilní s jakýmkoliv

jazykem nebo frameworkem, který umožňuje komunikaci pomocí protokolu HTTP. Zároveň se uživatel nemusí učit žádný složitý nový jazyk, či systém, pouze propojuje funkce R se svou webovou aplikací.

Narozdíl od většiny projektů z předchozí kapitoly je OpenCPU projekt nový a stále se rozšiřuje. Web projektu nám nabízí kompletní, ucelenou a přehlednou dokumentaci a online běžící vzorové aplikace včetně možnosti stažení jejich zdrojových kódů. V OpenCPU aplikaci můžeme použít jakékoliv funkce a balíčky R, můžeme použít funkce ostatních uživatelů OpenCPU, nebo si napsat vlastní. OpenCPU aplikace je vlastně balíček R, je tedy možné ho použít lokálně v R, můžeme ho rozběhnout na vlastním serveru, nebo použít jeden z veřejných serverů, které OpenCPU poskytuje.

Další nezanedbatelnou výhodou tohoto systému je, že je pro všechny uživatele zcela zdarma a je také využitelný pro komerční účely.

1.3.2 Seznámení s OpenCPU

Projekt vznikl jako součást (post) doktoranského výzkumu Jeroena Oomse na katedře statistiky na University of California v Los Angeles, USA [22]. Verze 1.0 byla představena v srpnu 2013 a od té doby bylo vydáno několik aktualizací. V současné době (duben 2014) poslední aktualizace na blogu autora pojektu uvádí verzi OpenCPU 1.4.6. [23].

OpenCPU slouží pro podporu on-line zabudovaných vědeckých výpočtů a reprodukovaných vědeckých výzkumů. OpenCPU server poskytuje spolehlivé a spolupráce schopné HTTP API pro analýzu dat postavenou na jazyce a prostředí R. Uživatelé mohou využít vlastní server, nebo server tímto projektem veřejně poskytovaný. Ten díky použití webhook obsahuje vždy nejaktuálnější verzi našeho balíčku. OpenCPU poskytuje také klientskou knihovnu JavaScriptu postavenou na jQuery, která umožňuje volat funkce R za použití AJAXu.

Velmi diskutovaný je rozdíl mezi projektem Shiny a OpenCPU systémem. Hlavním rozdílem je účel, ke kterému tyto projekty slouží. Shiny je framework pro tvorbu interaktivních webových demonstrací v R, naproti tomu OpenCPU je API, tedy tvoří pouze most, který spojuje webovou část s částí analytickou obsahující volané funkce R.

OpenCPU je kompatibilní s jakýmkoliv jazykem nebo frameworkem, který umožňuje komunikaci pomocí protokolu HTTP. Uživatel není omezen sadou dostupných panelů nebo widgetů. OpenCPU spravuje příchozí požadavky, bezpečnost, přidělování zdrojů, vstup a výstup dat a další technické náležitosti. Nic víc, nic míň.

Rozdíl se systémem Shiny je také v tom, jak klient udržuje spojení se serverem. V Shiny podobně jako v terminálu R jde o klasický stavový protokol, tedy klient se serverem udržují neustálé spojení, relaci, a tak server pozná, že po sobě jdoucí požadavky spolu souvisí, tedy že pocházejí od stejného klienta. OpenCPU je ale čisté HTTP, tedy protokol bezestavový, a tak ačkoliv

poskytuje funkce pro relační objekty, funguje trochu jinak, než jsme u stavových aplikací zvyklí. Po každém volání funkce, OpenCPU ukončí (případně zabije) R proces, který byl použit ke zpracování požadavku. Nicméně, všechny výstupy každého volání funkce, jako návratové hodnoty, grafika nebo soubory v pracovním adresáři, jsou uloženy na serveru a ID relace je vrácena klientovi. To může být poté použito v budoucích HTTP požadavcích. Například může klient získat výstupy v různých formátech, sdílet je s ostatními, nebo použít uložené R objekty jako argumenty v následujících voláních funkcí. Objekty nejsou nijak přiřazeny globálnímu prostředí a to má hned několik výhod: všechno je asynchronní, a tak GUI nebude blokováno během čekání na výstup z R; další výhodou je, že pokud se volání R zasekne, chybuje nebo havaruje, aplikace nespadne; aplikace díky tomuto provedení počítají už z principu paralelně, klienti tedy mohou provádět požadavky současně a spojit jejich výstupy později [24].

Způsob, jakým jsou odděleny GUI a funkce R poskytují základy pro týmovou spolupráci na projektu. Je tak umožněno, že weboví vývojáři mohou použít jejich oblíbený jazyk pro volání R funkcí přes HTTP. OpenCPU poskytuje most mezi těmito dvěma systémy, a tak se analytik, který vytváří funkce pro datovou analýzu v R nemusí vůbec zabývat vývojem webu a zároveň se webový vývojář vůbec nemusí dostat do styku s jazykem R. Tyto dvě vrstvy jsou v aplikaci naprosto odděleny.

Všechny části OpenCPU jsou zveřejněny pod licencí Apache2, může tak být šířen a upravován jako open source, nebo použit pro komerční účely.

1.4 Jak používat OpenCPU

Tato sekce hovoří o tom, jak OpenCPU používat, co všechno je k jeho chodu potřeba, kam umístit funkce, které chce uživatel volat z webové stránky a jak k nim přistupovat. Sekce je rozdělena do několika podkapitol. První podkapitola je o funkcích R, poté následuje serverová část a až nakonec část o tom, jak k funkcím přistupovat z webové aplikace.

1.4.1 Funkce R

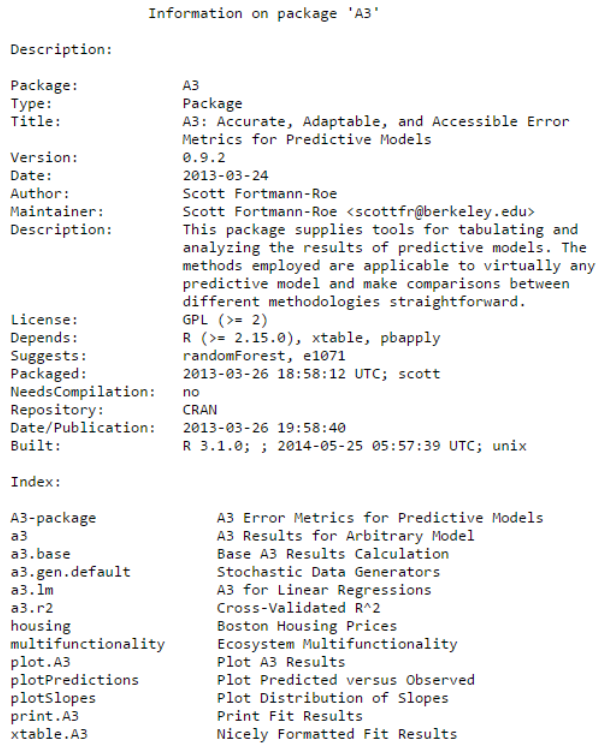
Má-li uživatel již vlastní běžící webovou aplikaci, do nichž bude výpočty R integrovat, nebo s tvorbou celé aplikace teprve začíná, každopádně bude muset zjistit, jaké funkce R k jeho záměrům bude potřebovat. Není vždy nutné tvořit hned vlastní, je možné použít jakékoliv funkce z veřejně dostupných balíčků. Pokud chce tvořit OpenCPU aplikaci, tedy balíček, který zároveň obsahuje webové stránky aplikace, vyžaduje OpenCPU k udržení přehlednosti, aby byly v této aplikaci použity pouze funkce z tohoto balíčku. Pro použití funkcí z jiných balíčků je nutné použít obalovací funkce a správně deklarovat závislosti na těchto balíčcích. Princip obalovací funkce spočívá v tom, že vlastně sama žádný výpočet nedělá, pouze volá funkci, případně funkce, které

výpočet provádějí. Deklarovat závislosti je nutné v souboru NAMESPACE, jenž je povinnou součástí každého R balíčku a dostaneme se k němu později. Seznam dostupných balíčků na veřejném serveru OpenCPU je k dispozici zde: `https://demo.ocpu.io/`.

Informace o daném balíčku umístěném na tomto serveru je možné najít na adrese:

`https://demo.ocpu.io/{název_balíčku}/info`

Tedy například na adrese `https://demo.ocpu.io/A3/info` jsou informace o balíčku *A3*. Struktura tohoto výpisu je vidět na obrázku 1.1 .



```
                Information on package 'A3'

Description:

Package:            A3
Type:               Package
Title:              A3: Accurate, Adaptable, and Accessible Error
                    Metrics for Predictive Models
Version:            0.9.2
Date:               2013-03-24
Author:             Scott Fortmann-Roe
Maintainer:         Scott Fortmann-Roe <scottfr@berkeley.edu>
Description:        This package supplies tools for tabulating and
                    analyzing the results of predictive models. The
                    methods employed are applicable to virtually any
                    predictive model and make comparisons between
                    different methodologies straightforward.
License:            GPL (>= 2)
Depends:            R (>= 2.15.0), xtable, pbapply
Suggests:           randomForest, e1071
Packaged:           2013-03-26 18:58:12 UTC; scott
NeedsCompilation:   no
Repository:         CRAN
Date/Publication:   2013-03-26 19:58:40
Built:              R 3.1.0; ; 2014-05-25 05:57:39 UTC; unix

Index:

A3-package              A3 Error Metrics for Predictive Models
a3                      A3 Results for Arbitrary Model
a3.base                 Base A3 Results Calculation
a3.gen.default          Stochastic Data Generators
a3.lm                   A3 for Linear Regressions
a3.r2                   Cross-Validated R^2
housing                 Boston Housing Prices
multifunctionality      Ecosystem Multifunctionality
plot.A3                 Plot A3 Results
plotPredictions         Plot Predicted versus Observed
plotSlopes              Plot Distribution of Slopes
print.A3                Print Fit Results
xtable.A3               Nicely Formatted Fit Results
```

Obrázek 1.1: Ukázka popisu balíčku R [25]

Tato stránka s popisem obsahuje informace obsažené v souboru *DESCRIPTION*, který je jednou z povinných součástí každého R balíčku. Najdeme zde název, verzi, jméno autora, licenci, pod kterou je balíček zveřejněn, slovní popis funkcionalit, datum vydání a další nepovinné atributy. Pod základními charakteristikami balíčku je zobrazen výpis funkcí, které balíček obsahuje a u každé popis této funkce z její manuálové stránky. Seznam funkcí obsažených v balíčku lze najít také na stránce:

`https://demo.ocpu.io/{název_balíčku}/R`

a kompletní kód konkrétní funkce potom na:

`https://demo.ocpu.io/{název_balíčku}/R/{název_funkce}`

kompletní kód funkce *a3* balíčku *A3* tedy je na adrese: `https://demo.ocpu.io/A3/R/a3`. Dále seznam manuálových stránek balíčku:

`https://demo.ocpu.io/{název_balíčku}/man`

a konkrétní manuálovou stránku obdobně na:

`https://demo.ocpu.io/{název_balíčku}/man/{název_funkce}`

Takto může uživatel najít funkci, kterou použije ve své webové aplikaci.

Pokud chce uživatel použít funkci, která není v některém z těchto balíčků, je nutné sestavit vlastní R balíček, a tam funkci umístit.

Zdrojové kódy balíčku se skládají z podadresáře obsahujícího soubory DESCRIPTION (popis) a NAMESPACE (jmenný prostor) a podadresářů R: data, demo, exec, inst, man, po, src, tests, tools a vignettes (některé z nich mohou chybět, ale žádný by neměl být prázdný). Podadresář balíčku může také obsahovat soubory INDEX, configure, cleanup, LICENSE, LICENCE a NEWS. Ostatní soubory jako je INSTALL (pro instrukce k nestandartní instalaci), README, nebo ChangeLog bude R ignorovat, ale mohou být užitečné pro koncového uživatele. V hlavní složce balíčku by se neměly vyskytovat skryté soubory (začínající tečkou). Volitelné soubory configure a cleanup jsou (Bourne shell) skripty, které jsou spuštěny před a (za předpokladu, že byla volána volba –clean) po instalaci na operačních systémech založených na bázi linuxu, analogicky na systému Windows jsou configure.win a cleanup.win.

Název podadresáře balíčku by se měl jmenovat stejně jako samotný balíček. Protože některé systémy nejsou citlivé na velká a malá písmena je k udržení přenositelnosti silně doporučeno neužívat velikost písmen k odlišení názvů balíčků. Například máme-li vytvořen balíček *package*, není vhodné vytvářet nový balíček pojmenovaný *Package*. Také soubory ve stejném podadresáři není vhodné odlišovat pouze rozdílnou velikostí písmen. Z důvodu přenositelnosti mezi systémy musí také názvy balíčků obsahovat pouze ASCII znaky a znaky z anglické abecedy, není ale povoleno v názvech používat speciální znaky a mezery. Zároveň není povoleno použít několik speciálních vyhrazených názvů. Balíčky jsou obvykle distribuovány jako tarball (archiv s koncovkou .tar) a ty mají limit na délku cesty 100 bytů. Zdrojový balíček by, je-li to možné, neměl obsahovat spustitelné soubory [26].

Soubor DESCRIPTION obsahuje popis balíčku. Povinnými údaji jsou *Package* - název balíčku, *Version* - verze, *License* - druh licence, *Description* - popis funkcionality, *Title* - titul, *Author* - autor a *Maintainer* - udržovatel, to by mělo obsahovat jméno a jeho kontaktní emailovou adresu, všechny ostatní atributy jsou volitelné. V souboru jsou vždy atributy uvedeny v tomto formátu:

```
Název atributu: obsah
```

tedy například:

```
Version: 1.0.1
```

Tento soubor by měl být psán pouze v ASCII znakové sadě, pokud to není možné, musí obsahovat také atribut *Encoding* - kódování.

Podadresář *R* obsahuje pouze soubory s kódem R. Ty bývají nejčasteji zakončeny koncovkou .R, ale mohou být také .S, .q, .r, nebo .s. Podadresář *man* obsahuje pouze dokumentaci ve formátu Rd (koncovka .Rd nebo .rd). Všechny funkce a objekty by měli mít svojí dokumentaci. Zdrojové a hlavičkové soubory pro zkompilovaný kód jsou v podadresáři *src*. Podadresář *demo* je pro R skripty (běžící přes demo()), které demonstrují některou z funkcionalit balíčku. V podadresáři *tests* najdeme pro balíček specifické dodatečné testy. *exec* může obsahovat další spustitelné skripty, které balíček potřebuje, typicky pro interprety jako je shell, Perl nebo Tlc. Podadresář *po* se používá pro soubory určené pro překlad chybových a varovných hlášení do různých jazyků. Podadresář *tools* je místo pro pomocné soubory potřebné při konfiguraci. Obsah podadresáře *inst* bude rekurzivně zkopírován do instalačního adresáře. Podadresáře tohoto adresáře by neměly zasahovat do těch podadresářů, které jsou užívány R (tedy R, data, demo, exec, man). Obvykle se do tohoto podadresáře vkládají soubory jako například CITATION (užívá funkce pro citování), AUTHORS, COPYRIGHTS nebo NEWS. Vytváří-li uživatel OpenCPU aplikaci, z konvence se webové stránky umisťují do složky inst/www/.

Při vytváření balíčku stačí složky R, man a soubory DESCRIPTION a NAMESPACE.

R má systém řízení jmenných prostorů pro kód v balíčcích. Tento systém umožňuje autorovi balíčku specifikovat, které proměnné mají být exportovány, a tak zpřístupněny uživatelům balíčku a které mají být importovány z ostatních balíčků. Zjednodušeně ke každé funkci, kterou chce tvůrce balíčku zpřístupnit jeho uživatelům, přidá do souboru NAMESPACE řádku pro export této funkce:

```
export(název_funkce)
```

Díky tomu bude tato funkce přístupná uživatelům balíčku. Naopak bude-li v balíčku chtít použít funkce z balíčku jiného, je nutné je do balíčku importovat:

```
importFrom("název balíčku", název_fce1, název_fce2, název_fce3)
```

V příkazové řádce se R balíček nainstaluje příkazem:

```
install.packages("cesta_k_našemu_zabalenému_balíčku")
```

Před použitím funkce z balíčku, je nutné nejprve balíček načíst z knihovny příkazem:

```
library(název_balíčku)
```

1.4.2 Serverová část

Jsou dvě možnosti, které může uživatel využít - vlastní nebo veřejný server. Pro lokální testování a spouštění OpenCPU aplikací je možné využít ještě třetí možnost, tzv. single-user server. Chce-li uživatel použít vlastní server, je nutné na něm nainstalovat několik balíčků, ze kterých se tento systém skládá. OpenCPU cloud server běží na Ubuntu 14.04 nebo vyšším a aktuální a podrobný návod na jeho instalaci je možné najít na této adrese: `http://jeroenooms.github.io/opencpu-manual/opencpu-server.pdf`.

Pro využití veřejného serveru, je třeba umístit balíček na GitHub. Stačí ho vložit do nového repozitáře na účtu na tomto serveru a v nastavení repozitáře přidat webhook. Ten zajistí, že veřejný server OpenCPU balíček najde a bude-li mít správnou strukturu, automaticky ho u sebe nainstaluje. Aktualizuje ho pak po každém 'pushnutí' balíčku do repozitáře (je možné nastavit, po jaké akci server reaguje a balíček aktualizuje). Je nutné dodržet následující pravidla: podadresář R obsahující zdrojové kódy R je umístěn v kořenové složce balíčku; jméno GitHub repozitáře je stejné jako název balíčku; uživatelský účet na GitHub má veřejnou emailovou adresu. Správné nastavení webhooku najdete na obrázku 1.2. Po úspěšné instalaci na veřejný server OpenCPU, je opět popis balíčku k dispozici: `https://{uživatelské_jméno_na_GitHub}.ocpu.io/{název_balíčku}/info`. V případě, že nemohlo dojít k jeho instalaci, znamená to, že nebyla dodržena jedna z podmínek uvedených v tomto odstavci, balíček nemá všechny povinné součásti, případně nesplňuje předepsanou strukturu nebo nebyl správně nastaven webhook u repozitáře. V případě, že nelze nainstalovat novější verze balíčku, popřípadě funkce, server nutně nehlásí chybu, ale ponechává si starší funkční verzi.

Single-user server vhodný pro lokální spuštění OpenCPU aplikací je vlastně balíček R s názvem *opencpu*, lze tak jednoduše v R nainstalovat příkazem:

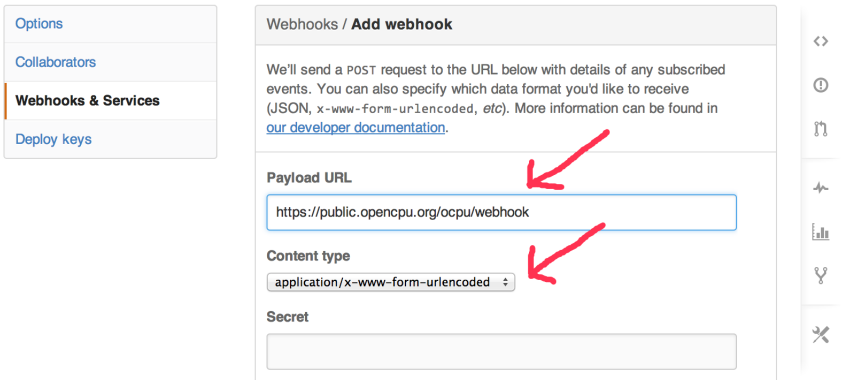
```
install.packages("opencpu")
```

Poté spustit příkazem:

```
library(opencpu)
```

Pak je možné jednoduše načíst aplikaci z GitHub pro lokální použití (k tomu je potřeba ještě balíček *devtools*, který obsahuje funkci install_github()), funkce opencpu$browse("cesta_k_balíčku") otevře aplikaci ve výchozím prohlížeči.

```
install_github("uživatel/název_balíčku")
opencpu$browse("library/název_balíčku/www")
```



Obrázek 1.2: Nastavení webhook na GitHub pro umístění R balíčku na veřejném serveru [27]

Argument funkce opencpu$browse je v tomto případě cesta k balíčku nainstalovaném v jedné z globálních knihoven na serveru. Chceme-li tedy například nainstalovat a spustit lokálně v prohlížeči OpenCPU aplikaci *gitstats* z veřejného repozitáře použijeme:

```
install_github("opencpu/gitstats")
opencpu$browse("library/gitstats/www")
```

Online pak tuto aplikaci najdeme na adrese: `https://public.opencpu.org/ocpu/library/gitstats/www`.

Aplikace konkrétních uživatelů na účtu na GitHub s přidaným webhook jsou vždy na adrese: `https://{uživatelské_jméno}.opencpu.org/{název_balíčku}/www`.

1.4.3 Volání R funkcí z webové stránky

OpenCPU v současné době používá jen HTTP metody GET a POST. GET se používá k načítání zdrojů, tedy například k načtení souboru nebo objektu, a POST k RPC (Remote Procedure Call), tedy k vzdálenému volání funkcí nebo spouštění skriptů. Požadavek POST je platný pouze s cílovou URL skriptu nebo funkce. Seznam návratových stavových kódu OpenCPU, které by klient měl být schopen interpretovat je uveden na adrese: `https://www.opencpu.org/api.html#api-status`.

K testování je možné také použít tuto testovací stránku OpenCPU: `https://public.opencpu.org/ocpu/test/`. Nebo je možné z příkazové řádky použít nástroj *curl*, který slouží k přenosu dat z/na server a jedním z protokolů, které podporuje je také HTTP. Při použití metody POST, slouží k přidání argumentů přepínač '-d'. Například k volání funkce rnorm s parametry n=10 a mean=5, z balíčku stats (jedna z oficiálních aplikací OpenCPU) použijeme:

```
curl https://public.opencpu.org/ocpu/library/stats/R/rnorm -d
"n=10&mean=5"
```

Pokud jde o jiný, než veřejný server OpenCPU, nahradíme adresu veřejného serveru:

```
https://{adresa_serveru}/ocpu/library/{název_balíčku}/
R/{název_funkce}
```

Kořenová složka /ocpu/ je výchozí nastavení serveru, jeho administrátor ho však může změnit.

Opět pokud jde o balíček konkrétního uživatele na veřejném serveru opencpu, bude URL:

```
https://{uživatelské_jméno}.opencpu.org/
{jméno_balíčku}/R/{název_funkce}
```

Jako výstup testovaného volání funkce dostaneme:

```
/ocpu/tmp/x0a411c0928/R/.val
/ocpu/tmp/x0a411c0928/stdout
/ocpu/tmp/x0a411c0928/source
/ocpu/tmp/x0a411c0928/console
/ocpu/tmp/x0a411c0928/info
/ocpu/tmp/x0a411c0928/files/DESCRIPTION
```

Kde „x0a411c0928“ je ID relace. Relace je vlastně kontejner, který obsahuje prostředky vytvořené vzdáleným voláním funkce nebo skriptu. Například cesta urlhttps://public.opencpu.org/ocpu/tmp/x0a411c0928/R/.val/print drží výstupní hodnoty funkce. Významy těch ostatních a vůbec všech možných cest, které můžete dostat jako návratovou hodnotu naleznete tady: `https://www.opencpu.org/api.html#api-session` a tady potom tabulku možných výstupních formátů: `https://www.opencpu.org/api.html#api-formats`.

Konkrétní kód, který bude volat funkci R z balíčku se odvíjí od použitého programovacího jazyka.

Další možnost je k volání funkcí R a k zobrazování jejich výstupů na stránce použít JavaScriptovou knihovnu opencpu.js, pak je nutné do hlavičky stránky zahrnout odkazy nejprve na javascriptovou knihovnu jquery.js a poté na knihovnu opencpu.js (je nutné dodržet toto pořadí). Je vhodné místo online odkazů na knihovnu přidávat použitou aktuální verzi jako součást balíčku, protože je knihovna v aktivním vývoji a aktuální verze se může čas od času radikálně změnit. Knihovna opencpu.js staví na knihovně jQuery a umožňuje tak volat R funkce pomocí AJAXu (Asynchronous JavaScript and XML). Knihovna funguje ve všech moderních prohlížečích, a je primárně určena pro vývoj aplikací. Úložiště veřejných OpenCPU aplikací je k dispozici na adrese `http://github.com/opencpu`.

Doporučená struktura pro OpenCPU aplikace je, aby byly webové stránky součástí balíčku R, podle konvence jsou tyto stránky umístěny v balíčku v podsložce /inst/www/. Díky tomu, že jsou stránky součástí balíčku, jsou aplikace snadno distribuovatelné a je možné je používat offline. Navíc to zajišťuje, že frontend aplikace a backend (kód R) jsou synchronizovány. Je zde ale také možnost použít tzv.CORS - Cross Domain OpenCPU Request, a tak použít knihovnu opencpu.js z externího webu. V tomto případě je nutné specifikovat adresu externího serveru OpenCPU pomocí ocpu.seturl():[24]

```
ocpu.seturl("//public.opencpu.org/ocpu/library/{název_balíčku}
/R").
```

Knihovna implementuje dvě základní bezestavové funkce k volání funkcí R, jednu pro funkce generující grafy, druhou pro funkce vracející nějaká data. Ty jsou snadno použitelné, protože berou přímo výstup z funkce R a není zapotřebí žádná správa relací. Pomocí pluginu rplot můžeme zobrazovat grafy v divu na naší webové stránce:

```
$("#div_id").rplot(fun, [,args] [,callback])
```

Funkce vrací jqXHR jQuery objekt. Argument *fun* obsahuje název funkce R, *args* je pole parametrů R funkce a *callback* je pro callback funkci (funkci zpětného volání) a pro zobrazení grafu v divu na stránce není potřeba, volá se jen pro relační objekty. Tato funkce zobrazí graf na webové stránce v divu, jehož id je div_id. Pro volání funkce R vracející data, využijeme funkci:

```
ocpu.rpc(fun, [,args] [,complete])
```

Funkce opět vrací jqXHR jQuery objekt. Argumenty *fun* a *args* mají stejný význam jako u předchozí funkce a argument *complete* je opět callback funkce, volá se pouze v případě úspěchu a má jeden argument, kterým je návratová hodnota funkce R.

Tyto funkce dobře poslouží pro aplikace pouze s jedním voláním funkce R a jedním výstupem. Nicméně jiné aplikace mohou potřebovat sofistikovanější interakci s R relací. Proto knihovna implementuje také funkce pro správu relací.

Relační ekvivalent k funkci ocpu.rpc je funkce

```
ocpu.call(fun, [,args] [,callback])
```

Argumenty *fun* a *args* jsou stejné, rozdíl je pouze v argumentu *callback*, tento argument je JSON objekt osahující data vrácená R funkcí. Jde o objekt relace. Objekt relace je třída Javascriptu, která obsahuje ID realce. Z toho poté lze asynchronně získat data, grafy, soubory atd. Tento objekt může být také použít k předání vrácené hodnoty R jako argument další funkci, aniž by objekt musel být načítán. V opencpu.js existují 4 typy argumentů: základní JavaScript hodnota / objekt (automaticky převedeny do R přes JSON), objekt

relace (představuje hodnotu R z předchozího volání funkce), soubor a fragment kódu [24]. Seznam a popis dostupných metod nad relačními objekty lze najít tady: `https://www.opencpu.org/jslib.html#lib-session`.

Kapitola 2

Analýza Sage a možnosti jeho použití ve webové aplikaci - Sage Cell Server

2.1 Seznámení se Sage

SageMath (často zkráceně *Sage*) je počítačový algebraický systém (CAS) sloužící hlavně pro matematické výpočty a geometrické experimentování. Název je zkratkou z anglického **S**oftware for **A**lgebra and **G**eometry **E**xperimentation. Tento software je Open Source vydaný pod licencí GNU [28]. Je multiplatformní, dostupný pro Linux, Mac OS X a díky virtualizaci také pro Windows. První verze byla odhalena v únoru 2005 [29]. Zakladatelem a vedoucí osobností celého projektu je William Stein z University of Washington, Seattle, Washington, USA.

Hlavním cílem projektu *Sage* je vytvořit životaschopnou Open Source alternativu k placeným algebraickým systémům Magma, Maple, Mathematica a MATLAB. Narozdíl od těchto systémů má však Sage všechny zdrojové kódy včetně historie úprav volně k dispozici [30]. V Sage developer trac může také kdokoliv nahlásit objevené chyby v systému, nebo se podílet na jeho vývoji. Sage tak kolem sebe od roku 2005 vytvořil silnou uživatelskou komunitu, jenž dále tento systém rozvíjí. Vyvojáři však musí respektovat, že je Sage zveřejněn pod licencí GNU (chtějí-li tak své úpravy a nadstavby systému dále šířit, musí je šířit také pod touto licencí).

Sage používá programovací jazyk Python, ale obsahuje také kompilátory jazyka C/C++ a používá jazyk *Cython*. *Cython* je upravená verze Pythonu, která je před interpretací převedena na kód v jazyce C a posléze kompilátorem jazyka C zkompilována. To je užitečné zejména ke složitějším matematickým výpočtům, protože jazyk *Cython* je výrazně rychlejší, než jazyk Python [31].

Sage je možné používat ve dvou rozhraních - konzolovém, nebo webovém.

Konzolové rozhraní je vhodné ke kratším výpočtům nebo ke spouštění externích skriptů. Webové rozhraní se spouští z konzolového rozhraní příkazem `notebook()`. Zde pak uživatel pracuje s tzv. *worksheety*. Vlastní GUI Sage nemá, je však možné použít online cloud [32]. Zde má uživatel po registraci, která je zdarma, možnost spravovat své projekty, případně je sdílet s jinými uživateli. Toto online rozhraní pro Sage také usnadňuje uživateli práci, protože mu umožňuje vkládání předdefinovaných funkcí a grafů.

Sage poskytuje rozhraní také pro jiné algebraické systémy a umožňuje tak spouštět funkce mnoha systémů z jednoho konzolového rozhraní. Tyto systémy však musí mít uživatel nainstalované. Instalační balíček Sage zahrnuje např. balíčky GAP, PARI, Singular, Maxima a také R. Nezahrnuje však balíčky jako např. Octave, a také placené systémy MAGMA, Maple a Mathematica [33]. Použití těchto systémů z rozhraní Sage, obzvlášť z toho konzolového, ale nemusí být tak komfortní, jak jsou uživatelé zvyklí. Pracuje-li uživatel ve worksheetu webového rozhraní, je možné zde specifikovat jazyk, v kterém má být interpretován. Toto nastavení se ale týká celého worksheetu, není tedy možné dvě jeho různé buňky interpretovat odlišně.

Velkou předností Sage je schopnost vytvářet interaktivní demonstrace. To je možné díky příkazu `interact`, který je součástí Sage od roku 2008. Ten umožňuje vkládat do buňky Sage textová pole, posuvníky, tlačítka a rozbalovací menu. Archiv aplikací vytvořených komunitou Sage Interact je k dispozici online [34].

Integraci *Sage* do webové aplikace je možné provézt díky projektu *Sage Cell Server*, jeho první verze vyšla v lednu 2011 [35].

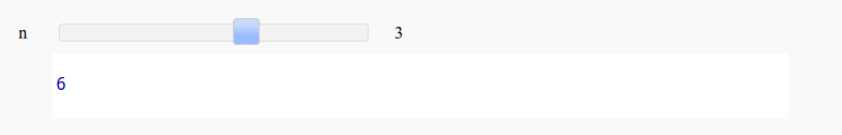
2.2 Statistické výpočty v Sage

Sage používá syntaxi jazyka Python a je tak možné použít všechny jeho dostupné knihovny. Zároveň může Sage využívat knihovny napsané pro jazyk C/C++. Python je interpretovaný jazyk. Díky tomu je jednoduchý pro implementaci a ladění, ale jeho kód může být pomalejší. To je jeden z důvodů, proč Sage využívá ještě vlastní modifikaci jazyka Python, *Cython* [36]. Cython obsahuje oproti jazyku Python některá rozšíření a umožňuje tak v Sage vysokoúrovňové, objektově orientované, funkcionální a dynamické programování. Kód tohoto jazyka může vypadat jako kód jazyka Python, je ale nejprve přeložen na optimalizovaný kód jazyka C/C++ a poté zkompilován. Díky tomu je jazyk Cython dobře znám vyvojářům v Pythonu, ale má potenciál být mnohem rychlejší.

Příkaz `interact` nám v aplikaci umožňuje využít interaktivní prvky pro nastavení proměnných uživatelem a následné překreslování výsledků. Pro použití příkazu je nutné vytvořit třídu, které v parametrech konstruktoru předáme informace o proměnných, jejichž hodnoty se budou měnit pomocí ovládacích prvků. Při každé změně hodnoty se vykoná celý kód za příkazem interact.

Jednoduše lze použít příkaz interact například pro výpočet faktoriálu, kde uživatel volí hodnotu x z množiny $\{0, 1, 2, 3, 4, 5\}$ (obrázek 2.1). Kód pro použití příkazu interact pro výpočet $f(x) = x!$, kde $x \in \{0, 1, 2, 3, 4, 5\}$ vypadá následovně:

```
@interact
def factorial(n=(0..5)):
   return 1 if n == 0 else n * factorial(n - 1)
```



Obrázek 2.1: Jednoduché použití příkazu `interact` pro výpočet faktoriálu.

Čistý jazyk Python, podobně jako jiné programovací jazyky, obsahuje základní matematické funkce a operátory. Pro složitější výpočty v Pythonu je pak nutné zvolit některé z jeho knihoven. Python nabízí mnoho knihoven poskytující funkce vhodné pro statistické výpočty:

random Pseudo-náhodné číselné generátory pro různá rozdělení [37].

statistics Obsahuje funkce pro statistické výpočty nad reálnými číselnými daty. Jde však pouze o několik základních funkcí jako výberový průměr a rozptyl [38].

Numpy, Matplotlib, SciPy, IPython, pandas Tyto nástroje jsou součástí tzv. *SciPy Stack* kolekce open source softwarů pro vědecké výpočty v Pythonu [39]. *Numpy* je základní balíček pro numerické výpočty. Definuje numerické pole, maticové typy a základní operace mezi nimi [40]. *Matplotlib* je balíček pro vykreslovaní jak 2D, tak 3D grafů [41]. Knihovna *SciPy* je sbírka numerických algoritmů a doménově specifických nástrojů, včetně nástrojů pro optimalizaci, statistiku a mnoho dalšího [39]. Interaktivní rozhraní *IPython* umožňuje uživateli zpracovávat data a testovat hypotézy [42]. *Pandas* poskytuje vysoce výkonné a snadno použitelné datové struktury [43].

rpy2 *Rpy* poskytuje nízkoúrovňové rozhraní pro R v Pythonu. *Rpy2* je jeho přepracovaná nová verze. Poskytuje rozhraní vyšší úrovně, je tak možné volat funkce z grafických balíčků a používat struktury a funkce R [44].

python-statlib Balíček *python-statlib* [45] kombinuje tři statistické moduly pro Python: *stats* (kolekce základních statistických funkcí pro Python [46]), *pstats* (poskytuje užitečné nástroje pro manipulaci s listy a polemi

[47]) a *matfunc* (modul pro matematické operace nad vektory, tabulkami a maticemi [48]).

statsmodels Modul umožňující uživateli zkoumat data, odhadovat statistické modely a provádět statistické testy [49].

PyMC Model realizující Bayesovské statistické modely a stochastické metody používající pseudonáhodná čísla (např. Monte Carlo). Je flexibilní a rozšiřitelný, díky čemuž je použitelný pro velkou sadu problémů [50].

PyMix Knihovna poskytující funkce pro aplikace zpracovávající data. Poskytuje konečné modely pro diskrétní i spojité funkce, širokou škálu rozdělení (normální, exponenciální, diskrétní,...), Bayesovské modely, parametrové odhady a další [51].

Druhou možností, jak v Sage provádět statistické výpočty, je zvolit jiný programovací jazyk, který snadno umožňuje rozsáhlou škálu statistických výpočtů a který Sage umí interpretovat. Ideální je vybrat jazyk, který je k dispozici již se základní instalací Sage, tedy jazyk R. Implementace statistických výpočtů se tak pro uživatele zjednodušuje, ale zároveň přichází o výhodu Sage, příkaz `interact`. V případě použití ve webové aplikaci však lze vlastnosti interaktivních Sage aplikací snadno nahradit použitím JavaScriptu.

2.3 Sage Cell Server

Sage Cell Server je open-source webové rozhraní pro *Sage*. Kromě toho také může být použit k vložení *Sage* do libovolné webové stránky. Jedna vložená buňka Sage Cell Server pak odpovídá jedné buňce ve webovém rozhraní.

K samotným výpočtům můžeme využít vlastní (zdrojové kódy k instalaci jsou k dispozici na GitHub [52]), nebo veřejný server poskytovaný tímto projektem.

Vložení *Sage* do stránky pak probíhá jednoduše ve dvou krocích:

1. Do hlavičky HTML stránky uživatel vloží následující kód:

```
<script src="https://sagecell.sagemath.org/static/
jquery.min.js">
</script>
<script src="https://sagecell.sagemath.org/static/
embedded_sagecell.js">
</script>
<script>
sagecell.makeSagecell("inputLocation": ".sage");
</script>
```

Adresa sagecell.sagemath.org může být případně nahrazena adresou serveru uživatele, na kterém je nainstalována služba *SageCell*. První dva tagy <skript> slouží k načtení Javascriptové knihovny jquery.js a skriptu *SageCell* do webové stránky, třetí pak k převedení všech HTML elementů s třídou *sage* na buňku *SageCell*.

Pokud uživatel do stránky vkládá mnoho stylů (jde-li například o blog nebo deck.js prezentaci), může nastat konflikt mezi stylem stránky a stylem *SageCell*, pak je možné pod tyto tagy připojit ještě jeden, který načte speciální styly pro *SageCell* objekty:

```
<link rel="stylesheet" type="text/css"
href="https://sagecell.sagemath.org/static
/sagecell_embed.css">
```

2. Poté je třeba vložit kód do těla stránky. Kód je obalen do tagů <script>, díky čemuž není interpretován jako kód HTML.

```
<div class="sage">
  <script type="text/x-sage">kód_Sage</script>
</div>
```

Při vkládání *Sage* modulů lze upravit nastavení buněk. Je možné nastavit v jakém jazyce budou buňky interpretovány. Uživatel může upravit popis potvrzovacího tlačítka, nebo ho úplně skrýt, stejně jako další části modulu.

Kapitola 3

Vybrané statistické metody

Matematická statistika jako vědecká disciplína zkoumá, zpracovává a vyhodnocuje data. Provádí experimenty, jejichž cílem je zjistit data o dané populaci (soubor jakýchkoliv prvků, které budeme zkoumat). Narozdíl od teorie pravděpodobnosti, která na základě znalostí chování určité náhodné veličiny určuje pravděpodobnost určitého výsledku, matematická statistika na základě dat hledá vlastnosti náhodné veličiny. V testovací aplikaci data dostaneme zadána uživatelem. Aplikace pak tato data analyzuje. V této kapitole se proto budeme zabývat především analýzou náhodných veličin. Definice a věty jsou čerpány ze skript pana Jiřího Anděla [53], ze skript Aplikovaná statistika pana Jiřího Pavlíka [54] a z knihy Graphical methods for data analysis Johna M. Chamberse [55].

3.1 Základní pojmy teorie pravděpodobnosti

Pokud není výsledek nějakého pokusu nebo děje jednoznačně určen podmínkami, za nichž se odehrává, můžeme různě možné výsledky považovat za *elementární jevy*. Ty budeme v tomto textu značit ω, množinu všech elemntárních jevů, nazývaný také *prostor elementárních jevů*, pak Ω. Nechť je na prostoru Ω dána nějaká σ-algebra $\mathcal{A}$ jeho podmnožin. Tyto podmnožiny se nazývají *náhodné jevy*. Právě jednotlivým množinám patřícím do $\mathcal{A}$ se pak připisuje pravděpodobnost pomocí nějaké pravděpodobnostní míry $\mathtt{P}$. Trojice $(\Omega, \mathcal{A}, \mathtt{P})$ se nazývá *pravděpodobnostní prostor*.

Nechť $\mathbb{R}$ je reálná přímka a $\mathcal{B}$ systém borelovských podmnožin. Nechť $X(\omega)$ je měřitelná funkce z $(\Omega, \mathcal{A}, \mathtt{P})$ do $(\mathbb{R}, \mathcal{B})$. Pak se $X(\omega)$ nazývá *náhodná veličina* a značí se X. Každé borelovské množině $B \in \mathcal{B}$ lze přiřadit její vzor $X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\}$ a pravděpodobnostní míru $Q(B) = \mathtt{P}X^{-1}(B)$ [53]. *Indukovanou míru* Q nebo také *zákon rozdělení náhodné veličiny* X budeme obvykle nazývat stručně *rozdělení* X.

Položíme-li $B = (-\infty, x\rangle$, dostaneme

$$F(x) = \mathtt{P}(X \leq x).$$

Této funkci F říkáme *distribuční funkce* a její hodnoty jednoznačně určují rozdělení X.

Distribuční funkce F náhodné veličiny má následující vlastnosti: je neklesající, zleva spojitá a platí, že $\lim_{x\to-\infty} F(x) = 0$ a $\lim_{x\to\infty} F(x) = 1$. Tato funkce může mít nejvýše spočetně mnoho bodů nespojitosti.

V teorii pravděpodobnosti mluvíme o dvou druzích distribuční funkce:

(a) Je-li distribuční funkce F po částech konstantní, pak jde o *diskrétní rozdělení*.

(b) Existuje-li taková funkce $f(x)$, že platí

$$F(x) = \int_{-\infty}^{x} f(t)dt,$$

pak mluvíme o *spojitém rozdělení*. Funkce $F(x)$ se v takovém případě nazývá hustota pravděpodobnosti náhodné veličiny X.

Pro integrály náhodných veličin se ve statistice používá symbol $\mathtt{E}$ (expectation) [53]. $\mathtt{E}X = \int_R x dQ(x)$, kde Q je rozdělení náhodné veličiny X. Pro diskrétní náhodnou veličinu X dostáváme $\mathtt{E}X = \sum_i x_i \mathtt{P}(X = x_i)$. Pro spojitou náhodnou veličinu X s hustotou pravděpodobnosti $f(x)$ dostáváme $\mathtt{E}X = \int_R x f(x)dx$. $X(\omega)d\mathtt{P}(\omega)$ je *střední hodnota*, pokud tento integrál existuje. Obvykle se provádí konkrétní výpočet $\mathtt{E}X = \int \mathbb{R} x dF(x)$, protože většinou známe pouze distribuční funkci. Pokud a je nějaká konstanta, pak $\mathtt{E}a = a$.

Označme $\mu'_k = \mathtt{E}X^k, k = 1, 2, ...$ Číslo μ'_k se nazývá *obecný moment* k-tého řádu. Existuje-li moment μ'_1 a je-li konečný, pak definujeme $\mu_k = \mathtt{E}(X - \mathtt{E}X)^k, k = 0, 1, ...$ Číslo μ_k se nazývá *centrální moment* k-tého řádu [53]. Nejdůležitější centrální moment je μ_2, který se nazývá *rozptyl* a obvykle se značí σ^2, někdy ho ale budeme značit $\operatorname{var} X$. *Směrodatná odchylka* σ je definována $\sigma = \sqrt{\operatorname{var} X}$ a platí, že $\sigma \geq 0$. Po úpravě dostáváme, že $\sigma^2 = \mathtt{E}X^2 - (\mathtt{E}X)^2$. Dále platí, že pokud jsou a a b reálná čísla a existuje-li $\mathtt{E}X$, pak $\mathtt{E}(a+bX) = a+b\mathtt{E}X$. Pokud $\mathtt{E}X^2 < \infty$, pak $\operatorname{var}(a + bX) = b^2 \operatorname{var} X$. Pro náhodné veličiny X a Y s konečnou střední hodnotou platí $\mathtt{E}(aX + bY) = a\mathtt{E}X + b\mathtt{E}Y$. Kromě střední hodnoty jsou důležitými charakteristikami *kvantily*. 100p%-ním kvantilem spojité náhodné veličiny X (přičemž pro p platí, že $0 < p < 1$) nazveme číslo u_p takové, že

$$\mathtt{P}(X \leq u_p) = p \text{ neboli } F(u_p) = p,$$

kde $F(x)$ je distribuční funkce veličiny X. 50% kvantil nazýváme *medián*.

3.1.1 Kovariance a korelace

Kromě číselných charakteristik jednotlivých náhodných veličin X a Y jsou důležité číselné charakteristiky, které vyjadřují jejich vzájemnou souvislost.

Mějme náhodné veličiny X a Y s konečnými druhými momenty. Pak definujeme *kovarianci* náhodných veličin X a Y vztahem

$$\operatorname{cov}(X,Y) = \mathtt{E}\left[(X-\mathtt{E}X)(Y-\mathtt{E}Y)\right].$$

A (Pearsonův) *korelační koeficient*

$$\varrho(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var} X}\sqrt{\operatorname{var} Y}}$$

3.2 Vybraná spojitá rozdělení

Zde uvedeme příklady normálního, exponenciálního a rovnoměrného rozdělení spolu s jejich základními charakteristikami - střední hodnotou a rozptylem.

3.2.1 Normální rozdělení

Nechť $\mu \in \mathbb{R}$ a $\sigma > 0$ jsou dané konstanty (parametry). *Normální rozdělení* (nebo také *Gaussovo*) je určeno hustotou

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ pro } x \in (-\infty, +\infty).$$

a označuje se symbolem $N \sim (\mu, \sigma^2)$.

Střední hodnota $\mathtt{E}X$ a rozptyl $\operatorname{var} X$ náhodné veličiny s normálním rozdělením jsou rovna:

$$\mathtt{E}X = \mu,$$
$$\operatorname{var} X = \sigma^2.$$

3.2.2 Exponenciální rozdělení

Nechť $\lambda > 0$. *Exponenciální rozdělení* $Exp(\lambda)$ má hustotu

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{pro } x > 0, \\ 0 & \text{jinde.} \end{cases}$$

Střední hodnota a rozptyl náhodné veličiny s exponenciálním rozdělením:

$$\mathtt{E}X = \frac{1}{\lambda},$$

$$\operatorname{var} X = \frac{1}{\lambda^2}.$$

3.2.3 Rovnoměrné rozdělení

Nechť (a, b) je konečný nedegenerovaný interval, kde $a < b, a, b \in \mathbb{R}$. *Rovnoměrné rozdělení* $U(a, b)$ má hustotu

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{pro } x \in \langle a, b \rangle, \\ 0 & \text{jinde.} \end{cases}$$

a platí pro něj

$$\mathtt{E}X = \frac{a+b}{2},$$

$$\operatorname{var} X = \frac{(b-a)^2}{12}.$$

3.3 Bodové odhady

Nechť $X_1, ..., X_n$ je posloupnost nezávislých stejně rozdělených náhodných veličin s rozdělením Q. Pak říkáme, že $X_1, ..., X_n$ je *náhodný výběr* z rozdělení Q. Číslo n se nazývá *rozsah výběru*.

Nechť $\mathbf{X} = (X_1, \ldots, X_n)$ má hustotu $f(x, \boldsymbol{\theta})$, kde $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$ je neznámý m-rozměrný parametr. Na základě vektoru $\mathbf{X}$ je třeba získat pokud možno co nejlepší odhad parametru $\boldsymbol{\theta}$, o kterém je předem známo pouze tolik, že patří do nějakého parametrického prostoru $\Omega \subset \mathbb{R}^m$. Jde-li nám o *bodový odhad*, hledáme nějaké měřitelné zobrazení $g : (\mathbb{R}^n, \mathcal{B}) \to (\mathbb{R}^m, \mathcal{B}_m)$ tak, aby náhodný vektor $\hat{\boldsymbol{\theta}} = g(\mathbf{X})$ v nějakém rozumném smyslu co nejlépe aproximoval hodnotu $\boldsymbol{\theta}$.

3.3.1 Vlastnosti odhadu

Řekneme, že odhad $\hat{\boldsymbol{\theta}}$ parametru $\boldsymbol{\theta}$ je *nestranný*, platí-li $\mathtt{E}\mathbf{T} = \boldsymbol{\theta}$ pro každé $\boldsymbol{\theta} \in \Omega$. Platí-li $\mathtt{E}\mathbf{T} = \boldsymbol{\theta} + \mathbf{b}(\boldsymbol{\theta})$, kde funkce $\mathbf{b}$ není identicky rovna nule na množině Ω, nazývá se odhad $\hat{\boldsymbol{\theta}}$ *vychýlený*. Vektoru $\mathbf{b}(\boldsymbol{\theta})$ se pak říká *vychýlení odhadu* $\mathbf{T}$ v bodě $\boldsymbol{\theta}$. Obvykle se snažíme, aby byl odhad také nestranný a konzistentní. Pro definici těchto pojmů odkazujeme čtenáře na [53, p.101-102].

3.3.2 Střední hodnota, rozptyl, směrodatná odchylka a medián

Mezi nejužívanější bodové odhady patří odhady střední hodnoty, rozptylu a směrodatné odchylky.

Položme

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \text{a} \quad S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2.$$

Veličina $\bar{X}$ se nazývá *výběrový průměr* (bodový odhad střední hodnoty) a veličina S^2 *výběrový rozptyl* (bodový odhad rozptylu). Hodnotu $\sqrt{S^2}$ nazýváme *výběrová směrodatná odchylka* (bodový odhad směrodatné odchylky). Základním bodovým odhadem *mediánu* $\tilde{x}$ pro náhodný výběr pro liché n: $x_{((n+1)/2)}$, pro sudé n pak: $(x_{(n/2)} + x_{((n/2)+1)})/2$.

3.3.3 Metoda maximální věrohodnosti

Jednou z metod získávání velmi dobrých odhadů je *metoda maximální verohodnosti*. Jejím cílem je maximalizovat věrohodnostní funkci pro dané naměřené hodnoty.

Mějme náhodný výběr $X_1, ..., X_n$ jehož sdružené rozdělení je určeno pomocí:

$$f(\boldsymbol{x}, \boldsymbol{\theta}(x_i)) = \prod_{i=1}^{n} f(x_i) \text{ pro spojité rozdělení, resp.}$$

$$p(\boldsymbol{x}, \boldsymbol{\theta}(x_i)) = \prod_{i=1}^{n} \mathsf{P}(X_i = x_i) \text{ pro diskrétní rozdělení,}$$

kde $\boldsymbol{\theta} \in \Omega$. Při pevné hodnotě $\boldsymbol{x}$ se funkce $f(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$, resp. $p(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$ jakožto funkce $\boldsymbol{\theta}$ nazývá *věrohodnostní funkce* a značí se $L(\theta, \boldsymbol{x})$.

Hodnota $\hat{\boldsymbol{\theta}}$ parametru $\boldsymbol{\theta}$, která maximalizuje věrohodnostní funkci $f(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$, resp. $p(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$ pro dané $\boldsymbol{x}$, se nazývá *maximálně věrohodný odhad* (MLE) parametru $\boldsymbol{\theta}$.

Funkce $l(\boldsymbol{\theta}, \boldsymbol{x}) = \ln f(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$, resp. $l(\boldsymbol{\theta}, \boldsymbol{x}) = \ln p(\boldsymbol{x}, \boldsymbol{\theta}(x_i))$ se jakožto funkce proměnné $\boldsymbol{\theta}$ při pevném $\boldsymbol{x}$ nazývá *logaritmická věrohodnostní funkce*. Často se vyplatí maximalizovat funkci $l(\boldsymbol{\theta}, \boldsymbol{x})$, protože tak v rovnici dostaneme sumu za produkt. Přičemž maxima obou funkcí jsou stejná.

Nyní provedeme odhad parametrů spojitých rozdělení normálního, exponenciálního a rovnoměrného, pomocí metody maximální věrohodnosti.

3.3.3.1 Normální rozdělení

Nechť $X_1, \ldots, X_n$ je náhodný výběr z normálního rozdělení $X \sim N(\mu, \sigma^2)$

V tomto případě je výhodné maximalizovat logaritmickou věrohodnostní funkci

$$\begin{aligned} l(\mu, \sigma^2, \boldsymbol{x}) &= \ln(L(\mu, \sigma^2, \boldsymbol{x})) \\ &= \ln((2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2})). \end{aligned}$$

O maximu můžeme říct, že pro něj platí

$$\frac{\partial}{\partial \mu} l(\mu, \sigma^2, \boldsymbol{x}) = 0,$$

$$\frac{\partial}{\partial \sigma^2} l(\mu, \sigma^2, \boldsymbol{x}) = 0.$$

Parciální derivaci podle střední hodnoty upravujeme

$$\begin{aligned}
&\frac{\partial}{\partial \mu} l(\mu, \sigma^2, \boldsymbol{x}) \\
&\qquad = \frac{\partial}{\partial \mu}(-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j - \mu)^2) \\
&\qquad = \frac{1}{\sigma^2}\sum_{j=1}^{n}(x_j - \mu) \\
&\qquad = \frac{1}{\sigma^2}(\sum_{j=1}^{n} x_j - n\mu).
\end{aligned}$$

A to je rovno nule jen když platí, že

$$\sum_{j=1}^{n} x_j - n\mu = 0.$$

Z první podmínky pro maximum tedy dostáváme maximálně věrohodný odhad $\hat{\mu}$

$$\hat{\mu} = \frac{1}{n}\sum_{j=1}^{n} x_j = \bar{X}.$$

Parciální derivace podle rozptylu

$$\begin{aligned}
&\frac{\partial}{\partial\sigma^2} l(\mu, \sigma^2, \boldsymbol{x}) \\
&\qquad = \frac{\partial}{\partial\sigma^2}(-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2) \\
&\qquad = -\frac{n}{2\sigma^2} - \left[\frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^2\right]\frac{d}{d\sigma^2}\left(\frac{1}{\sigma^2}\right) \\
&\qquad = -\frac{n}{2\sigma^2} - \left[\frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^2\right]\left(-\frac{1}{(\sigma^2)^2}\right) \\
&\qquad = -\frac{n}{2\sigma^2} + \left[\frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^2\right]\frac{1}{(\sigma^2)^2} \\
&\qquad = \frac{1}{2\sigma^2}\left[\frac{1}{\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2 - n\right]\frac{1}{(\sigma^2)^2}.
\end{aligned}$$

Musí platit, že $\sigma^2 \neq 0$, pak se tento výraz rovná nule jen když

$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}(x_j-\mu)^2.$$

Dostáváme tedy maximálně věrohodný odhad $\hat{\sigma^2}$

$$\hat{\sigma^2} = \frac{1}{n}\sum_{j=1}^{n}(x_j-\hat{\mu})^2.$$

3.3.3.2 Exponenciální rozdělení

Nechť $X_1, \ldots, X_n$ je náhodný výběr z exponenciálního rozdělení $X \sim Exp(\lambda)$

V tomto případě je opět výhodnější maximalizovat logaritmickou věrohodnostní funkci

$$\begin{aligned}
l(\lambda, \boldsymbol{x}) &= \ln(L(\lambda, \boldsymbol{x})) \\
&= \ln(\lambda^n e^{-\lambda\sum_{j=1}^{n} x_j})
\end{aligned}$$

Pro maximum platí

$$\frac{d}{d\lambda} l(\lambda, \boldsymbol{x}) = 0.$$

Zderivováním logaritmické věrohodnostní funkce získáváme

$$\frac{d}{d\lambda} l(\lambda, \boldsymbol{x}) = \frac{d}{d\lambda} l(n \ln(\lambda) - \lambda \sum_{j=1}^{n} x_j) = \frac{n}{\lambda} - \sum_{j=1}^{n} x_j.$$

Položíme rovné nule a dostáváme maximálně věrohodný odhad parametru λ

$$\hat{\lambda} = \frac{n}{\sum_{j=1}^{n} x_j} = \frac{1}{\bar{X}}.$$

3.3.3.3 Rovnoměrné rozdělení

Nechť $X_1, \ldots, X_n$ je náhodný výběr z rovnoměrného rozdělení $X \sim U(a, b)$
Upravíme věrohodnostní funkci

$$\begin{aligned} L(a, b, \boldsymbol{x}) &= \prod_{j=1}^{n} f(x_j) \\ &= \prod_{j=1}^{n} \frac{1}{b-a} \chi_{\langle a,b \rangle}(x) \\ &= \frac{1}{(b-a)^n} \prod_{j=1}^{n} \chi_{\langle a,b \rangle}(x) \end{aligned}$$

kde $\chi_{\langle a,b \rangle}(x)$ je charakteristická funkce množiny hodnot z intervalu $\langle a, b \rangle$. Celý výraz $\prod_{j=1}^{n} \chi_{\langle a,b \rangle}(x)$ může nabývat pouze hodnoty 0 nebo 1. Hodnoty 1 nabyde v případě, že pro všechna x_j z $x_1, ..., x_n$ platí, že jsou z intervalu $\langle a, b \rangle$. V případě, že $\exists x_j \notin \langle a, b \rangle$, pak celý výraz nabývá hodnoty 0. Funkce nabývá vždy nezáporných hodnot, protože $a < b$, proto chceme-li najít maximum, musí výraz $\prod_{j=1}^{n} \chi_{\langle a,b \rangle}(x) = 1$, z čehož plyne, že $a \leq \min x_{j_j} < \max x_{j_j} \leq b$. Nyní tedy hledáme maximum výrazu $\frac{1}{(b-a)^n}$ takové, aby platil předchozí interval. Aby tento zlomek nabýval co nejvyšších hodnot, musí jmenovatel být co nejmenší, pak tedy musí být rozdíl $b - a$ co nejmenší, zároveň však musí být $a \leq \min x_{j_j}$ a $b \geq \max x_{j_j}$. Z toho plyne, že maximálně věrohodný odhad parametrů a a b je

$$\hat{a} = \min X, \hat{b} = \max X$$

3.4 Odhady tvaru rozdělení

3.4.1 Histogram

Histogram je graf znázorňující, jak jsou četnosti rozděleny mezi zvolené intervaly hodnot zkoumané veličiny. Graf je rozdělen do tříd (intervalů), každé třídě

vždy odpovídá jeden sloupec grafu. Šířka sloupce odpovídá velikosti intervalu a jeho výška četnosti hodnot v tomto intervalu. Díky histogramu můžeme odhadovat tvar rozdělení. Tvar rozdělení můžeme dobře odhadnout také pomocí grafu empirické distribuční funkce (CDF).

3.4.2 Empirická distribuční funkce

Nechť $X_1, ..., Xn$ je náhodný výběr z rozdělení, které má distribuční funkci F. Budiž x dané reálné číslo. Zaveďme náhodné veličiny

$$\xi_i(x) = \begin{cases} 1 & \text{, je-li } X_i < x, \\ 0 & \text{, je-li } X_i \geq x. \end{cases}$$

pro $i = 1, ..., m$. *Empirická distribuční funkce* je pro každé $x \in \mathbb{R}$ definována vztahem

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \xi_i(x).$$

3.4.3 Quantile-Quantile plot

Quantile-Quantile plot (*Q-Q plot*) je jednou z velmi účinných vizuálních metod k porovnání dvou různých rozdělení. Je zkonstruován vynesením kvantilů jednoho empirického rozdělení do grafu proti odpovídajícím kvantilům druhého rozdělení [55, p.193-203]. Označíme-li tyto dvě sady dat x_i, pro $i = 1..n$ a y_j pro $j = 1..m$, pak empirický Q-Q plot je graf $Q_y(p)$ proti $Q_x(p)$ pro rozsah p hodnot.

Pokud jsou tato dvě rozdělení stejná, body leží na přímce $y = x$, odchylky od této funkce nám pak dávají detailní informace o tom, jak se tato rozdělení liší.

3.4.4 Scatter plot

Scatter plot je graf znázorňující závislost obvykle dvou veličin. Data jsou v grafu znázorněna jako množina bodů, jejichž umístění na svislé ose udává hodnota první proměnné a umístění na ose vodorovné hodnota druhé proměnné (případně na ose kolmé k těmto dvěma osám hodnota třetí proměnné). Pomocí korelačního diagramu je možné jednoduše vizuálně porovnat vzájemný vztah mezi těmito proměnnými bod po bodu [55, p.82-87]. Pro lepší názornost je vhodné nastavit na všech osách grafu stejné stupnice.

3.5 Testování hypotéz

Testováním hypotéz testujeme úsudek o vybraných vlastnostech nějakého statistického souboru dat. Nejprve zformulujeme výrok o nějakém parametru

rozdělení tohoto souboru, a pak na základě náhodného výběru testujeme, zda tento výrok (*hypotéza*) platí, či neplatí. Metody, kterými ověřujeme platnost hypotézy nazýváme *testování hypotéz*. Mezi speciální testy patří testy o hodnotách parametru $\boldsymbol{\theta}$ určujícím rozdělení náhodného výběru.

Používáme pojmy *nulová hypotéza*, kterou značíme H_0 a která označuje tvrzení, jehož platnost chceme potvrdit, případně vyvrátit. *Alternativní hypotéza*, kterou značíme H_A je tvrzení opačné k H_0, a často se mu říká pouze zkráceně *alternativa*. Ta může být *oboustranná*, či *jednostranná*. Ukažme si to na příkladu: Chceme ověřit nulovou hypotézu $H_0 : \mu = 10$. Alternativou k této nulové hypotéze může být hypotéza $H_A : \mu \neq 10$, pak jde o *oboustrannou alternativu* (to, že zamítneme nulovou hypotézu nám neříká, zda je střední hodnota ve skutečnosti menší, nebo větší než testovaná hodnota). Jinou alternativou k této nulové hypotéze ale může být také $H_A : \mu < 10$, pak jde o *jednostrannou alternativu*. Testování hypotéz o hodnotách parametru $\boldsymbol{\theta}$ rozdělení můžeme provádět pomocí konfidenčních intervalů.

Má-li soubor z něhož provádíme náhodný výběr rozdělení dané parametrem $\boldsymbol{\theta} \in \mathbb{R}^d$ (ověřované tvrzení se týká parametru $\boldsymbol{\theta}$), pak jde o *parametrický* typ hypotézy. Vybíráme-li z obecného rozdělení (tvrzení se týkají různých vlastností rozdělení, případně tvaru celého rozdělení), pak jde o *neparametrický* druh hypotézy. V případě parametrických testů se nejčastěji testuje nulová hypotéza $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, kde $\boldsymbol{\theta}_0$ je pevná hodnota proti alternativě $H_A : \boldsymbol{\theta} \neq \boldsymbol{\theta}_0$, tzv. oboustranná alternativa,

Může se stát, že hypotézu H_0 zamítneme, ačkoliv je správná. Pak se dopustíme *chyby prvního druhu*. Když naopak tuto hypotézu nezamítneme, ačkoli není správná, dopustíme se *chyby druhého druhu*. Nemůžeme kontrolovat hodnotu obou těchto chyb, proto kontrolujeme, aby chyba prvního druhu byla rovna předem danému číslu α. Toto číslo nazýváme *hladina významnosti testu*. Minimální hladina významnosti, na které lze hypotézu H_0 zamítnout se nazývá *p-hodnota* a záleží na realizaci náhodného výběru. Velikost této hodnoty nám udává sílu zamítnutí, nebo nezamítnutí hypotézy H_0. Čím menší tato hodnota je, tím významnější je zamítnutí H_0.

Parametrické testy můžeme provádět dvěma rozdílnými postupy. První postup je založen na konfidenčních intervalech (intervalech spolehlivosti) a druhý na konstrukci testovací statistiky R a kritickém oboru testu W_α. První postup probíhá následovně: nejprve zvolíme hladinu významnosti α, poté provedeme náhodný výběr, zkonstruujeme interval spolehlivosti následující alternativu H_A a nakonec rozhodneme o platnosti nulové hypotézy. Tu zamítáme, pokud $\boldsymbol{\theta}$ nenáleží intervalu spolehlivosti.

Druhý postup, založený na konstrukci testovací statistiky R a kritickém oboru testu W_α probíhá následujícím způsobem: opět nejprve zvolíme hladinu významnosti α, provedeme náhodný výběr, poté vypočteme testovací statistiku R a pokud $R \in W_\alpha$, pak zamítáme H_0. Kritický obor W_α lze často převézt na interval spolehlivosti, který jsme používali v prvním postupu.

Testovací statistika je funkcí náhodného výběru $X_1, ..., X_n$ a budeme ji

obecně značit R. Je to tedy náhodná veličina. Obor všech hodnot této statistiky, pro které se nulová hypotéza H_0 zamítá, se nazývá *kritický obor testu* hypotézy H_0 a budeme ho značit W_α. Testovací statistika záleží na testovaném parametru, kritický obor testu pak na této testovací statistice a na konkrétní testované hypotéze. V naší testovací aplikaci budeme provádět pouze testy o střední hodnotě a testy dobré shody. Konkrétní hodnoty testovací statistiky a kritického oboru testu pak uvedeme u popisu jednotlivých testů.

3.5.1 T-test o střední hodnotě

T-test je metoda, která umožňuje ověřit, zda normální rozdělení, z něhož pochází určitý náhodný výběr, má určitou konkrétní střední hodnotu, přičemž rozptyl je neznámý, nebo zda dvě normální rozdělení mající stejný (byť neznámý) rozptyl, z nichž pocházejí dva nezávislé náhodné výběry, mají stejné střední hodnoty, nebo zda je rozdíl těchto středních hodnot roven určitému číslu.

Princip testu spočívá v tom, že pokud náhodný výběr pochází z normálního rozdělení, pak výběrový průměr má také normální rozdělení se stejnou střední hodnotou. Rozdíl výběrového průměru a střední hodnoty normovaný pomocí skutečného rozptylu by pak měl normální rozdělení s nulovou střední hodnotou a jednotkovým rozptylem. Skutečný rozptyl ale neznáme, pokud jej však nahradíme odhadem pomocí výběrového rozptylu, dostaneme studentovo (t) rozdělení, které je podobné normálnímu rozdělení.

Máme tři druhy t-testů, jednovýběrový, dvouvýběrový a párový.

Jednovýběrovým t-testem o střední hodnotě testujeme hodnotu střední hodnoty za předpokladu, že neznáme skutečný rozptyl σ^2. Testovací statistika R pro tento případ je rovna

$$R = \frac{\bar{X} - \mu_0}{S}\sqrt{n}$$

a při platnosti hypotézy $H_0 : \mu = \mu_0$ má Studentovo t-rozdělení s (n-1) stupni volnosti. Konkrétní hodnoty tohoto rozdělení nalezneme ve statistických tabulkách. V následující tabulce pak nalezneme hodnoty kritického oboru W_α, jak pro testy s jednostrannou, tak oboustrannou alternativou s hladinou významnosti α.

Tabulka 3.1: Jednovýběrový t-test o střední hodnotě [54, tab. 9.3.1]

H_0	H_1	Testovací statistika R	Kritický obor W_α
$\mu = \mu_0$	$\mu \neq \mu_0$	$R = \frac{\bar{X}-\mu_0}{S}\sqrt{n}$	$\lvert R\rvert = t_\alpha(n-1)$
$\mu \leq \mu_0$	$\mu > \mu_0$		$R > t_{2\alpha}(n-1)$
$\mu \geq \mu_0$	$\mu < \mu_0$		$R < -t_{2\alpha}(n-1)$

Dvouvýběrovým t-testem rovnosti středních hodnot testujeme hypotézy o vztahu středních hodnot μ_1, μ_2 pomocí dvou náhodných výběrů s normál-

ním rozdělením. Tyto výběry mají obecně různé rozsahy n_1,n_2, přitom ale předpokládáme, že tyto dva náhodné výběry jsou na sobě nezávislé. Zpravidla nevíme, zda rozptyly rozdělení dvou porovnávaných souborů jsou nebo nejsou stejné. Pak je nutné před vlastním testem rovnosti středním hodnot provést předběžný oboustranný Fisherův F-test hypotézy rovnosti rozptylů (viz kapitola 9.3.3 [54]). Zamítne-li se předběžná hypotéza rovnosti rozptylů σ_1^2, σ_2^2, musíme použít test pro $\sigma_1^2 \neq \sigma_2^2$, nezamítne-li se můžeme použít test pro $\sigma_1^2 = \sigma_2^2$. Oba jsou popsány v následujících tabulkách 3.2 a 3.3.

Tabulka 3.2: Dvouvýběrový t-test rovnosti středních hodnot pro $\sigma_1^2 = \sigma_2^2$ [54, tab. 9.3.5]

H_0	H_1	Testovací statistika R	Kritický obor W_α
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$R = \frac{\bar{X}_1 - \bar{X}_2}{S_{12}} \sqrt{\frac{n_1 n_2}{n_1 + n_2}}$	$\lvert R \rvert > t_\alpha(n_1 + n_2 - 2)$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$R > t_{2\alpha}(n_1 + n_2 - 2)$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$R < -t_{2\alpha}(n_1 + n_2 - 2)$

Tabulka 3.3: Dvouvýběrový t-test rovnosti středních hodnot pro $\sigma_1^2 \neq \sigma_2^2$ [54, tab. 9.3.6]

H_0	H_1	Testovací statistika R	Kritický obor W_α
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$R = \frac{\bar{X}_1 - \bar{X}_2}{S_d}$	$\lvert R \rvert > t_\alpha(n_d)$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$R > t_{2\alpha}(n_d)$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$R < -t_{2\alpha}(n_d)$

Kde

$$S_{12} = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}.$$

$$S_d = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}, n_d = \frac{S_d^4}{\frac{1}{n_1 - 1}\left(\frac{S_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{S_2^2}{n_2}\right)^2}.$$

Párový t-test rovnosti středních hodnot dvou náhodných veličin se obecně používá v situaci, kdy máme základní soubor s dvěma naměřenými veličinami X_1 a X_2, případně dva „spárované“ základní soubory, v nichž v každém z nich máme naměřenu jednu z těchto veličin, s nějakým dvourozměrným rozdělením pravděpodobnosti (budeme předpokládat, že jde o dvourozměrné normální rozdělení) s neznámými středními hodnotami μ_1, μ_2 a rozptyly σ_1^2, σ_2^2. Budeme testovat hypotézy o vztahu mezi středními hodnotami μ_1, μ_2 těchto veličin X_1 a X_2 na základě dvourozměrného náhodného výběru (dvojic hodnot) (X_{11},X_{21}),(X_{12},X_{22}),...,(X_{1n},X_{2n}).

Mějme náhodnou veličinu $Z = X_1 - X_2$. Z předpokladu mormality rozdělení veličin X_1,X_2 plyne, že $Z \sim N(\mu, \sigma^2)$, kde zřejmě $\mu = \mu_1 - \mu_2$, ale $\sigma^2 \neq \sigma_1^2 + \sigma_2^2$, protože veličiny X_1 a X_2 nejsou nezávislé.

Tabulka 3.4: Párový t-test rovnosti středních hodnot [54, tab. 9.3.7]

H_0	H_1	Testovací statistika R	Kritický obor W_α
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$R = \frac{\bar{Z}}{S_Z}\sqrt{n}$	$\lvert R\rvert > t_\alpha(n-1)$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$R > t_{2\alpha}(n-1)$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$R < -t_{2\alpha}(n-1)$

Směrodatná odchylka veličiny S_Z je dána

$$S_Z = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \bar{Z})^2}.$$

3.5.2 Kolmogorov-Smirnovův test

Testy dobré shody narozdíl od testů o parametrech rozdělení umožňují testovat hypotézy o celém tvaru tohoto rozdělení. Tak je možné rozhodnout, z jakého rozdělení je či není námi provedený náhodný výběr. *Kolmogorov-Smirnovův test* je jedním z testů dobré shody rozdělení. Může být jednovýběrový, pak porovnáváme, zda je rozdělení populace rovné nějakému teoretickému rozdělení, nebo dvouvýběrový, pak zjišťujeme, zda dané dvě populace mají shodné rozdělení.

Jednovýběrový Kolmogorov-Smirnovův test dobré shody (test o rozdělení náhodné veličiny) vypadá tak, že máme nulovou hypotézu H_0: rozdělení veličiny X má distribuční funkci $F_0(x)$ a alternativu H_1: rozdělení veličiny X nemá distribuční funkci $F_0(x)$. Mějme nyní náhodný výběr $X_1, ..., X_n$. Další postup spočívá v tom, že porovnáváme zjištěnou empirickou distibuční funkci s hypotetickou distribuční funkcí $F_0(x)$. Nulovou hypotézu H_0 pak zamítáme, je-li

$$R = \sup_x |F_n(x) - F_0(x)| > D_\alpha(n),$$

kde $D_\alpha(n)$ je kritická hodnota pro Kolmogorov-Smirnovův test (a tedy kritická hodnota testovací statistiky R) a tuto hodnotu nalezneme v tabulkách. Dá se ukázat [54, p.134], že vzhledem k definici empirické distribuční funkce lze testovací statistika R spočítat ze vztahu

$$R = \max_{1 \geq i \geq n} \left\{ \max\left(R_i, R_i^-\right) \right\},$$

kde

$$R_i = \left| F_0(x_{(i)}) - \frac{i}{n} \right|, R_i^- = \left| F_0(x_{(i)}) - \frac{i-1}{n} \right|.$$

Dvouvýběrový Kolmogorov-Smirnovův test dobré shody porovnává shodnost rozdělení dvou náhodných veličin. Mějme tedy náhodný výběr $X_1, ..., X_m$ z rozdělení se spojitou distribuční funkcí F a na něm nezávislý náhodný výběr $Y_1, ..., Y_n$ z rozdělení se spojitou distribuční funkcí G. Nyní mame nulovou hypotézu H_0: $F = G$ proti alternativě H_1: $F \neq G$. Označme F_m empirickou distribuční funkci prvního výběru a G_n empirickou distribuční funkci druhého výběru. Podle Glivenkovy věty [53, Věta 11.10] víme, že se funkce F_m a G_n při rostoucích m a n blíží distribučním funkcím F a G. Označme

$$D_{m,n} = \sup_x |F_m(x) - G_n(x)| .$$

Jsou-li čísla m a n malá, porovná se $D_{m,n}$ s přesnými kritickými hodnotami $D_{m,n}(\alpha)$. V případě větších hodnot m a n se využije Smirnovova věta [53, Věta 11.11]:
Označme $M = mn/(m + n)$. Nechť

$$K(\lambda) = 1 - 2\sum_{k=1}^{\infty}(-1)^{k+1}e^{-2k^2\lambda^2}.$$

Pak pro každé $\lambda > 0$ platí

$$\lim_{m,n\to\infty} \mathtt{P}(\sqrt{M}D_{m,n} < \lambda) = K(\lambda).$$

Funkce $K(\lambda)$ z této věty se aproximuje pomocí počátečních členů $1 - 2e^{-2\lambda^2}$. Po použití této aproximace dostáváme

$$\mathtt{P}\left(D_{m,n} < \frac{\lambda}{\sqrt{M}}\right) \doteq 1 - 2e^{-2\lambda^2}.$$

Výraz na pravé straně je roven $1 - \alpha$ pro $\lambda = \lambda_\alpha = \sqrt{\frac{1}{2}\ln\frac{2}{\alpha}}$. Aproximativní kritická hodnota je tedy

$$D^*_{m,n}(\alpha) = \frac{\lambda_\alpha}{\sqrt{M}} = \sqrt{\frac{1}{2M}\ln\frac{2}{\alpha}}.$$

Po použití Smirnovovy věty se tedy v případě větších hodnot m a n položí

$$\lambda_0 = \sqrt{M}D_{m,n}.$$

Nyní se vypočte se hodnota $K(\lambda_0)$. Pokud vyjde $K(\lambda_0) \geq 1 - \alpha$, zamítne se nulová hypotéza H_0 na hladině, která se s rostoucími rozsahy blíží číslu α. Přitom se při větších hodnotách m a n kritická hodnota pro veličinu $D_{m,n}$ obvykle aproximuje číslem $D^*_{m,n}(\alpha)$. Hypotéza H_0 se pak zamítá, když $D_{m,n} \geq D^*_{m,n}(\alpha)$.

Kapitola 4

Tvorba testovací aplikace

K tvorbě testovací webové aplikace bylo použito převážně jazyka PHP, dále pak HTML a Javascriptu. Aplikace také využívá jednoduchou MySQL databázi. Vzhled aplikace byl upraven pomocí CSS a některé navigační prvky byly použity z projektu Bootstrap. Jazyk PHP, který je nyní nejrozšířenějším jazykem webových aplikací [56], a databázový systém MySQL byly zvoleny hlavně kvůli jejich velkému rozšíření a dostupnosti na serverech. Tento jazyk mohl být zvolen také díky tomu, že jeho použití nijak nebrání integraci R, za pomocí OpenCPU API, a Sage, za pomocí Sage Cell Server, do webové aplikace, což vyplynulo z předchozí analýzy.

Pro implementaci samotných statistických výpočtů byly použity jazyky R a Sage. K jejich integraci do webové aplikace bylo pro R použito API poskytované službou OpenCPU, pro Sage služba Sage Cell Server. K lepšímu srovnání obou rozhraní byly stejné výpočty provedené v obou systémech umístěny vždy do jedné webové stránky, kde mezi nimi může uživatel snadno přepínat. Ze stejného důvodu byla snaha udržet jejich vzhled pokud možno stejný.

Na úvodní stránce aplikace má uživatel možnost nahrát data v souboru formátu CSV, a tak vytvořit nový projekt, nebo zvolit některý ze stávajících projektů. Po přechodu na stránku projektu vidí uživatel přehled proměnných, které datový soubor obsahuje. Jsou-li proměnné numerického typu, lze je dále analyzovat. Uživatel má možnost analyzovat buď jednu proměnnou a nebo porovnávat dvě proměnné mezi sebou.

V analýze jedné veličiny je proveden bodový odhad základních charakteristik, jako je výběrový průměr, výběrový rozptyl, výběrová smerodatná odchylka a medián. Dále jsou zde vykresleny grafy - histogram, graf empirické distribuční funkce a Q-Q plot. U histogramu a grafu empirické distribuční funkce má uživatel možnost zvolit teoretické rozdělení k porovnání s náhodnou veličinou. Do grafu je poté vykreslena křivka vybraného rozdělení. V případě Q-Q plotu uživatel také volí teoretické rozdělení, s nímž jsou zadaná data porovnávána. Ve všech případech jsou parametry teoretického rozdělení odhadnuty metodou maximální věrohodnosti (MLE). Aplikace umožňuje porovnávat

data pouze se základními spojitými rozděleními - normálním, exponenciálním a rovnoměrným. Dále může uživatel provádět oboustranný t-test o střední hodnotě μ, u něhož může zvolit hladinu významnosti α tohoto testu. Aplikace mu poté zobrazí výsledky testu. Poslední funkcí pro jednu proměnnou je Kolmogorov-Smirnovův test dobré shody, kde uživatel opět volí teoretické rozdělení k porovnání a opět je zobrazen výsledek.

Na stránce porovnávající dvě veličiny je uživateli opět k dispozici přehled některých základních charakteristik a to výběrový rozptyl obou veličin, kovariance a korelační koeficient. Dále je zde vykreslen Scatter plot a proveden párový t-test o rovnosti středních hodnot a opět Kolmogorov-Smirnonův test o rovnosti rozdělení, tentokrát ovšem dvouvýběrový.

Databáze aplikace obsahuje pouze jednu tabulku *files*, jejímž obsahem jsou informace o jednotlivých projektech a nahraných souborech. Další tabulky jsou tvořeny dynamicky za běhu aplikace. Ty pak obsahují informace o konkrétních proměnných ze souboru.

Aplikace byla navržena tak, že jsou potřebná data uložena v CSV souborech a jednotlivá rozhraní mají k dispozici proměnnou obsahující cestu k příslušnému souboru, či souborům. Soubor nahraný uživatelem je čten pouze při jeho nahrávání na server. V tu chvíli jsou také do databázové tabulky *files* uloženy informace o nově vzniklém projektu a vytvořena nová tabulka pro informace o jednotlivých proměnných v souboru. Při načtení je soubor rozparsován na jednotlivé sloupce, které obsahují vždy jednu veličinu. Každý sloupec je pak uložen do zvláštního pomocného souboru. Cesty právě k těmto pomocným souborům jsou pak posílány jako pamaretry funkcím. Toto řešení je sice poněkud náročnější na paměť, ta je ovšem vzhledem k velikosti CSV souborů a dostupnosti místa na serveru zanedbatelná. Naopak je tak umožněno rychlejší zpracovávání dat.

4.1 OpenCPU

4.1.1 Návrh

Vzhledem k předešlé analýze bylo k integraci R do webové aplikace použito služby OpenCPU a jeho veřejně dostupný server. Dále bylo zjištěno, že je nutné vytvořit balíček R se všemi potřebnými funkcemi pro výpočty a umístit tento balíček do veřejného repozitáře na GitHub.com. Vzhledem k tomu, že je OpenCPU kompatibilní s jakýmkoliv jazykem nebo frameworkem, který umožňuje komunikaci pomocí protokolu HTTP, nebylo třeba v tomto ohledu nijak přizpůsobovat použitý jazyk aplikace. Protože OpenCPU je pouze spojení webu se statickým prostředím R, bylo nutné všechny interaktivní prvky a zpracovávání vstupů uživatele zajistit v jazyce PHP nebo Javascript. V analýze bylo dále zjištěno, že OpenCPU aplikace počítají ze své podstaty paralelně, funkce R proto byly navrženy tak, aby vždy řešili jeden dílčí problém aplikace a ze stránky pak byly požadavky na jejich volání odesílány současně. Uká-

zalo se také, že prostředí R poskytuje ve svých standartních balíčcích všechny funkce potřebné k implementaci výpočtů v testovací aplikaci a nebylo tak nutné používat žádné další.

4.1.2 Integrace

Samotná integrace funkcí R do webové aplikace proběhla v několika krocích:

1. Na základě analýzy byla vytvořena jednoduchá struktura balíčku R obsahující složky R, man a soubory DESCRIPTION a NAMESPACE. Do složky R pak byly vloženy potřebné funkce R a do složky man jejich manuálové stránky. Základní charakteristiky balíčku byly definovány v souboru DESCRIPTION a do souboru NAMESPACE byly vepsány potřebné exporty ke každé funkci použité ve webové aplikaci.

2. Balíček byl poté umístěn do veřejného repozitáře na GitHub.com.

3. Poté proběhlo samotné volání funkcí R pomocí HTTP požadavku POST. Protože je aplikace postavena na jazyce PHP, bylo volání funkcí R provedeno pomocí knihovny PHP cURL (client URL Library). Ta umožňuje spojení s mnoha servery za použití různých protokolů, včetně protokolu HTTP a HTTPS. Volání funkce R pak proběhlo následovně:

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$cesta_k_funkci);
curl_setopt($ch, CURLOPT_POST, $pocet_parametru);
curl_setopt($ch, CURLOPT_POSTFIELDS,
"parametry_funkce_R");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
$vystup = curl_exec($ch);
curl_close($ch);
```

Nejprve byla funkcí `curl_init` inicializována relace a manipulátor relace byl uložen do proměnné $ch. Poté byla provedena nastavení této relace, určena cílová URL, bylo specifikováno, že jde o metodu POST a také počet a obsah jednotlivých parametrů. Poté byl povolen *RETURNTRANSFER*, díky čemuž je výstup volání funkce `curl_exec` návratová hodnota požadavku a protože jde o zabezpečený protokol HTTPS, bylo zakázáno ověřování SSL certifikátu. Po spuštění dílčích připojení byl výstup relace uložen do proměnné $vystup a nakonec bylo spojení ukončeno.

4. Ke zpracování výstupu pak bylo použito id relace, které, kromě jiného, proměnná $vystup obsahuje. To bylo poté použito dvěma způsoby. Buď jako parametr volání jiné funkce R, nebo k zobrazení výsledných grafů či výpočtů v hledaném formátu.

4.1.3 Implementace výpočtů

Pro načítání dat ze souborů byla použita R funkce `read.csv`. K výpočtům základních charakteristik jednotlivých veličin byly použity funkce R: mean(data) - výběrový průměr, var(data) - výběrový rozptyl, median(data) - medián, sd(data) - výběrová směrodatná odchylka, cov(data1,data2) - kovariance, cor(data1, data2) - korelační koeficient. Tyto funkce jsou v balíčku obaleny funkcemi, které pouze řeší přenos dat z webové stránky.

Funkce pro vykreslování grafů byly implementovány následovně:

- **Histogram**

  Funkce přijímá parametry se zadanými daty, šířkou binu danou uživatelem a rozdělění, jehož křivka má být pro srovnání vykreslena do grafu. Poté využívá funkci R `hist(data, breaks, prob=TRUE)`, kde breaks je parametr udávající rozdělení binů a je dán sekvencí čísel podle šířky binu zadané uživatelem. Díky parametru `prob` nejsou do grafu vykreslovány frekvence, nýbrž pravděpodobnosti jednotlivých intervalů. Parametry rozdělení k porovnání jsou odhadnuty metodou maximální věrohodnosti (viz analýza) a hustota jednotlivých rozdělení je poté vygenerována funkcí `dnorm`, `dexp`, `dunif`, kterým jsou dány spočtené parametry rozdělení a zadaná data. Požadovaná křivka pro porovnání je poté do grafu zakreslena funkcí `lines`.

- **Graf empirické distribuční funkce**

  Funkce opět přijímá zadaná data a rozdělení, jejichž křivky mají být pro porovnání do grafu vykresleny. Nejprve je spočtena souhrnná empirická distribuční funkce pomocí funkce R `ecdf(data)` a poté vykreslen graf této funkce pomocí plot(ecdf(data),do.points = FALSE). Podobně jako u histogramu jsou vykresleny křivky teoretických rozdělení k porovnání, v tomto případě však nevykreslujeme hustotu, nýbrž distribuční funkci, proto používáme funkce `pnorm`, `pexp`, `punif`.

- **Q-Q plot** Funkci jsou dána data a teoretické rozdělení, s nímž chce uživatel data porovnat. Opět je spočítáno teoretické rozdělení a graf vykreslen pomocí funkce R `qqplot(data,teoreticke_rozdeleni,plot.it = TRUE)`.

- **Scatter plot**

  Scatter plot je jednoduše vykreslen pomocí R funkce `plot(data1,data2)`.

Testování hypotéz je provedeno pomocí funkce R `t.test`, které jsou předána data, nulová hypotéza a také hladina testu α. Pomocí parametru `alternative` je v případě jedné veličiny odlišeno, že jde o oboustranný (`two.sided`) t-test. V případě t-testu u porovnávání dvou veličin je poté pomocí logického parametru `paired` nastaveno, že jde o párový t-test. U Kolmogorov-Smirnovova testu je u jedné veličiny opět spočteno teoretické rozdělení k porovnání. U dvou veličin jsou pak testovací funkci předána data obou veličin. K testu je použita R funkce `ks.test(data1, data2, exact = TRUE)`. Parametr `exact` udává, zda má být spočtena p-hodnota testu.

4.2 Sage Cell Server

4.2.1 Návrh

K integraci Sage do webové aplikace byla použita služba Sage Cell Server a veřejně dostupný server. Při návrhu implementace statistických výpočtů za pomocí Sage bylo vycházeno z předchozí analýzy. Z té vyplynulo, že lze buňky Sage vložit do libovolné webové stránky, v Sage provádět statistické výpočty v podstatě dvěma odlišnými způsoby. První způsob spočívá v tom, že jsou výsledné buňky interpretovány jako buňky v jazyce Sage, a tedy jsou psány především v jazyce Python. Druhou možností je využít, že služba Sage nabízí také interprety jiných jazyků a zvolit tak pro statistické výpočty vhodnější jazyk R. V rámci návrhu aplikace byly vyzkoušeny obě tyto varianty. Sage má ale zatím problém s podporou grafiky R na některých platformách a veřejný server Sage Cell Server nám tak nedovoluje grafy R vůbec vykreslovat (kvůli absenci systému X11). To byl hlavní důvod, proč byl k implementaci statistických funkcí vybrán jazyk Python. Odliší-li uživatel třídy objektů, v nichž je ve webové stránce umístěn kód Sage k vyhodnocení, je možné vkládat různě interpretované buňky Sage do jedné webové stránky. Tak by bylo možné použít jazyk R alespoň k výpočtům s textovými výstupy. Ukázalo se však, že Sage Cell Server při vyhodnocování buněk R vynechává prázdné řádky podle řádků kódu (viz obrázek 4.1), pro testovací aplikaci tak bylo jeho použití nevhodné.

Obrázek 4.1: Formátování výstupu kódu R v buňce Sage Cell Server.

Použití jazyka Python také umožňuje využít příkaz `interact`, a tak usnadnit tvorbu interaktivních prvků aplikace. Ke statistickým výpočtům v testovací aplikaci je vhodné použít knihovny Pythonu *numpy*, *scipy*, *statsmodels* a *matplotlib*. K práci s datovými soubory pak knihovnu *urllib2*, která nám umožňuje jednoduše načítat data ze souborů.

4.2.2 Integrace

Integrace modulů Sage do webové aplikace pomocí Sage Cell Server proběhla následovně:

1. Nejprve byly do HTML hlavičky stránky umístěny nutné tagy skript s odkazem na javascriptové knihovny jquery a Sage Cell Server. Do skriptu stránky pak byla umístěna funkce, která posléze mění objekty se zvolenou třídou (v našem případě objekty div se třídou compute) na buňky Sage Cell Server. V atributech této funkce bylo ponecháno výchozí nastavení pro interpretaci kódu (jazyk Sage), ostaní nastavení bylo změněno tak, aby se kód automaticky vyhodnocoval při načtení stránky a aby bylo skryto vyhodnocovací tlačítko a zdrojový kód buněk.

```
sagecell.makeSagecell({inputLocation: 'div.compute',
                       template: sagecell.templates.minimal,
                       autoeval: true,
                       hide: ["evalButton",]});
    });
```

2. Poté byly do těla dokumenty vloženy buňky (viz analýza) s kódem Sage.

4.2.3 Implementace výpočtů

Načítání souborů v Sage bylo provedeno pomocí knihovny *urllib2* a její funkce `urlopen(url)`. Výpočty základních charakteristik veličin byly provedeny hlavně pomocí funkcí obsažených v knihovně Pythonu *numpy*, a to konkrétně `median(data)`, `mean(data)`, `cov(data1,data2)`, jenž funkcionalitou odpovídají stejnojmeným funkcím v R. Dále byla použita funkce `var(data)` pro výpočet rozptylu, ta ovčem nepočítá bodový odhad výběrového rozptylu, a tak musel být výsledek upraven. Byla také použita funkce z knihovny Pythonu *scipy* `stats.pearsonr(data1, data2)` pro výpočet korelačního koeficientu. Grafy byly vykresleny s pomocí knihoven *matplotlib*, *numpy*, *statsmodels* a *scipy*, zároveň byl použit příkaz `interact`.

1. **Histogram**

   Stejně jako v R byly odhadnuty hodnoty pro teoretická rozdělení metodou maximální věrohodnosti (viz analýza). Pro získání jeich

hustoty pravděpodobnosti byly použity funkce `scipy.stats.norm.pdf`, `scipy.stats.expon.pdf`, `scipy.stats.uniform.pdf` s příslušnými parametry. Samotný histogram pak byl vykreslen funkcí `matplotlib.pyplot.hist`.

2. **Graf empirické distribuční funkce**

   Opět byly odhadnuty parametry teoretických rozdělení a jako parametry grafu poté použita funkce rozdělení získaná pomocí funkcí `scipy.stats.norm.cdf`, `scipy.stats.expon.cdf` a `scipy.stats.uniform.cdf`. Souhrnná empirická distribuční funkce byla spočítána pomocí funkce `statsmodels.api.distributions.ECDF(data)` a schodovitá křivka tohoto grafu vykreslena pomocí `matplotlib.pyplot.step(data,ecdf(data)`.

3. **Q-Q plot**

   V tomto případě byla pro vykreslení grafu použita funkce `scipy.stats.problot`, jíž byly jako parametry předány odhadnuté parametry a název teoretického rozdělení, dále také zadaná data a objekt grafu, do něhož byla data vykreslena.

4. **Scatter plot**

   Tento graf byl vykreslen pomocí funkce `matplotlib.pyplot.scatter(data1,data2)`.

K testování hypotéz byly použity funkce z knihovny *scipy*. Konkrétně potom `stats.ttest_1samp(data, null_hypothesis)` pro oboustranný jednovýběrový t-test o střední hodnotě μ, `stats.ttest_rel(data1, data2)` pro párový t-test o rovnosti středních hodnot, pro Kolmogorov-Smirnovův test dobré shody `stats.kstest(data, teorethical_distribution)` pro porovnání s teoretickým rozdělením a `scipy.stats.ks_2samp(data1,data2)` pro porovnání dvou veličin. Použití těchto funkcí uživateli nedovoluje nastavit hladinu testu α. Pro testování hypotéz bylo opět využito příkazu `interact`.

4.3 Srovnání systémů a vyhodnocení

Pro integraci prostředí R do webové stránky je možné využít mnoho nástrojů. Jedním z nich je také API poskytované projektem OpenCPU. Od ostatních projektů se OpenCPU liší hlavně tím, že je kompatibilní s jakýmkoliv jazykem či frameworkem, který umožňuje komunikaci pomocí protokolu HTTP. Další jeho přednost je, že vytváří pouze spojení mezi prostředím R a webovou aplikací a všechno ostatní je ponecháno na uživateli.

Sage poskytuje k integraci systému do webové stránky službu Sage Cell Server. Je to zatím jediné webové rozhraní pro Sage, kromě přímé integrace

Sage u webových stránek postavených na jazyce Python. Umožňuje nám jednoduchou možnost vložení buněk Sage do libovolné webové stránky a to pouze vložením několika skriptů v jazyce javascript a tagů se speciální třídou. Je tak umožněno, že kód může být vyhodnocen automaticky s načtením stránky.

Použití systému OpenCPU je naproti tomu složitější. K volání vlastních R funkcí je nutné vytvořit balíček R a umístit ho na server poskytující tuto službu. Volání funkcí a manipulace s jejich výstupy je pak prováděna pomocí HTTP požadavků GET a POST.

Oba systémy jsou Open Source a jsou tak uživatelům volně k dispozici. Poskytují pro použití veřejné servery nebo je uživateli umožněna instalace služby na serveru vlastním, pro což je u obou k dispozici podrobný návod a čas instalace je srovnatelný. Obě služby také uživatelům poskytují rozsáhlou dokumentaci včetně funkčních příkladů použití.

Služba Sage poskytuje pro vyhodnocení buněk interprety různých jazyků, včetně pro statistické výpočty vhodného jazyka R. Jejich použití však zdaleka nemusí být tak komfortní, jak jsou uživatelé u těchto prostředí zvyklí. Veřejný server Sage Cell Server také zatím nepodporuje některé jejich funkcionality.

OpenCPU je oproti tomu určeno pouze pro jazyk R. Jeho použití však zachovává všechny funkcionality tohoto jazyka a kód R pro použití API poskytované projektem OpenCPU není nutné nijak měnit.

Velkou výhodou Sage je příkaz `interact`. Ten uživateli umožňuje vkládat do svých modulů interaktivní prvky jako jsou posuvníky, zaškrtávací, textová a výběrová pole. To značně usnadňuje tvorbu modulů, kde je nutná interakce s uživatelem.

Prostředí R je oproti tomu statické. OpenCPU API neposkytuje žádnou možnost vkládání interaktivních prvků. Ty lze ale ve stránce nahradit použitím jazyka PHP, nebo Javascriptu. Javascriptová knihovna opencpu poskytuje funkce usnadňující tvorbu interaktivních prvků ve webové stránce.

Sage v jistém smyslu omezuje svého uživatele nutností zobrazovat výstup ve výstupní buňce Sage. Veškeré jeho formátování je tedy nutné provádět v jazyce, v kterém je buňka interpretována. Výstupní buňky mají jednotný vzhled, jsou ohraničeny rámečkem s odkazem na domovskou stránku SageMath, který nelze skrýt.

OpenCPU naproti tomu nechává svému uživateli naprostou volnost ve zpracování výstupu. Id relace požadavku může být dokonce použito jako vstup dalšího volání funkce. Pro výstup lze zvolit velké množství formátů, mezi které patří také json, csv, png, svg a pdf, nebo lze použít výstup jako prostý text a jeho formátování provést v jazyce webové aplikace.

Sage je šířen pod licencí GNU. Jednou z podmínek této licence je, že případná díla odvozená musí být dále šířena také pod touto licencí. To sice úplně nevylučuje komerční šíření programu, je ale nutné v tomto případě nechávat jeho kód opět volně k dispozici. Pro cílového uživatele webové aplikace to

také znamená nutnost potvrzovat souhlas s licenčními podmínkami při vstupu na stránku s buňkami Sage.

OpenCPU je šířeno pod licencí Apache2, které narozdíl od licence GNU nepožaduje po uživateli copyleft. Uživatel tak nemusí modifikovaný program šířit opět pod stejnou licencí. Případná díla odvozená musí pouze zachovávat autorství nemodifikovaných částí, můžou však být použita buď jako open source nebo pro komerční či akademické účely.

Jazyk a prostředí R jsou primárně určeny ke statistickým výpočtům a grafice. Poskytují tak k tomuto účelu širokou škálu funkcí a rozšiřujících balíčků. K základním statistickým výpočtům stačí využít standartní nabídku funkcí, není nutné instalovat žádné další balíčky.

Jazyk Sage je určen obecně pro matematické výpočty a geometrické experimentování. Některé knihovny jazyka Python poskytují základní statistické funkce. Jejich využití ale není tak konfortní, jako využití jazyka R a v některých funkcích tak například nemáme tak široké možnosti nastavení jako v prostředí R.

Závěr

Hlavním cílem práce bylo prozkoumat možnosti integrace volně dostupných matematických systémů R a Sage do webových aplikací. Prvním cílem analytické části práce bylo analyzovat možností napojení statistického software R do webových aplikací, zvláště pak využitím API poskytovaného projektem OpenCPU a obdobně poté analyzovat možnosti napojení matematického software Sage, se zaměřením na službu Sage Cell Server. Dalším cílem analytické části pak bylo prozkoumat závislost těchto řešení na konkrétních webových technologiích. Cílem praktické části práce bylo navrhnout a vytvořit jednoduchou webovou aplikaci testující možnosti obou těchto systémů na několika základních statistických úlohách. Na závěr pak bylo cílem tato dvě řešení porovnat především z hlediska praktické použitelnosti, efektivity a jednoduchosti integrace.

V první části práce byly nejprve analyzovány možnosti integrace R do webové aplikace také v závislosti na konkrétních webových technologiích. Poté byl rozebrán systém OpenCPU a popsány možnosti, které tento projekt nabízí. V další části byl pak obdobně rozebrán projekt Sage a jeho služba pro integraci do webové aplikace Sage Cell Server. Analytická část práce by se tak dala zhodnotit jako úspěšná. Bylo zjištěno, že systém OpenCPU není příliš závislý na použitých webových technologiích, protože je kompatibilní s jakýmkoliv frameworkem nebo jazykem, který umožňuje komunikaci pomocí protokolu HTTP. Také služba Sage Cell Server umožňuje vložení buněk Sage do libovolné webové stránky.

K implementaci testovací aplikace byl zvolen jazyk PHP. Ve třetí části pak byla prozkoumána teorie k výpočtům prováděným v této aplikaci. Byla tak vytvořena jednoduchá interaktivní webová aplikace pro základní statistické výpočty, která úspěšně integrovala systémy R i Sage.

V závěrečné části práce byla porovnána řešení pomocí služby OpenCPU a Sage Cell Server. Bylo zjištěno, že velkou předností Sage je jednoduchost jeho integrace, dále pak možnost využití různých interpretů pro vyhodnocení kódu, ačkoliv ve webové aplikaci pouze s omezenými možnostmi, a také pří-

kaz `interact`. Přednostmi projektu OpenCPU jsou pak neomezené možnosti formátování výstupů, použitá licence, která umožňuje šířit modifikace bez nutnosti nechávat kód veřejně k dispozici a pro účely statistických výpočtů je to především možnost plnohodnotně použít jazyk R. Ukázalo se také, že integrace systému OpenCPU do webové aplikace je naproti Sage složitější a lze tak říci, že je z hlediska praktické použitelnosti lepší systém Sage Cell Server. V případě složitějších statistických výpočtů je pak ale vhodnější použít jazyk R a to se všemi možnostmi, které nabízí, a tak je vhodné použít systém OpenCPU. Ten byl také vyhodnocen jako efektivnější, díky způsobu, jakým spravuje relace. To funguje tak, že si server pamatuje id proběhlých relací a v případě opětovného volání stejných požadavků nejsou funkce prováděny znovu, pouze je vráceno id relace, která již proběhla. Naproti tomu jsou buňky Sage opětovně vyhodnocovány po každém znovunačtení stránky.

Hlavním přínosem práce je provedená analýza možností integrace systémů R a Sage do webové aplikace a porovnání obou provedených řešení, které pomohlo upozornit na kladné i záporné vlastnosti obou systémů. Práce na testovací aplikaci by mohla dále pokračovat implementací složitějších statistických úloh a mohla by se tak věnovat hlubší analýze možností použití obou systémů pro statistické výpočty ve webových aplikacích.

Literatura

[1] STIGLER, Stephen M.. *The History of Statistics: The Measurement of Uncertainty before 1900.* Harvard : Belknap Press of Harvard University Press, 1990. ISBN 978-0674403413.

[2] *R: What is R?* [online]. The R Foundation. [cit. 2015-04-26]. Dostupné z: http://www.r-project.org/

[3] *R-project* [online]. The R Foundation. [cit. 2015-04-26]. Dostupné z: http://www.r-project.org/

[4] *R FAQ – R Web Interfaces* [online]. The R Foundation. [cit. 2015-04-26]. Dostupné z: http://cran.r-project.org/doc/FAQ/R-FAQ.html#R-Web-Interfaces

[5] *Rweb* [online]. BANFIELD, Jeff. [cit. 2015-04-26]. Dostupné z: http://www.math.montana.edu/Rweb/

[6] BANFIELD, Jeff. Rweb: Web-based Statistical Analysis. *Journal of Statistical Software* [online]. 1999, 1-15. ISSN 1548-7660. [cit. 2015-04-26]. Dostupné z: http://www.jstatsoft.org/v04/i01/paper

[7] *General Rweb interface* [online]. THIOULOUSE, Jean. (2008). [cit. 2015-04-26]. Dostupné z: http://pbil.univ-lyon1.fr/Rweb/Rweb.general.html

[8] *Open Source Visions – R online* [online]. BARTEL, Ulf. (2002). [cit. 2015-04-26]. Dostupné z: http://www.osvisions.com/r_online/

[9] *Web Decomp* [online]. SATO, Seisho. [cit. 2015-04-26]. Dostupné z: http://ssnt.ism.ac.jp/inets2/title.html

[10] *About Rserve* [online]. URBANEK, Simon. [cit. 2015-04-26]. Dostupné z: http://rforge.net/Rserve/

[11] *What is FastRWeb?* [online]. URBANEK, Simon. (2012). [cit. 2015-04-26]. Dostupné z: http://rforge.net/FastRWeb/

[12] FIRTH, David. CGIwithR: Facilities for processing web forms using R. *Journal of Statistical Software* [online]. 2003, 1-8. ISSN 1548-7660. [cit. 2015-04-26]. Dostupné z: http://www.jstatsoft.org/v08/i10/paper

[13] *CGIwithR package* [online]. FIRTH, David. (2014). [cit. 2015-04-26]. Dostupné z: http://www.omegahat.org/CGIwithR/

[14] KRAUS, Jiří et al. *Nový akademický slovník cizích slov A-Ž.* Praha: ACADEMIA, 2007. ISBN 80-200-1415-2.

[15] *Rook package* [online]. HORNER, Jeffrey. (2014). [cit. 2015-04-27]. Dostupné z: http://cran.r-project.org/web/packages/Rook/Rook.pdf

[16] *Web Application Development with R and Apache* [online]. Vanderbilt University. (2013). [cit. 2015-04-27]. Dostupné z: http://rapache.net/index.html

[17] *Package brew* [online]. HORNER, Jeffrey. (2010). [cit. 2015-04-27]. Dostupné z: http://cran.r-project.org/web/packages/brew/brew.pdf

[18] *Rwui: A web application to create user friendly web interfaces for R scripts* [online]. NEWTON, R., WERNISCH, L.. [cit. 2015-04-27]. Dostupné z: http://sysbio.mrc-bsu.cam.ac.uk/Rwui/tutorial/Rwui_Rnews_final.pdf

[19] *R-php* [online]. PONTILLO, Alfredo, MINEO, Angelo. (2005). [cit. 2015-04-27]. Dostupné z: http://dssm.unipa.it/R-php/?cmd=home

[20] *R_PHP_ONLINE* [online]. Steve Chen. (2003). [cit. 2015-04-27]. Dostupné z: http://steve-chen.net/document/r/r_php

[21] *Shiny* [online]. RStudio. (2014). [cit. 2015-04-27]. Dostupné z: http://shiny.rstudio.com/

[22] *OpenCPU – Papers* [online]. OpenCPU. (2013). [cit. 2015-04-27]. Dostupné z: https://www.opencpu.org/papers.html

[23] OOMS, Jeroen. OpenCPU release 1.4.6: gzip and systemd. *OpenCPU* [online]. OpenCPU, 2014. [cit. 2015-04-27]. Dostupné z: https://www.opencpu.org/posts/opencpu-release-1-4-6/

[24] *OpenCPU – JavaScript Client* [online]. OpenCPU. (2014). [cit. 2015-04-27]. Dostupné z: https://www.opencpu.org/jslib.html

[25] *Information on package 'A3'* [online]. OOMS, Jeroen. (2013). [cit. 2015-04-27]. Dostupné z: https://demo.ocpu.io/A3/info

[26] *Writing R Extensions* [online]. The R Foundation. [cit. 2015-04-27]. Dostupné z: http://cran.r-project.org/doc/manuals/r-release/R-exts.html#Creating-R-packages

[27] *OpenCPU – API Docs* [online]. OpenCPU. [cit. 2015-04-27]. Dostupné z: https://www.opencpu.org/api.html#api-ci

[28] *SageMath* [online]. SageMath Mathematical Software. [cit. 2015-04-27]. Dostupné z: http://www.sagemath.org/

[29] STEIN, William. Sage - Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab. In: *Foundations of Computational Mathematics, Budapest 2011.* Cambridge, England: Cambridge University Press, 2012. ISBN 9781107604070.

[30] *Sage Developer Trac* [online]. SageMath Mathematical Software. (2015). [cit. 2015-04-26]. Dostupné z: http://trac.sagemath.org/

[31] *About Cython* [online]. [cit. 2015-05-01]. Dostupné z: http://cython.org/#about

[32] *SageMathCloud* [online]. SageMathCloud. [cit. 2015-04-26]. Dostupné z: https://cloud.sagemath.com/

[33] *Interpreter Interfaces* [online]. The Sage Development Team. (2005 – 2015). [cit. 2015-04-26]. Dostupné z: http://www.sagemath.org/doc/reference/interfaces/

[34] *Interact – Sage Wiki* [online]. [cit. 2015-05-01]. Dostupné z: http://wiki.sagemath.org/interact/

[35] *About the Sage Cell Server* [online]. (2011). [cit. 2015-05-01]. Dostupné z: http://sagecell.sagemath.org/static/about.html?v=15adefe8b7e89fcf49eda7af5303abd4

[36] *Coding in Cython – Sage Developer's Guide v6.6.beta0* [online]. The Sage Development Team. (2005 – 2015). [cit. 2015-05-01]. Dostupné z: http://www.sagemath.org/doc/developer/coding_in_cython.html

[37] *Generate pseudo-random numbers* [online]. Python Software Foundation. (1990-2015). [cit. 2015-04-26]. Dostupné z: https://docs.python.org/3.4/library/random.html

[38] *Mathematical statistics functions* [online]. Python Software Foundation. (1990-2015). [cit. 2015-04-26]. Dostupné z: https://docs.python.org/3/library/statistics.html

[39] *Scientific Computing Tools for Python – SciPy.org* [online]. SciPy developers. (2015). [cit. 2015-04-26]. Dostupné z: http://www.scipy.org/about.html

[40] *NumPy* [online]. Numpy developers. (2013). [cit. 2015-04-26]. Dostupné z: http://www.numpy.org/

[41] *matplotlib: python plotting* [online]. HUNTER, John et al.. (2002-2012). [cit. 2015-04-26]. Dostupné z: http://matplotlib.org/

[42] *Jupyter and the future of IPython* [online]. IPython development team. (2015). [cit. 2015-04-26]. Dostupné z: http://ipython.org/

[43] *pandas: Python Data Analysis Library* [online]. Python Data Analysis Library. (2015). [cit. 2015-04-26]. Dostupné z: http://pandas.pydata.org/

[44] *rpy2.* [online]. GAUTIER, L.. (2014). [cit. 2015-04-26]. Dostupné z: http://rpy.sourceforge.net/

[45] *python-statlib – descriptive statistics for the python programming language* [online]. Google Project Hosting. (2007). [cit. 2015-04-26]. Dostupné z: https://code.google.com/p/python-statlib/

[46] *StatsDoc – python-statlib – documentation for the stats module.* [online]. Google Project Hosting. [cit. 2015-04-26]. Dostupné z: https://code.google.com/p/python-statlib/wiki/StatsDoc

[47] *PstatDoc – python-statlib – documentation for the pstat module.* [online]. Google Project Hosting. [cit. 2015-04-26]. Dostupné z: https://code.google.com/p/python-statlib/wiki/PstatDoc

[48] *Documentation for MATFUNC.* [online]. HETTINGER, Raymond. (2001). [cit. 2015-04-26]. Dostupné z: http://users.rcn.com/python/download/matfunc.htm

[49] *StatsModels: Statistics in Python* [online]. The Statsmodels Development Team. (2012). [cit. 2015-04-26]. Dostupné z: http://statsmodels.sourceforge.net/

[50] *Introduction* [online]. FONNESBECK, Christopher J.. (2014). [cit. 2015-04-26]. Dostupné z: https://pymc-devs.github.io/pymc/README.html

[51] *PyMix/home* [online]. [cit. 2015-04-26]. Dostupné z: `http://www.pymix.org/pymix/`

[52] *sagemath/sagecell* [online]. [cit. 2015-04-26]. Dostupné z: `https://github.com/sagemath/sagecell`

[53] ANDĚL, Jiří. *Základy matematické statistiky.* Druhé upravené vydání. Praha: MATFYZPRESS, 2007. ISBN 80-7378-001-1.

[54] PAVLÍK, Jiří et al. *Aplikovaná statistika.* Praha: VŠCHT Praha, 2005. ISBN 80-7080-569-2.

[55] CHAMBERS, John M. et al. *Graphical methods for data analysis.* New Jersey: Wadsworth & Brooks, 1983. ISBN 0-534-98052.

[56] *Usage Statistics and Market Share of Server-side Programming Languages for Websites.* [online]. W3Techs. (2015). [cit. 2015-05-10]. Dostupné z: `http://w3techs.com/technologies/overview/programming_language/all`

Příloha **A**

Seznam použitých zkratek

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

CAS Computer Algebra System

CDF Cumulative Distribution Function

CGI Common Gateway Interface

CORS Cross Domain OpenCPU Request

CRAN Comprehensive R Archive Network

CSS Cascading Style Sheets

CSV Comma-Separated Values

cURL client URL Library

FAQ Frequently Asked Questions

GNU General Public License

GPL General Public License

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IRT Item Response Theory

JSON JavaScript Object Notation

MLE Maximum Likelihood Estimation

PHP Hypertext Preprocessor

Q-Q plot Quantile-Quantile plot

RPC Remote Procedure Call

SSL Secure Sockets Layer

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `www` ..... adresář se zdrojovými kódy webové aplikace
  - `img` ..... obrázky do webových stránek
  - `css` ..... kaskádové styly aplikace
  - `*` ..... PHP stránky a potřebné javascriptové knihovny
- `src`
  - `newPackage` ..... balíček R funkcí
  - `data` ..... data ve formátu CSV pro testování aplikace
  - `text` ..... soubory k vygenerování textu práce
    - `BP_Ernekerová_Jana_2015.tex` .. zdrojová forma práce ve formátu LaTeX
    - `*` ..... ostatní soubory šablony potřebné k vygenerování práce
- `text` ..... text práce
  - `BP_Ernekerová_Jana_2015.pdf` ..... text práce ve formátu PDF