

Uživatelské rozhraní pro realizaci aplikačních úloh pro práci s R systémem

A User Interface for the Execution of
Application Tasks in the R

Libor Štefek



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Libor ŠTEFEK**
Osobní číslo: **A10887**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Uživatelské rozhraní pro realizaci aplikačních úloh pro práci s R systémem**

Zásady pro vypracování:

1. Nastudujte práci se systémem R.
2. Vypracujte rešerši hodnotící uživatelská rozhraní pro práci se statistickým software R.
3. Navrhněte vlastní model uživatelského rozhraní podle požadavků zadavatele. Uživatelské rozhraní bude zaměřeno na využití existujícího modulu funkcí pro datovou analýzu v prostředí R (funkční package).
4. Zvolte a nastudujte vhodnou technologii.
5. Realizujte navržený model uživatelského rozhraní.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. The R Journal [online časopis]. 2009– [cit. 2012]. Dostupné z: <http://journal.r-project.org/>. ISSN: 2073-4859.
2. R News [online časopis]. 2001–2008 [cit. 2012]. Dostupné z: <http://journal.r-project.org/>. ISSN: 1609-3631.
3. Dalgaard, Peter. *Introductory Statistics with R*. New York: Springer, 2002. ISBN 0387954759.
4. The R Foundation. *The R Project for Statistical Computing* [online]. [cit. 2012]. Dostupné z: <http://www.r-project.org/>.
5. Garfinkel, Simson a Gene Spafford. *Bezpečnost v UNIXu a Internetu v praxi*. Praha: Computer Press, 1998. ISBN 80-722-6082-0.
6. Falkner, Jayson a Kevin Jones. *Servlets and JavaServer Pages: The J2EE Technology Web Tier*. Boston, MA: Addison Wesley, 2003. ISBN 0-321-13649-7.
7. Kurniawan, Budi. *Java for the Web with Servlets, JSP and EJB*. 1st ed. Indianapolis : New Riders, 2002. ISBN 0-7357-1195-X.
8. Wutka, M., Moffet, A. a Mittal K. *Sams Teach Yourself JavaServer Pages 2.0 with Apache Tomcat in 24 Hour*. Indianapolis, Ind.: Sams Publishing, 2003. ISBN 0-672-32597-7.
9. Kabir, Mohammed J. *Apache Server 2 : kompletní příručka administrátora*. Brno: Computer Press, 2004. ISBN 80-251-0319-6.

Vedoucí diplomové práce:

Ing. Tomáš Dulík

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

21. května 2012

Ve Zlíně dne 24. února 2012



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Tato práce si klade za cíl zvolit vhodnou technologii, navrhnout a implementovat systém pro spouštění uživatelských aplikačních úloh se systémem R. V teoretické části jednak popisuje základní vlastnosti R a dále také shrnuje informace o existujících grafických uživatelských rozhraních pro R. V praktické části pak prezentuje dvě možné cesty k dosažení cílů této práce. Jednak je to aplikace Rweb, která na základní úrovni ukazuje možnost integrace webové aplikace a R. V poslední části pak je představeno řešení pomocí existujícího programového produktu LabKey.

Klíčová slova: R, R software, R project, datová analýza, GUI, Java, LabKey

ABSTRACT

This thesis aims to choose and implement an appropriate technology as well to design a system for the execution of application tasks in the R. In the theoretical part it describes core functionality of the R system and summarizes various existing graphical user interfaces for the R. The practical part presents two possible ways to achieve the objectives of this thesis. The first one represents the application RWeb which shows basic interaction of a web application and the R engine. The second one is using an existing software product LabKey and it shows practical, nearly step by step, implementation focusing to resolve all provided requirements.

Keywords: R software, R project, data analysis, GUI, Java, LabKey

PODĚKOVÁNÍ

Chtěl bych na tomto místě poděkovat všem, bez kterých by tato práce jen stěží vznikla. Jsou to ti, kteří mi pomohli dobrou radou, ale také ti, kteří mě podporovali a byli pro mě základem nezbytným k napsání této práce, zejména pak své manželce Marii. Děkuji Ing. Tomáši Dulíkovi, Ph.D., za trpělivost, vstřícnost a cenné připomínky při vedení diplomové práce.

Motto

„Devadesát procent práce na projekt zabere 90% času,
zbylých deset procent zabere dalších devadesát procent času.“

Myrphyho zákon o přesném odhadu pracnosti projektu.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

Obsah

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 ÚVOD DO SYSTÉMU R	12
1.1 Co je to R?	12
1.2 Historie	12
1.3 The R Project	13
1.4 Instalace R	14
1.4.1 GNU/Linux	14
1.4.2 MS Windows	15
1.5 Základy práce s R	15
1.5.1 Spuštění a ukončení R, nápověda	15
1.5.2 Základní datové typy, vektory, proměnné a objekty v R	17
1.5.3 Základní matematické operace	19
1.5.4 Další datové typy	19
1.5.5 Práce s datovými soubory	22
1.5.6 Grafy	23
1.5.7 Řízení běhu programu	26
1.5.8 Uživatelem definované funkce	27
1.5.9 Balíčky	28
2 GRAFICKÁ UŽIVATELSKÁ ROZHRAŇÍ PRO R	29
2.1 Cíle a kritéria porovnání	29
2.2 R Commander	30
2.2.1 Charakteristika	30
2.2.2 Technické informace	32
2.2.3 Instalace	33
2.2.4 Hodnocení	34
2.3 Poor Man's GUI – PMG	34
2.3.1 Charakteristika	34
2.3.2 Technické informace a instalace	35
2.3.3 Hodnocení	36
2.4 RStudio	36

2.4.1	Charakteristika	36
2.4.2	Technické informace.....	36
2.4.3	Instalace.....	37
2.4.4	Hodnocení	37
2.5	RKWARD	38
2.5.1	Charakteristika	38
2.5.2	Technické informace a instalace.....	38
2.5.3	Hodnocení	39
2.6	RATTLE	40
2.6.1	Charakteristika	40
2.6.2	Technické informace.....	41
2.6.3	Instalace.....	41
2.6.4	Hodnocení	42
2.7	REXCEL.....	42
2.7.1	Charakteristika	42
2.7.2	Technické informace a instalace.....	42
2.8	RGNUMERIC.....	44
2.8.1	Charakteristika	44
2.9	JGR – JAVA GUI FOR R	44
2.9.1	Charakteristika	44
2.9.2	Technické informace a instalace.....	45
2.10	DEDUCER.....	45
2.10.1	Charakteristika	45
2.10.2	Technické informace a instalace.....	47
2.11	RED-R.....	48
2.11.1	Charakteristika	48
2.11.2	Technické informace a instalace.....	48
2.12	OSTATNÍ – NÁSTROJE PRO GUI.....	48
2.12.1	R-Tcl/Tk.....	48
2.12.2	RGtk	49
2.12.3	RGG.....	50
2.13	OSTATNÍ – EDITORY A ROZŠÍŘENÍ PRO EDITORY	51
2.14	OSTATNÍ NÁSTROJE	51

II	PRAKTICKÁ ČÁST	53
3	ÚVOD DO PRAKTICKÉ ČÁSTI	54
4	WEBOVÉ UŽIVATELSKÉ ROZHRAŇÍ PRO R	55
	4.1 RWEB.....	55
	4.2 DALŠÍ VÝVOJ RWEB	56
5	LABKEY	58
	5.1 ZÁKLADNÍ INFORMACE.....	58
	5.2 TECHNICKÁ ARCHITEKTURA LABKEY.....	59
	5.3 INSTALACE A ÚVODNÍ KONFIGURACE.....	60
	5.3.1 Instalace LabKey ve Windows.....	60
	5.3.2 Administrace LabKey	61
	5.4 INTEGRACE LABKEY A R	62
	5.5 POUŽITÍ LABKEY – PRVNÍ PROJEKT	63
	5.5.1 Projekt typu Study.....	64
	5.6 LABKEY – PROBLÉMY A NÁMĚTY NA VYLEPŠENÍ	67
	5.6.1 Lokalizace LabKey.....	68
	5.6.2 Problémy s diakritikou	69
	5.6.3 Způsob editace dat	69
	5.6.4 Úroveň integrace s R.....	69
	5.7 LABKEY – ZÁVĚR	69
	ZÁVĚR	71
	ZÁVĚR V ANGLIČTINĚ	72
	SEZNAM POUŽITÉ LITERATURY	73
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	76
	SEZNAM OBRÁZKŮ	77
	SEZNAM TABULEK	78

ÚVOD

Velmi stručně shrnuto, cílem této práce je analyzovat, navrhnout a realizovat systém, který umožní uživateli spouštět předpřipravené úlohy z oblasti statistické analýzy dat nad vlastními daty, přičemž vlastní výkonná úloha bude prováděna v prostředí R. To znamená, že cílový systém má umožnit uživateli nejprve zadat nebo importovat data k analýze, dále je různým způsobem upravovat a poté umožnit spouštět úlohy z oblasti statistické analýzy dat. Výstupem těchto úloh je jak textový, tak grafický výstup.

První, teoretická část této práce se skládá ze dvou kapitol a to jednak kapitola uvádějící do samotného prostředí a jazyka R a v druhé kapitole je provedeno hodnocení některých existujících grafických uživatelských rozhraní a posouzení vhodnosti použití těchto nástrojů (s případným rozšířením) pro dosažení cílů této práce.

V druhé, praktické části práce jsou v rámci úvodní analýzy shrnuty požadavky na cílový systém, v další kapitole je pak představen prototyp aplikace implementující grafické uživatelské rozhraní v prostředí webového serveru a klienta v internetovém prohlížeči. V poslední kapitole je pak představeno komplexní řešení požadované funkcionality pomocí existujícího programového balíku LabKey.

I. TEORETICKÁ ČÁST

1 ÚVOD DO SYSTÉMU R

1.1 Co je to R?

R je integrovaný softwarový balík pro datové manipulace, výpočty a grafický výstup. Jádrem systému R je programovací jazyk a výpočetní prostředí zejména pro statistické výpočty a související grafický výstup. Jde o volně šířený software, který je distribuován pod licencí GPL a je oficiální součástí GNU projektu pod názvem „GNU S“. Programovací jazyk R je otevřenou reimplementací jazyka S, který byl spolu s prostředím S-PLUS vyvinut v Bellových laboratořích jako komerční produkt. Ačkoli mezi jazyky R a S existují některé důležité rozdíly, lze většinou kód psaný pro S spustit beze změn pod R.

R nabízí širokou škálu statistických funkcí (lineární a nelineární modelování, klasické statistické testy, analýza časových řad, klasifikace, ...), která je obsažena už v základní instalaci nebo jako doplňující balíčky. Další výraznou vlastností R je dobrá podpora tvorby grafických výstupů a to i včetně matematických symbolů a vzorců.

Ve srovnání s programy zaměřenými na statistickou analýzu jako je např. Statsoft Statistica je R skutečně hlavně programovací jazyk – není zde žádné primární grafické uživatelské rozhraní, které by sloužilo k ovládání celého prostředí R. Existuje nicméně celá řada různě zaměřených grafických uživatelských rozhraní, které lze do R dodatečně doinstalovat.

1.2 Historie

Kořeny syntaxe jazyka R lze nalézt v Bell Laboratories, kde na počátku 80-tých let minulého století vyvinul tým Johna Chamberse jazyk S a prostředí S-PLUS interpretující tento jazyk. Následně se tento systém rozšířil mezi statistiky a získal zde velkou oblibu. V současnosti je tato komerční implementace S a S-PLUS vyvíjena firmou TIBCO Software Inc. pod názvem TIBCO Spotfire S+.

Vlastními autory R však jsou Robert Gentleman a Ross Ihaka z Ústavu statistiky na univerzitě v Aucklandu na Novém Zélandu, kteří se v roce 1992 rozhodli o vývoji vlastního softwaru pro podporu výuky statistiky a zvolili právě syntaxi jazyka S, kterou implementovali ve svém interpretru. Název R zvolili trochu jako žert podle svých jmen Robert a Ross.

V roce 1994 se rozhodli o uvolnění zdrojových kódů pod GPL licencí, ale vývoj

nadále prováděli především vlastními silami. Až konečně v roce 1997 došlo k založení CRAN archivu, zpřístupnění CVS repositáře širšímu týmu vývojářů (the „R Core Team“) a dokonce k přijetí R do rodiny GNU programů jako „GNU S“ projekt.

1.3 The R project

Domovská stránka projektu zastřešujícího vývoj prostředí R je na adrese <http://www.R-project.org/> a jeho oficiální název je „The R Project for Statistical Computing“. R projekt je provozován a řízen neziskovou organizací „The R Foundation for Statistical Computing“ sídlící ve Vídni. Jak už bylo naznačeno výše, R je integrovaná sada softwarových nástrojů pro manipulaci s daty, výpočty a jejich grafické zobrazení. Obsahuje:

- efektivní nástroje pro zpracování dat a jejich ukládání,
- sadu operátorů a funkcí pro výpočty s maticemi a poli,
- rozsáhlý, ucelený a integrovaný soubor dílčích nástrojů pro analýzu dat,
- nástroj pro grafickou reprezentaci datové analýzy a její výstup na obrazovce do souboru v tiskové kvalitě a
- dobře rozvinutý, jednoduchý a efektivní programovací jazyk [4].

R je navržen kolem skutečného počítačového jazyka, umožňuje uživatelům přidávat další funkčnost a má pro tento účel zavedený systém pro správu a distribuci nových funkcí. Značná část systému je napsána přímo v jazyku R. Pro výpočetně náročné úlohy je možné psát funkce v C, C++ nebo Fortranu a tento kód může být propojen s R za běhu.

Většina uživatelů považuje R za výhradně statistický systém. R je však natolik otevřené prostředí, že lze jeho uplatnění spatřit i v jiných matematicky orientovaných odvětvích a aplikacích. Pro některé úlohy jej lze použít například namísto MatLabu / GNU Octave nebo podobných aplikací.

R má vlastní dokumentaci v několika formátech dostupných na webových stránkách projektu a tato dokumentace je součástí instalace R a je dostupná přímo z produktu.

1.4 Instalace R

Kompletní software R je dostupný ze sítě webových (http a ftp); serverů - CRAN (The Comprehensive R Archive Network) [5].

Aktuálně je R dostupné pro platformy GNU/Linux, MacOS X a Windows. Pro některé linuxové distribuce a verze MacOS X existují předpřipravené balíčky o něco zjednodušující instalaci prostředky konkrétního operačního systému. Pro Windows je připraven instalační program, který obsahuje základní sestavu balíčků.

Všechny funkce, které lze v R použít, jsou soustředěny v balíčcích tzv. packages (někdy označovaných jako knihovny/libraries). Základním balíkem, který obsahuje mnoho používaných funkcí, a který je instalován automaticky, je balík base. Pro použití jiných funkcí, které v tomto balíku nejsou zahrnuty, je nutné příslušný balík nainstalovat (importovat). Balíky jsou rozděleny do dvou skupin :

- recommended – doporučené, často používané, obvykle bývají součástí instalačních balíčků a
- contributed – rozšiřující, vytvářené širokou komunitou přispěvatelů.

Aktuálně existuje více než 3000 rozšiřujících balíčků v různém stádiu vývoje. Seznam těchto balíčků spolu s krátkým popisem nalezneme v CRAN [5] – zde odkaz Packages.

Pro instalaci základu R je potřeba rezervovat asi 100MB volného diskového prostoru. Tento požadovaný diskový prostor však může snadno narůst až na několiknásobek podle množství doinstalovaných doplňkových balíčků a může dosáhnout až 500MB.

Paměťové nároky běžícího R začínají, v závislosti na architektuře procesoru, na 30–40 MB paměti, další nárůst spotřebované paměti závisí zejména na rozsahu zpracovávaných dat a komplexnosti zpracování.

1.4.1 GNU/Linux

R je součástí většiny linuxových distribucí, které používají centrální repositáře aplikací pro instalaci, takže obvykle stačí nainstalovat balíky R-base, případně R-common pomocí příslušného balíkovacího systému.

1.4.2 MS Windows

Instalace R v prostředí MS Windows není také nijak problematická. Navíc spolu se základní instalací získáte i základní grafické uživatelské rozhraní, které usnadní základní ovládání R, jako je instalace například balíčků, načtení a uložení sezení a podobně (obr. 1).

1.5 Základy práce s R

Většina zdrojů informací o prostředí R je k dispozici anglicky a to v poměrně dostatečném množství formou, jak volně dostupných popisů, návodů a tutoriálů, tak tištěných publikací. Česky publikovaných zdrojů je naopak velmi málo a navíc jsou někdy obtížně dohledatelné. Za zmínku určitě stojí „Cvičení z biostatistiky“ [7] nebo bakalářská práce „Výuka jazyka R“ [8].

1.5.1 Spuštění a ukončení R, nápověda

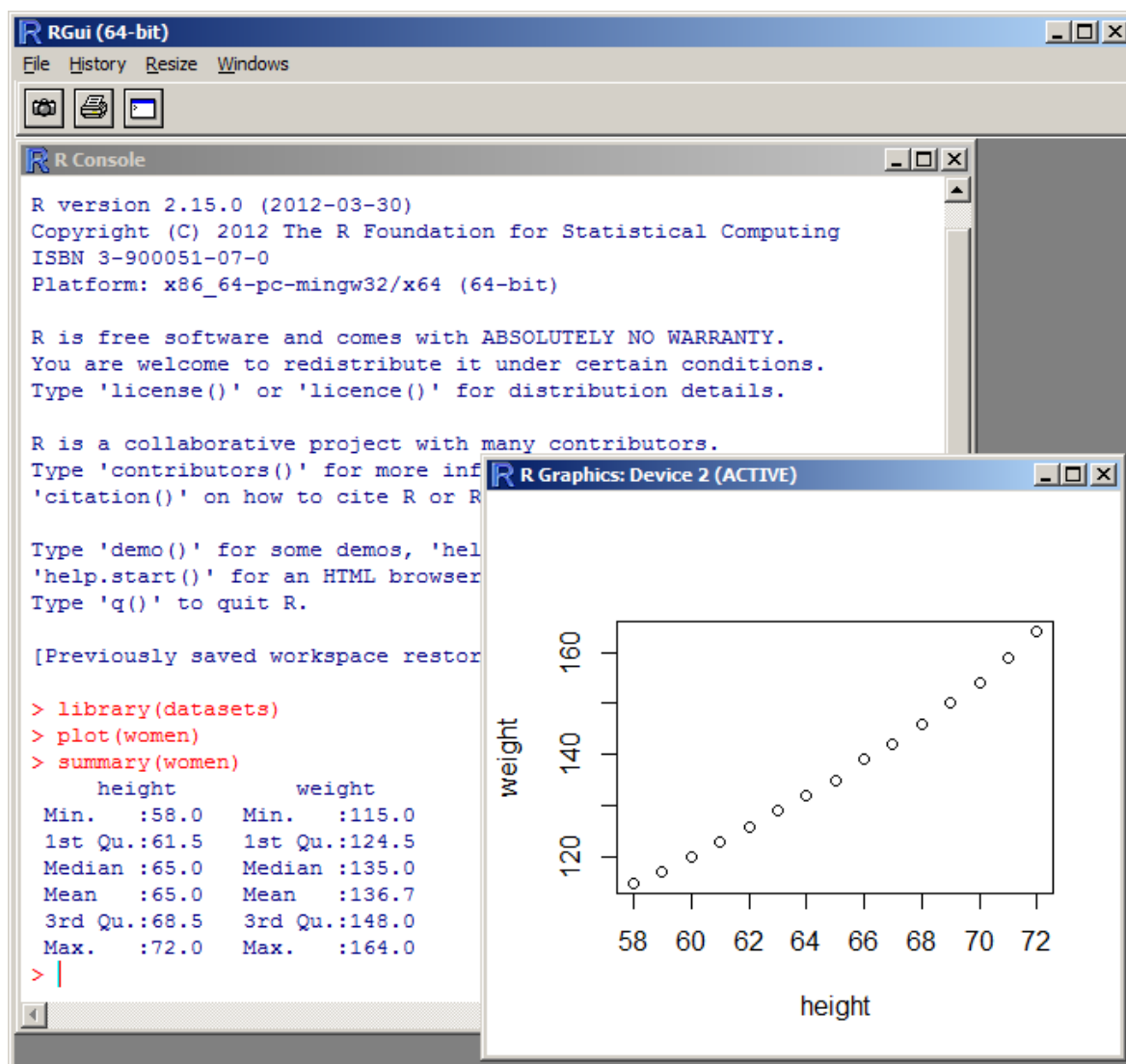
V závislosti na konkrétním operačním systému se poněkud liší způsob spouštění a základní uživatelské rozhraní systému R.

Pro spuštění R v linuxovém terminálu stačí zadat R a spustit; v tomto případě R nevytváří vlastní okno (obr. 2). Pro ukončení lze použít R funkci `q()` nebo i kombinaci kláves `Ctrl-D`. R se při ukončení zeptá, zda má uložit obraz sezení, tj. všechny objekty právě existující v paměti R, do souboru pro pozdější opětovné použití: `'Save workspace image? [y/n/c]:'` a očekává odpověď `y` (ano), `n` (ne) nebo `c` (zrušení požadavku na ukončení).

V případě operačního systému Windows máme už v základní instalaci k dispozici dvě varianty spuštění R a to jednak jako konzolovou aplikaci (`R.exe`), nebo R se základním grafickým rozhraním (`RGui.exe`).

Jak již bylo uvedeno, R je ve své podstatě příkazový interpret programovacího jazyka a tento interpret lze používat jednak interaktivně, kdy zadáváme jednotlivé příkazy z klávesnice pomocí příkazového řádku, nebo v dávkovém režimu, kdy předáme R interpretru soubor s připraveným příkazovým skriptem. Technicky řečeno, program R čte svůj standardní vstup, zapisuje textový výsledek na standardní výstup a chyby na chybový výstup.

Jazyk R je citlivý na velká a malá písmena a tedy například proměnné `X` a `x` se



Obrázek 1. R ve Windows

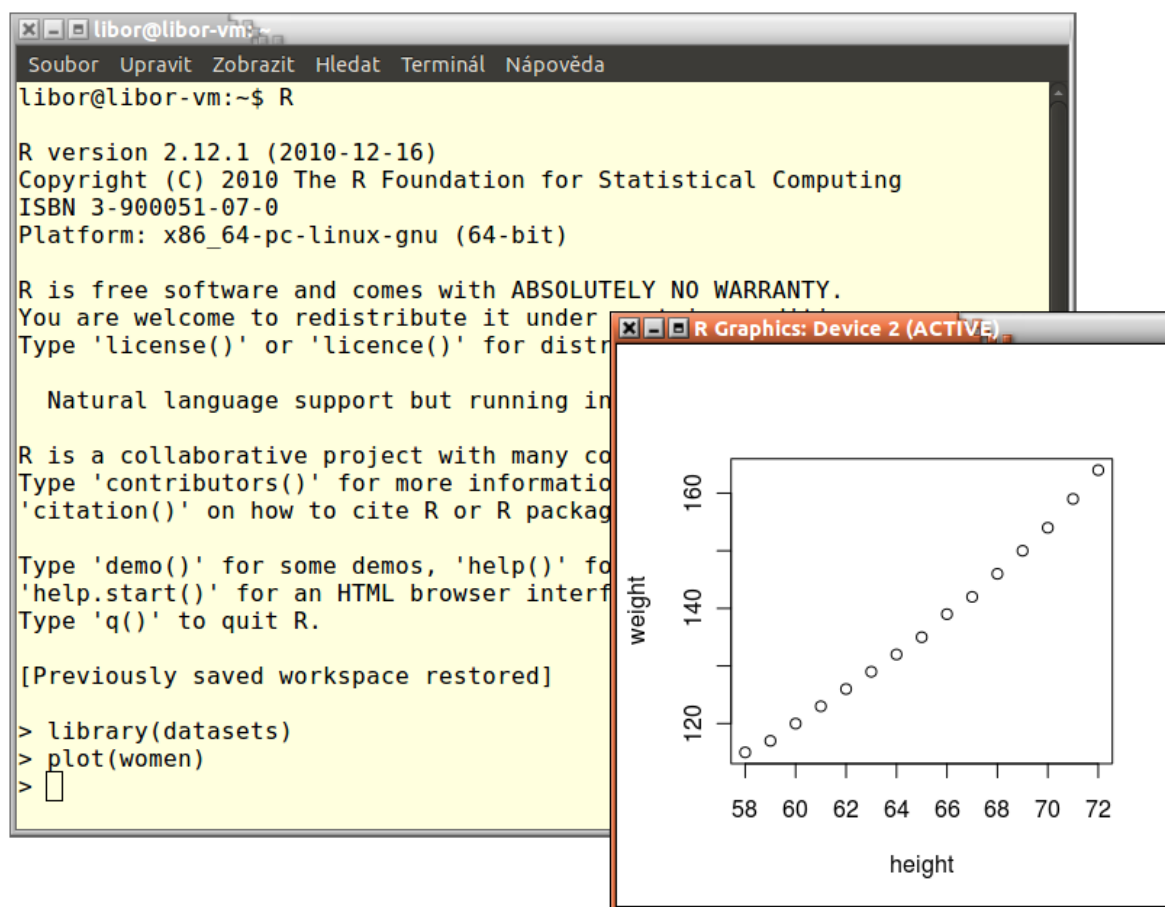
liší. Délka jmen identifikátorů není prakticky omezena (ve starších verzích to bylo 256 znaků). Identifikátory musí začínat písmenem, následovat mohou libovolnými alfanumerickými znaky a případně znaky '.' nebo '_'. Tečka nemá nějaký speciální význam, ale často bývá používána pro logické dělení názvu, podobně jako znak podtržení.

Komentářem je vše od znaku # do konce řádku, což ovšem pochopitelně neplatí uvnitř řetězců.

Získání nápovědy

Nápovědu pro určitou funkci, zde například `sqrt`, lze vyvolat pomocí zápisu: `?sqrt` nebo také `help(sqrt)`. Vyhledávání určitého řetězce v nápovědě lze provést pomocí zdvojeného otazníku: `??hledanýtext`.

V oblasti systému nápovědy lze ovšem nalézt určité rezervy. Hlavní nevýhodou pro



Obrázek 2. R v linuxovém terminálu

českého uživatele je absence lokalizované nápovědy. Celkově také míra detailu a kvalita informací nedosahuje úroveň, kterou nalezneme u komerčních statistických produktů.

1.5.2 Základní datové typy, vektory, proměnné a objekty v R

V R je do značné míry uplatněn objektový přístup a tak veškeré entity, které R vytváří a se kterými manipuluje, lze chápat jako objekty. Sem patří proměnné, textové řetězce, veškeré datové objekty a dokonce i funkce. Pro zjištění třídy daného objektu lze použít funkci `class()`.

Základní datové typy v R neboli módy jsou `numeric`, `complex`, `logical`, `character` a `raw`, ale v R se vyskytují vždy jako vektory. Tedy i například samostatná numerická hodnota je chápána jako číselný vektor o délce 1.

Vektor

Pro vytvoření vektoru se nejčastěji používá funkce `c(...)`, která přijímá jako argumenty jednotlivé prvky výsledného vektoru a platí, že všechny prvky *musí* mít stejný

datový typ. Funkce `vector(<mode>, <length>)` vytváří vektor daného módu a zadané délky `length` s nulovými hodnotami. Pro zjištění datového typu vektoru slouží funkce `mode(<vektor>)`, další užitečná funkce `length(<vektor>)` vrací počet prvků ve vektoru.

K výpisu určité podmnožiny prvků vektoru můžeme použít hranatých závorek, `[]`. Obecně má pro vektor `x` tento příkaz tvar `x[<index>]`, kde `<index>` je vektor jednoho z následujících tvarů.

- Vektor přirozených čísel. Jedná se o vektor indexů, který nabývá hodnot z množiny $\{1; 2; \dots; length(x)\}$.
- Vektor záporných celých čísel. Vektorem záporných celých čísel `index` vymezujeme ty hodnoty vektoru `x`, které nemají být zahrnuty do výsledku. Všechny prvky vektoru `x`, kromě těch specifikovaných vektorem `index`, jsou do výsledné podoby vypsaný v jejich původním pořadí. Délka výsledného vektoru je `length(x) - length(index)`.
- Logický vektor. V tomto případě musí mít vektor `index` stejnou délku jako vektor `x`. Prvky odpovídající hodnotám `TRUE` vektoru `index` jsou vypisovány, zatímco prvky odpovídající hodnotám `FALSE` jsou vynechány. Například: `> x[c(TRUE, FALSE, TRUE, FALSE, FALSE)]` vrátí první a třetí prvek vektoru `x`.
- Vektor znakových řetězců. Tento způsob může být použit pouze v případě, kdy prvky vektoru mají názvy. V tom případě může být vektor `index` použit stejným způsobem jako v případě přirozených čísel v odrážce 1, řetězce ve vektoru `index` odpovídají názvům prvků vektoru `x`.

Přiřazení

Poněkud nestandardně má R několik způsobů přiřazení, jednak běžné `=`, ale také `<-`, respektive `->`. Použití tradičního znaku `=` je možné, ale v R není příliš obvyklé, běžnější je další způsob pomocí textové šipky, který navíc umožňuje psát cíl přiřazení vlevo či vpravo. Posledním způsobem přiřazení je samostatná funkce `assign(<cíl přiřazení>, <hodnota>)`. Několik příkladů vytvoření proměnné a přiřazení hodnoty:

```
# Několika způsoby vytvoříme proměnné x, y, z.  
> x=123  
> 123 -> y  
> z <- c(123)  
# Obsah proměnných je stejný.
```

Pro vypsání obsahu proměnné stačí napsat její jméno a potvrdit klávesou Enter. R zobrazuje jednak počáteční index vektoru v hranatých závorkách a následně hodnotu(y).

1.5.3 Základní matematické operace

R lze efektivně použít jako kalkulátor a uživatel má k dispozici obvyklé matematické operátory a funkce. Je však potřeba si uvědomit, že R pracuje s vektory i u běžné matematiky.

+	sčítání	–	odečítání
*	násobení	/	dělení
^	mocnina	%\%	celočíslné dělení
%%	zbytek po dělení (modulo)	%%*	maticové násobení

Tabulka 1. Základní operátory v R

Význam většiny běžných matematických funkcí, uvedených v tabulce 2, je zřejmý, pro ostatní je možné zjistit více informací v nápovědě. Navíc je zde několik funkcí pro práci s komplexními čísly : `Arg`, `Conj`, `Im`, `Re` a `Mod`.

<code>abs</code>	<code>cummax</code>	<code>log</code>	<code>sinh</code>
<code>acos</code>	<code>cummin</code>	<code>log10</code>	<code>sqrt</code>
<code>acosh</code>	<code>cumprod</code>	<code>log1p</code>	<code>sum.</code>
<code>asin</code>	<code>cumsum</code>	<code>log2</code>	<code>tan</code>
<code>asinh</code>	<code>digamma</code>	<code>max</code>	<code>tanh</code>
<code>atan</code>	<code>exp</code>	<code>min</code>	<code>trigamma</code>
<code>atanh</code>	<code>expm1</code>	<code>prod</code>	<code>trunc</code>
<code>ceiling</code>	<code>floor</code>	<code>range</code>	
<code>cos</code>	<code>gamma</code>	<code>sign</code>	
<code>cosh</code>	<code>lgamma</code>	<code>sin</code>	

Tabulka 2. Základní matematické funkce v R

1.5.4 Další datové typy

Faktor

Faktory jsou speciálním případem vektorů s nominálními nebo ordinálními daty. Jedná se o datovou strukturu, která umožňuje přiřadit názvy jednotlivým kategoriím. Na první pohled vypadají faktory podobně jako numerické a textové vektory, ale není tomu tak. Faktory v sobě navíc obsahují informaci `Levels`, jedná se o konečnou množinu hodnot, kterých kategorická proměnná může nabývat. Jednotlivé prvky `Levels`

jsou uspořádány podle jejich typu (numericky nebo abecedně), hodnoty NA zde ovšem nejsou zahrnuty. Je důležité si uvědomit, že prvky numerického faktoru nejsou interpretovány jako numerické hodnoty. Následující příklad ilustruje rozdíl mezi vektorem a faktorem.

```
> v=c(1,2,3,1,2,3)
> f=factor(v)
> f
[1] 1 2 3 1 2 3
Levels: 1 2 3

> mean(f)
[1] NA
Warning message:
In mean.default(f) : argument is not numeric or logical: returning NA

> mean(v)
[1] 2
```

Pole a matice

Matice je 2-dimenzionální datová struktura, která se skládá z řádků a sloupců. Stejně jako vektory, všechny prvky matice musí být stejného datového typu (numerický, komplexní, logický nebo textový), mohou rovněž obsahovat prvky s hodnotami NA. Pole je k-dimenzionální struktura, matice je tedy jejím speciálním typem pro $k = 2$. Nejjednodušším způsobem k vytvoření matice nebo pole je příkaz `dim()`, který vektor ve svém argumentu uspořádá po sloupcích do pole požadované dimenze. Další způsoby, jak vytvořit matice, případně pole, je použít příkazy `matrix()`, popř. `array()`:
> `matrix(v, 4, 5)` vytvoří matici o 4 řádcích a 5 sloupcích, prvky jsou skládány po sloupcích
> `matrix(v, 4, 5, byrow=TRUE)` vytvoří matici 4×5 , argument `byrow=TRUE` stanovuje, že prvky jsou skládány po řádcích.

K výpisu určité podmnožiny prvků matice či pole můžeme použít hranatých závorek `[]`. Obecně má pro pole `P` tento příkaz tvar `P[index1, index2, ..., indexN]`, kde `index1`, odkazuje na řádky, `index2` na sloupce a `index3` až `indexN` na ostatní dimenze. V případě, že některý z `indexů` není specifikován, v úvahu je brána celá délka příslušné dimenze.

Seznamy a datové tabulky

Seznam (`list`) je nejobecnější datová struktura v R, jeho vlastností je, že umí kombinovat soubory objektů do objektu většího rozsahu. Jedná se o datovou strukturu skládající se z posloupnosti objektů, kterým se říká složky. Každá složka může obsahovat objekt jakéhokoliv datového typu. Seznam tedy může obsahovat vektory různých

<code>%*%</code>	maticové násobení
<code>nrow()</code> , <code>ncol()</code>	počet řádků a sloupců matice/pole
<code>dim()</code>	dimenze pole
<code>t()</code>	transpozice matice
<code>diag()</code>	vypíše diagonálu matice
<code>diag(v)</code>	vytvoří diagonální matici se složkami vektoru <code>v</code> na diagonále
<code>diag(k)</code>	pro každé přirozené číslo <code>k</code> vygeneruje jednotkovou matici rozměru $k \times k$
<code>det()</code>	determinant matice
<code>eigen()</code>	výstupem je seznam o dvou položkách - <code>values</code> (vlastní čísla) a <code>vectors</code> (vlastní vektory).
<code>qr()</code>	QR rozklad. Jedním z výstupů je <code>i</code> hodnota matice, chceme-li zjistit pouze hodnotu matice, můžeme použít příkaz <code>qr()</code> <code>rank</code>
<code>svd()</code>	singulární rozklad trojúhelníkové matice
<code>norm()</code>	norma matice. Volitelným argumentem lze zvolit typ normy.
<code>sum(diag())</code>	stopa matice

Tabulka 3. Operátory a funkce pro manipulaci s maticemi

datových typů a délek, matice, pole, datové tabulky, funkce nebo jiné seznamy. Proto je použití seznamů velmi časté.

Funkce `list(nazev1=slozka1, ...)` slouží k vytvoření seznamu. Složky seznamu mohou být pojmenovány pomocí argumentů `nazev1=slozka1`, `nazev2=slozka2`,

K výpisu vlastností seznamu můžeme použít funkci `names()`, která vrací názvy složek seznamu. Funkce `dim()` a `contents()` u seznamu použít nemůžeme, můžeme je ovšem nahradit funkcemi `length()`, která vrací počet složek seznamu, a `str()`, která vypisuje vnitřní strukturu seznamu.

K vypsání podmnožiny seznamu můžeme použít jednoduchých `[]` nebo dvojitých `[[]]` hranatých závorek. Jednoduchými závorkami uvádíme, které složky seznamu chceme získat. Jestliže jednotlivé složky seznamu nejsou pojmenovány, požadovanou složku specifikujeme jejím číslem. K výpisu více složek můžeme použít operátor `:` nebo funkce `c()`. Jestliže jsou složky seznamu pojmenovány, požadované prvky specifikujeme jejich názvy v uvozovkách. Podmnožina seznamu vytvořená pomocí jednoduchých hranatých závorek je opět typu seznam. Naopak, příkaz pro vytváření podmnožiny pomocí dvojitých hranatých závorek vrací objekt takového typu, jakým byl při definování seznamu. V tomto případě je na každou složku odkazováno jednotlivě, nepoužívá se proto operátor `:` ani funkce `c()`. Stejně jako u jednoduchých hranatých závorek, na každou složku je odkazováno jejím číslem, má-li požadovaná složka název, můžeme na ni odkazovat jejím názvem v uvozovkách nebo můžeme použít operátor `$`

Datová tabulka (také datový rámeček, `data.frame`) je dvourozměrná struktura, která slouží k uchování souboru dat. Terminologií statistiky je soubor dat množina proměnných (sloupců), které jsou pozorovány na množství případů (řádky). Jednotlivé sloupce mohou být různých datových typů, ale prvky každého sloupce musí být stejného datového typu. Datové tabulky mohou být považovány i za zobecněné matice.

Na tomto místě je vhodné si uvědomit, že datová tabulka je speciálním případem seznamu. Lze říci, že datová tabulka je seznam, jehož složky jsou vektory stejné délky a odpovídající pozice vyjadřují stejné případy. Datovou tabulku vytvoříme příkazem `data.frame()`. Argumenty ve tvaru `nazev1=vektor1`, `nazev2=vektor2` atd., specifikujeme názvy sloupců (proměnných) a jejich hodnoty, názvy sloupců jsou nepovinné, stačí zadat pouze hodnoty. Argument `row.names` specifikuje názvy případů (implicitní nastavení `NULL` případy čísluje). Argument `check.names` s implicitním nastavením `TRUE` kontroluje, zda jsou názvy proměnných syntakticky správné a zda se neopakují, v případě duplikací se stará o jejich přejmenování.

Funkce `dim()`, `names()` a `contents()` slouží k výpisu vlastností datové tabulky. Funkce `dim()` vypisuje dimenze tabulky dat, funkce `names()` zobrazuje názvy proměnných.

Řádky nebo sloupce datové tabulky lze získat, stejně jako u seznamu, pomocí `[]`, `[[]]` nebo operátoru `$`. K výběru podmnožiny datové tabulky slouží i příkaz `subset(x,)`. Argument `x` specifikuje datovou tabulku, z níž podmnožinu vybíráme. Argument `subset` specifikuje řádky vyhovující dané podmínce, přičemž hodnoty `NA` jsou brány jako `FALSE`. Argument `select` specifikuje sloupce, které chceme vypsat, můžeme použít funkce `c()`, operátoru `:` i operátoru `-` pro vynechání složek. Na rozdíl od operátorů `[]`, `[[]]`, funkce `subset()` vždy vrací tabulku dat, i když má jen jeden řádek nebo sloupec. K tomu, aby vrátila jen jednoduchý vektor, musíme za vlastní definici podmnožiny datové tabulky použít operátor `$`, za nímž následuje název sloupce.

1.5.5 Práce s datovými soubory

R disponuje funkcemi pro načítání dat a jejich následné ukládání do souboru. Při práci se soubory je důležité znát pracovní adresář (working directory). Funkce `getwd()` vrátí jeho aktuální umístění a funkcí `setwd(dir)` můžeme pracovní adresář přepnout do jiného umístění.

Libovolný objekt lze uložit do souboru pomocí funkce `save(objekt, file="soubor"`

a opětovně načíst funkcí `load("soubor")`. Formát souboru je v tomto případě binární (komprimovaný) a je určen pouze pro použití z R.

Pokud chceme pracovat s textovým formátem souboru a mít tak možnost jej takto editovat nebo i vytvářet, potřebujeme funkce jiné. Častým typem objektu je datový rámec (`data.frame`), který můžeme uložit do souboru funkcí `write.table(promenna, file="soubor")` a opětovně načíst funkcí `promenna <- read.table("soubor")`.

Pro vektor nebo matici lze použít funkce `write()` pro uložení a `scan()` pro načtení. Funkce `scan()` se používá i pro čtení uživatelského vstupu z klávesnice.

R umí načíst i soubory jiných formátů např. Excel, SPSS atd., pomocí rozšiřujících balíčků. Obecně, funkce pro práci se soubory mají celou řadu parametrů, kterými lze nastavit chování při čtení a zápisu dat. Pomocí těchto parametrů lze například přeskakovat prázdné řádky, zvolit si znak, který reprezentuje oddělovač sloupců nebo naopak nastavit pevnou šířku sloupců.

1.5.6 Grafy

R má značné množství funkcí pro grafické zobrazování výsledků statistických a jiných analýz. Grafické funkce lze rozdělit do čtyř kategorií. První tzv. high level funkce vytvoří samotný graf. Druhá kategorie funkcí, tzv. low level, jsou metody, které mohou již existující grafy rozšiřovat, například doplňovat dodatečné elementy (popisy, typy čar, tečky atd.). Třetí skupina funkcí grafické parametry umí pracovat s jednotlivými prvky grafu z hlediska jejich barev, tvaru, rozměru, fontu atd. Poslední množina funkcí tzv. interaktivní funkce umožňuje pracovat s již vytvořeným grafem interaktivně.

Kromě toho, R obsahuje ještě mnoho dalších funkcí pro tvorbu grafů. Některé jsou obsaženy ve speciálních balíčcích např. balíček `lattice` a `grid`.

High level funkce

High level funkce zobrazují kompletní graf. R nabízí velké množství, jak samotných funkcí, tak i atributů těchto funkcí, které dělají grafické funkce mnohem flexibilnější. Základní funkcí pro tvorbu grafu je funkce `plot(x, y, xlim, ylim, type=p, main, xlab, ylab, ...)`. Argumenty `x` resp. `y` reprezentují x-ové resp. y-ové souřadnice zobrazovaných dat. Jednotlivé třídy objektu lze vynést do grafu pouze s určitými parametry. Pokud je uvedena pouze jedna souřadnice, tedy atributem je jeden vektor, pak druhou souřadnicí je automaticky index prvku v objektu, který je vynesena na x-ovou osu. Osy vektoru komplexního jsou jeho reálné a imaginární části. Matici lze graficky

zobrazit pouze, pokud má jen dva sloupce, kdy první sloupec je vynesena na x-ovou a druhý sloupec na y-ovou osu.

Další funkce pro vytvoření různých typů grafů jsou například `hist()`, `dotchart()`, `pairs()`, `barplot()`, `pie()` a další.

Low level funkce

Low-level funkce doplňují do již existujícího grafu různé prvky. Mohou to být body, přímky, různé formy popisu atd.

- `points(x,y)` - vykreslí do existujícího grafu body o souřadnicích $[x,y]$ nebo o souřadnicích $[index,x]$, pokud je zadán jen jeden vstupní atribut
- `lines(x,y)` - udělá to samé jako funkce `points()` a spojí zobrazené body přímkou
- `text(x, y, labels, ...)` - na stávající body dodá jejich popis, často je užívána tato funkce ve spojení s funkcí `plot(x,y,type=n)`, kdy jsou vykresleny pouze osy a pak dodá na souřadnice bodu jejich popisy
- `mtext(text,side,line)` - do grafu na zvolené místo určené parametry `side` a `line` doplní zadaný text
- `segments(x0, y0, x1, y1)` - nakreslí přímku vycházející z bodu $[x_0,y_0]$ do bodu $[x_1,y_1]$
- `abline(h=x)` - nakreslí vodorovné přímky procházející x-ovými souřadnicemi
- `abline(v=x)` - zobrazí svislé přímky procházející x-ovými souřadnicemi
- `arrows(x0, y0, x1, y1, angle, code)` - funkce pro kreslení šipek; počáteční a koncový bod šipky je zadán souřadnicemi; úhel, který svírají postranní krátké úsečky s dlouhou úsečkou šipky, je zadán atributem `angle`; atribut `code` udává směr šipky
- `polygon(x, y, ...)` - vykreslí polygon o zadaných souřadnicích
- `title()` - dokreslí do stávajícího grafu název atributem `main` nebo popis grafu atributem `sub`

Grafické parametry

Téměř každý vykreslený prvek grafu lze upravit z hlediska vizuálního. To znamená, že lze měnit rozměry, tvary, barvy jednotlivých elementů grafu. Grafické parametry lze nastavit pomocí funkce `par()`. Vstupních atributů této funkce je mnoho a zde

budou vyjmenovány pouze ty základní. Tyto atributy lze také použít přímo ve spojení s funkcemi pro vykreslení grafu. Avšak ve spojení s funkcí `par()` mají trvalý charakter, zatímco jako atribut některé high-level funkce mění grafické prostředí pouze pro daný graf.

- `adj` - zarovnání textu, 0-vlevo, 1-střed, 2-vpravo
- `bg` - barva pozadí, yellow, red, blue. .
- `bty` - specifikuje ohraničení grafu
- `cex` - udává velikost písma
- `col` - barva jednotlivých prvků; je zadávána stejně jako atribut `cex`
- `font` - font písma; nabývá hodnot 1-normal, 2-italic, 3-bold
- `lty` - typ čáry jednotlivých prvků v grafu
- `lwd` - tloušťka čáry
- `mar` - udává odsazení os od okraje grafu v pořadí dolní, levý, horní a pravý okraj
- `pch` - typ značky pro bod grafu; 1-kružnice, 2-trojúhelník, 3-křížek, atd.
- `mfc`, `mfrow` - umožní umístit více grafů vedle sebe

Interaktivní funkce

Interaktivní funkce pracují s již existujícím grafem. Umožňují dodávat do grafu nebo z grafu přebírat specifické informace interaktivně, tedy použitím myši. Funkce `locator(n,type)` odečítá z grafu souřadnice. Užívá se především, pokud se zjišťují souřadnice dodatečného prvku grafu. Kliknutím levým tlačítkem myši na zvolené místo grafu funkce zaznamená souřadnice (nebo více souřadnic) a vrátí jejich hodnoty. Další funkcí pracující interaktivně je funkce `identify(x,y,labels)`. Slouží pro označení prvku v grafu jmény.

Příklady některých typů grafů v R

Následuje krátký souhrnný příklad některých možností tvorby grafů diskutovaných v této podkapitole. Výsledná kombinace grafů je na obrázku 3 na straně 27.

```
par(mfrow=c(2,2))                                # matice grafů 2x2

trend <- c(2,4,5,4,7,5,8,11,15,11,12,18)
boxplot(trend)                                    # 1. graf
```

```

title("boxplot(trend)")

plot(trend, type="o", col="blue", main="plot()") # 2. graf
legend("topleft", "Prodej", cex=0.6           # popisek
      , bty="n", fill="blue")
abline(lsfrit(1:12,trend), col="blue")        # linearni regrese

rnd <- rnorm(1000)                            # 1000 náhodných hodnot
hist( rnd, col = "yellow", freq = FALSE      # 3. graf (histogram)
      , br = 20, main="Histogram pro rnorm(1000)")
lines(density(rnd), col = "red", lwd = 2)

podily <- c(10,20,70)
firmy <- c("Firma_A","Firma_B","Firma_X")
barvy <- c("purple", "violetred1", "green3")
pie( podily, labels=paste(podily,"%")
     , main="Podíl na trhu", col = barvy)    # 4. graf (pie)
legend("topleft", firmy, cex=0.8, bty="n", fill=barvy)

```

1.5.7 Řízení běhu programu

R není pouze aplikace pro statistické výpočty, ale je zároveň i programovacím jazykem, v němž lze psát vlastní programy. Programování v R není složité a programovací jazyk je navrhnut tak, aby i běžný uživatel byl schopen vytvořit vlastní program či uživatelské funkce. Řízení běhu programu, tedy podmínky a cykly, lze v R naprogramovat analogicky jako v jiných programovacích jazycích, syntaxe je zde celkem typická.

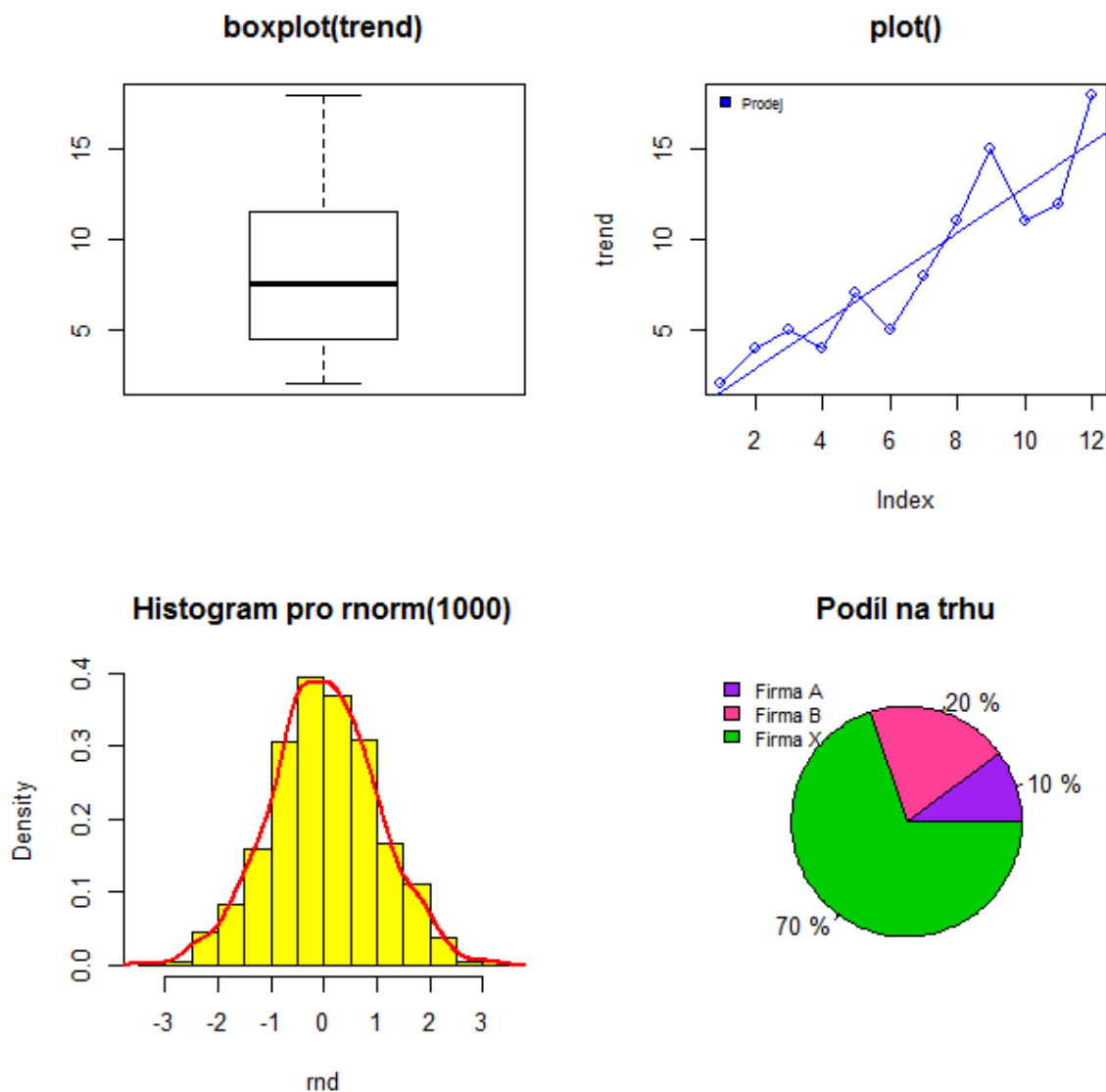
Výrazy v jazyku R lze seskupit pomocí složených závorek: { výraz1 ; výraz2 ; ... }, jednotlivé výrazy jsou odděleny středníkem.

Příkaz `if` má klasický formát: `if (podmínka) výraz1 else výraz2`, část `else` je nepovinná.

Příkazy cyklu jsou:

- `for (proměnná1 in výraz1) výraz2`, kde `výraz1` je typu vektor a proměnná `proměnná1` je v průběhu cyklu nastavována podle jednotlivých prvků vektoru,
- `while (podmínka) výraz`, kde `výraz` je vyhodnocován dokud není splněna podmínka `podmínka`,
- `repeat { vnořený kód }`, kód ve složených závorkách je vyhodnocován opakovaně, ukončení tohoto cyklu provede příkaz `break`.

Příkaz `break` provádí ukončení cyklu kteréhokoli typu, příkaz `next` provede ukončení jednoho cyklu a přechod k dalšímu.



Obrázek 3. Příklady některých typů grafů v R

1.5.8 Uživatelem definované funkce

Tvorba vlastních funkcí běžnou součástí programů v R. Vlastní funkce může obsahovat neomezené množství vstupních argumentů, ale také nemusí mít žádný. Funkci lze vytvořit přímo v příkazovém řádku a pak ji uložit do svého pracovního prostředí, nebo ji lze zapsat do souboru stejně jako program a vyvolat ji funkcí `source()`. Dále lze vytvořenou funkci používat, jako kteroukoliv jinou.

```
> funkce1 <- function() print(x)
> x <- 3
> funkce1()
[1] 3
```

Na příkladu funkce `funkce1()` je ukázána jedna potenciálně nebezpečná vlastnost jazyka R. V jazyce R totiž není nezbytně nutné deklarovat proměnné uvnitř funkce. Při vyhodnocování funkce R používá pravidlo „lexical scoping“, které rozhodne, zda je objekt lokální nebo globální proměnnou. Objekt `x` není definován uvnitř funkce `funkce1()`, proto R hledá v uzavřeném prostředí objekt s názvem `x` a vytiskne jeho hodnotu. V případě, že by objekt `x` nebyl nalezen ani v globálním prostředí, funkce se ukončí s chybovým hlášením `Error in print(x) : object 'x' not found`.

1.5.9 Balíčky

Všechny funkce jsou v R ukládány a distribuovány pomocí balíčků (označované také jako `package`, `library` nebo knihovna). U balíčků musíme rozlišovat dva stavy, jednak zda je balík nainstalován a pak zda je balík načten. Pro zjištění, které balíky jsou v dané instalaci R nainstalovány a tedy k dispozici pro načtení, použijeme funkci `library()`. Pro zjištění, které balíky máme načteny v aktuální instanci R, použijeme funkci `search()`.

K načtení balíku (což u některých balíčků způsobí i jeho spuštění) použijeme funkci `library(jméno_balíku)`. Instalace balíku se provádí funkcí `install.packages()` a případná aktualizace novější verzí pak funkcí `update.packages()`.

2 GRAFICKÁ UŽIVATELSKÁ ROZHRAŇÍ PRO R

Prostředí R poskytuje uživateli rozhraní na úrovni příkazového řádku (CLI - Command Line Interface) a pro řadu pokročilých uživatelů je toto základní rozhraní dostačující. Výhodou je, že tento způsob umožňuje přímé řízení postupu výpočtů a je velmi flexibilní. Zároveň však předpokládá dobrou znalost jazyka a určité zkušenosti získané praktickým používáním prostředí.

Na druhou stranu, pro začínajícího nebo občasného uživatele je příkazový řádek spíše nevýhodou. Rychlost a snadnost proniknutí do R je obvykle nižší než s grafickým uživatelským rozhraním, ačkoli se předpokládá, že vynaložené úsilí je později zúročeno lepším pochopením problematiky statistické analýzy. Jednou spuštěné příkazy lze snadno ukládat do skriptů a tak lze vytvářet opakovatelné postupy, které mohou výrazně usnadnit a urychlit komplikovanější analýzy.

Pokud bychom chtěli porovnat volně šiřitelné R a komerční TIBCO Spotfire S+ [6] (oba používají prakticky stejný jazyk), pak platí, že právě v grafickém uživatelském rozhraní je zde největší rozdíl, protože S+ implementuje velmi rozsáhlé a propracované GUI. Určitá část uživatelů R by jistě přivítala existenci kvalitního GUI, které by pro ně bylo bezpochyby přínosem. To platí hlavně pro příležitostné uživatele, pro ty, pro které je funkčnost poskytovaná R jen jedním z kroků a zaměřují se na komplexnější workflow činností, možná také pro učitele v rámci výuky a zcela jistě i pro další skupiny uživatelů.

Existuje několik projektů, které se buď věnují vývoji kompletního GUI nebo vyvíjejí komponenty, které lze následně použít pro vytvoření GUI pro R. V následující části bude uveden jejich přehled.

2.1 Cíle a kritéria porovnání

Cílem této kapitoly je uvést přehled existujících grafických uživatelských rozhraní pro systém R a zhodnotit jejich hlavní klady a zápory. Pramenem informací je jednak [4] (WWW stránky The R Project), kde je udržován, sice ne zcela aktuální, seznam GUI projektů s vazbou na R. Dalším zdrojem jsou výsledky vyhledávání pomocí internetových vyhledávačů (Google a pod.). Dále to jsou informace na internetových stránkách jednotlivých projektů, dostupná dokumentace k produktu, výsledky získané testováním nainstalovaného produktu a případně zkoumáním dalších zdrojů, jako jsou

zdrojové kódy aplikace. Důležitým faktorem je také aktuálnost produktu, aktivita vývoje a vývojového cyklu a podobně.

Dalším cílem tohoto přehledu je také identifikovat případný vhodný softwarový produkt, který by mohl usnadnit dosažení cílů této práce. Takovýto produkt musí obsahovat alespoň část požadované funkcionality a zároveň musí umožnit doplnění chybějící funkčnosti a to jak dodatečnou konfigurací nebo doprogramováním. Licenční podmínky produktu musí umožňovat produkt upravovat nebo rozšiřovat. Mělo by tedy jít o kategorii Open Source s licenci GPL nebo podobnou.

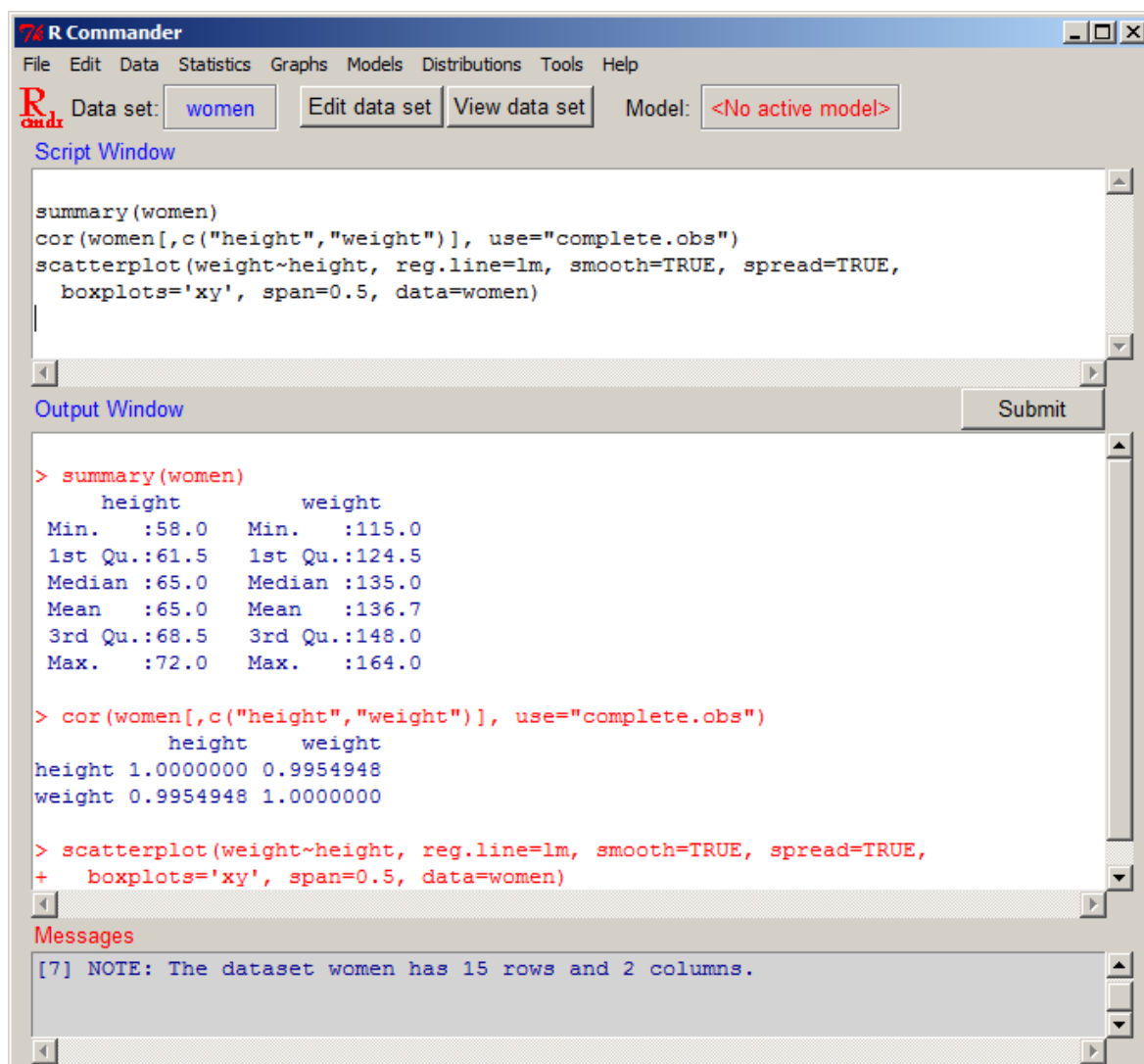
2.2 R Commander

2.2.1 Charakteristika

R Commander je multiplatformní grafické uživatelské rozhraní pro R, které zpřístupňuje velké množství statistických funkcí prostřednictvím série několikaúrovňových nabídek a dialogů. Ačkoli původní autorův záměr byl zpřístupnit základní statisticky orientovanou funkčnost a to zejména pro potřeby v rámci výuky úvodních kurzů statistiky, pozdější rozšiřování vedlo až k dnešní podobě, která zahrnuje i pokročilé statistické metody.

Spuštění R Commanderu se provádí z běžícího prostředí R a to jednak automaticky, při načtení balíčku `Rcmdr: library(Rcmdr)` a nebo, pokud již byl balík dříve načten, voláním funkce `Commander()`. Pracujeme tedy nejméně s dvěma okny, základní R konzolou a samostatným oknem R Commanderu. Toto okno (obr. 4) má přehledné členění na hlavní nabídku - menu, dále tzv. „Script Window“ - textová oblast kam se zaznamenávají příkazy a volání funkcí jazyka R, které jsou generovány při spouštění dialogů, pod ním je pak tzv. „Output Window“, které obsahuje textový výstup jednotlivých spuštěných funkcí a poslední část okna obsahuje „Messages“ - případná chybová a varovná hlášení generovaná spuštěnými funkcemi.

Program v jednom okamžiku pracuje s jednou zvolenou datovou oblastí - „Data set“, což je označení pro uživatelem vybraný datový rámeček (datový typ `data.frame`). Nad touto datovou oblastí program následně pracuje, dialogová okna tedy obsahují již předvyplněné nabídky s konkrétními názvy proměnných a uživatel pouze volí, které se mají pro danou funkci použít. Často stačí pouze několik kliknutí k dosažení určitého cíle, uživatel je tedy veden intuitivně a není potřeba znát přesná jména funkcí a jejich parametrů.



Obrázek 4. R Commander: Hlavní okno aplikace

Na druhou stranu uživatel získává záznam o provedených akcích ve „Script Window“ a může snadno později celou sérii kroků opakovat, případně modifikovat, bez nutnosti procházet celé menu a odpovídající dialogy znovu. Navíc, každý dialog obsahuje tlačítko pro vyvolání nápovědy v odpovídajícím kontextu.

Zkrácený přehled funkcí dostupných z menu

```
File - ...
Edit - ...
Data - New data set
      |- Load data set
      |- Merge data sets
      |- Import data - from text file, clipboard, URL
                        - from SPSS data set, Minitab data set, STATA data set
                        - from Excel, Access, or dBase data set
      |- Data in packages - (List/Read data sets)
      |- Active data set - ...
      |- Manage variables in active data set - ...
```

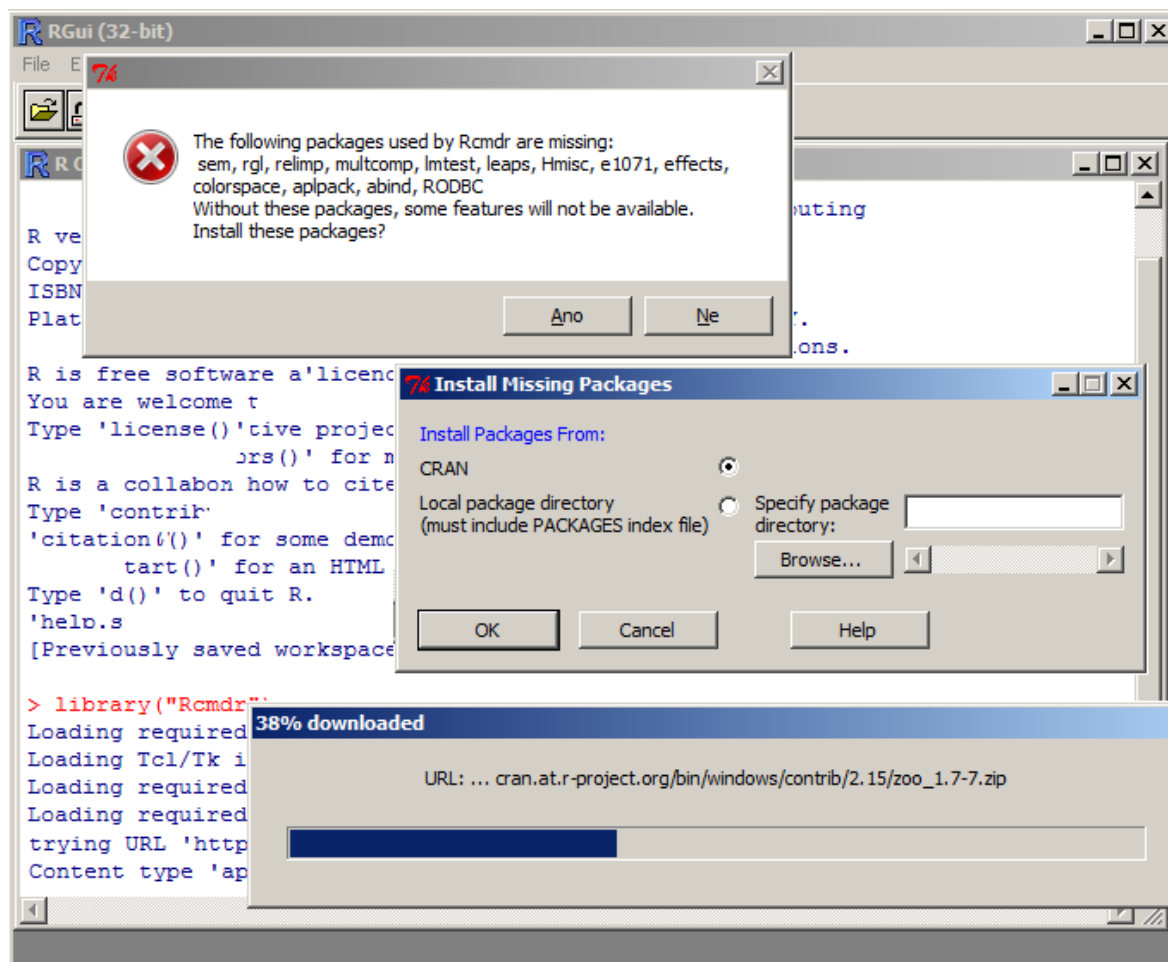
```
Statistics
|- Summaries - ...
|- Contingency Tables - ...
|- Means - ...
|- Proportions - ...
|- Variances - ...
|- Nonparametric tests - ...
|- Dimensional analysis - ...
|- Fit models - ...
Graphs
|- Color palette
|- Index plot
|- Histogram
|- Stem-and-leaf display
|- Boxplot
|- Quantile-comparison plot
|- Scatterplot
|- Scatterplot matrix
|- Line graph
|- XY conditioning plot
|- Plot of means
|- Strip chart
|- Bar graph
|- Pie chart
|- 3D graph - ...
|- Save graph to file - ...
Models
|- Select active model
|- Summarize model
|- Add observation statistics to data
|- Confidence intervals
|- Akaike Information Criterion (AIC)
|- Bayesian Information Criterion (BIC)
|- Stepwise model selection
|- Subset model selection
|- Hypothesis tests - ...
|- Numerical diagnostics - ...
|- Graphs - ...
Distributions
|- Continuous distributions
|- Normal distribution
|- Normal quantiles
|- Discrete distributions
|- Binomial distribution
|- Binomial quantiles
Tools - Load package(s) / plug-in(s)
|- Options
Help - ...
```

2.2.2 Technické informace

R Commander je naprogramován v jazyku R a jako okenní toolkit používá balík `tcltk` [9], což je v podstatě rozhraní mezi R a Tcl/Tk. Tcl/Tk je programovací jazyk Tcl obohacený o Tk - knihovnu komponent pro tvorbu GUI (grafický toolkit); jde o klasický vývojový nástroj, existující již poměrně dlouho, který je multiplatformní

a stále se postupně vyvíjí. Více informací o Tcl/Tk a o podrobnostech jeho použití v rámci prostředí R lze nalézt např. v [11].

Balík je dostupný včetně zdrojových kódů a je poskytován pod licencí GNU GPL v2.0. R Commander je dobře připraven na případnou lokalizaci do národního jazyka, stačí přeložit texty soustředěné do jednoho souboru ve formátu .po – „Portable Object“ (GNU gettext).



Obrázek 5. Dialogová okna při instalaci R Commanderu

2.2.3 Instalace

Instalaci lze provést ve Windows z menu Packages -> Install package(s) nebo z R konzoly příkazem `> install.packages("Rcmdr")`. Balík má několik závislostí na další balíky, které je nutné nechat doinstalovat – stačí spustit příkaz `library(Rcmdr)` z příkazového řádku R a případné dodatečné balíčky budou doinstalovány. Příklad dialogových oken při instalaci dodatečných balíčků do R prostředí je na obrázku 5.

V linuxu lze postupovat obdobně, ale například v Ubuntu linuxu můžeme použít přímo příkaz operačního systému:

```
# apt-get install r-cran-rcmdr.
```

2.2.4 Hodnocení

Celkově lze hodnotit R Commander velmi kladně. Jeho použití je jednoznačně předurčeno pro studenty a začínající uživatele a jeho používání pomůže jak v oblasti ovládnutí R, tak v orientaci v bohaté nabídce statistických funkcí. A právě zde spočívá největší přínos R Commanderu: přehledné zpřístupnění statistických funkcí.

Bohužel není k dispozici český překlad a tak si uživatel musí poradit s výchozí anglickou verzí. Celý balík je dostupný i se zdrojovými kódy, ale z technického hlediska jde o vnitřně velmi provázaný programový kód a pokus o zásadnější redukci kódu (s cílem později doplnit vlastní) se ukázal jako obtížně realizovatelný.

Klady

- Multiplatformní
- Velké množství funkcí dostupných prostřednictvím menu
- Záznam akcí do skriptu pro pozdější znovupoužití
- Rozšiřitelný, alespoň do určité míry

Zápory

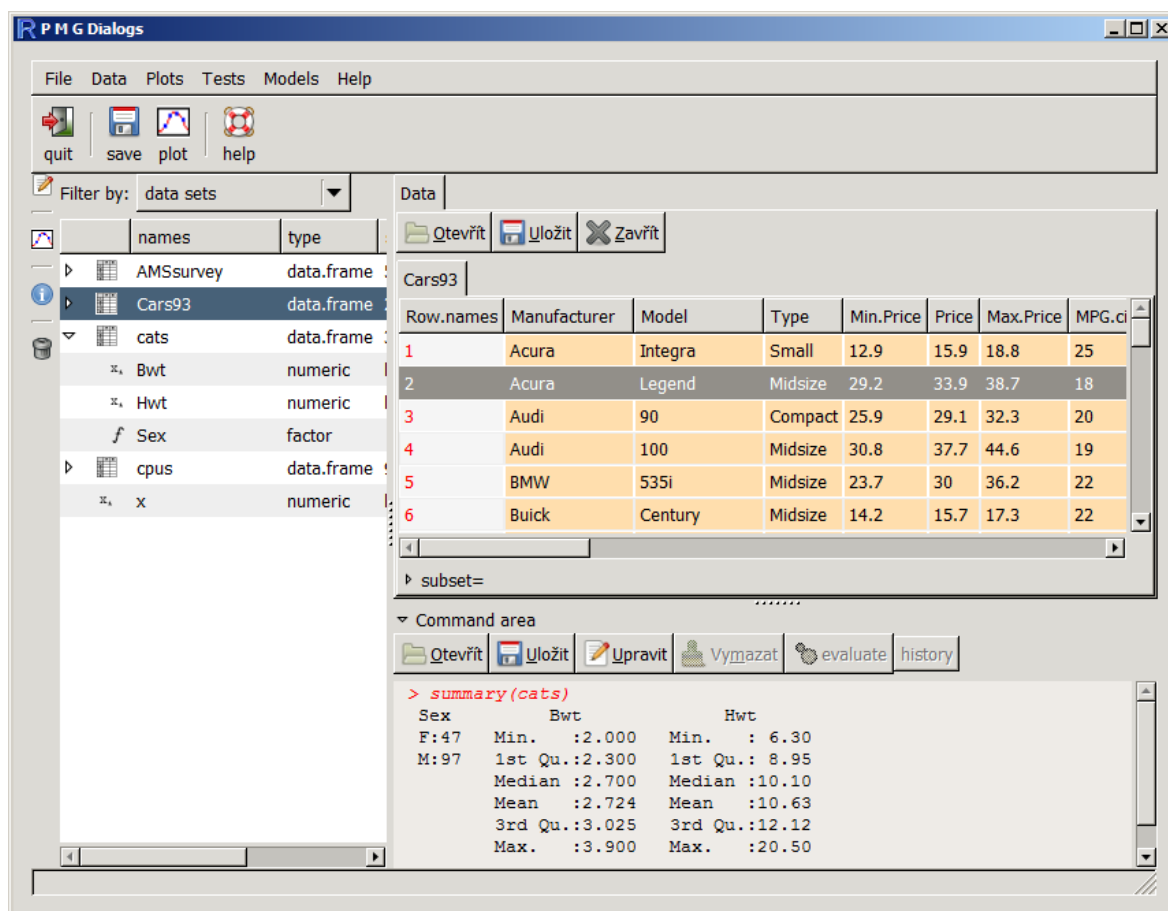
- Není k dispozici česká jazyková varianta
- Nedostatečná podpora editace dat
- Pracuje jako samostatné okno, nižší stupeň integrace s R

2.3 Poor Man's GUI – pmg

2.3.1 Charakteristika

Poor Man's GUI – pmg je do určité míry obdobou R Commanderu naprogramovanou v jazyce R a s pomocí grafického toolkitu RGtk2. Opět tedy jde o multiplatformní GUI, které si klade za cíl zpřístupnit statistické funkce R v rozsahu potřeb základního statistického kurzu a je tedy primárně určeno pro výuku. Použití RGtk2 v aplikaci je

na první pohled znatelné a umožňuje mimo jiné používat některé modernější funkce v rámci uživatelského rozhraní, například Drag&Drop.



Obrázek 6. Poor Man's GUI – pmg

Míra integrace Poor Man's GUI s vlastním R prostředím je podobná jako u R Commanderu, tedy uživatel používá dvě okna, grafický výstup není do pmg integrován.

2.3.2 Technické informace a instalace

Instalace se provede opět přímo z R konzoly pomocí příkazu:

```
> install.packages("pmg", dep=TRUE)
```

A samotné spuštění pak příkazem: `> library(pmg)`, nebo pokud již byl balík pmg načten voláním funkce: `> pmg()`.

Použití GTK+ způsobuje spojené určité problémy na Windows (např. mizení okna při jeho přesunu na obrazovce). Balík je dostupný včetně zdrojových kódů. Z kódu vyplývá, že není snadno lokalizovatelný. Vývoj není nijak aktivní, jestli se ještě vůbec vyvíjí.

2.3.3 Hodnocení

Vzhledem k obdobnému zaměření se nabízí porovnání s R Commanderem, ale výsledek není pro Poor Man's GUI příliš pozitivní. Orientace v menu je obtížnější a to i přesto, že nabídka statistických funkcí není tak bohatá. Opět není k dispozici český překlad, ale navíc není ani naděje, že by lokalizace mohla snadno vzniknout.

Klady

- Multiplatformní
- Poměrně velké množství funkcí dostupných prostřednictvím menu (ale menší než u R Commanderu)
- Lepší podpora editace dat než má R Commander

Zápory

- Není k dispozici česká jazyková varianta
- Stěží rozšiřitelný, vývoj není příliš aktivní
- Záznam akcí do skriptu není dobře vyřešen
- Pracuje jako samostatné okno, nižší stupeň integrace s R

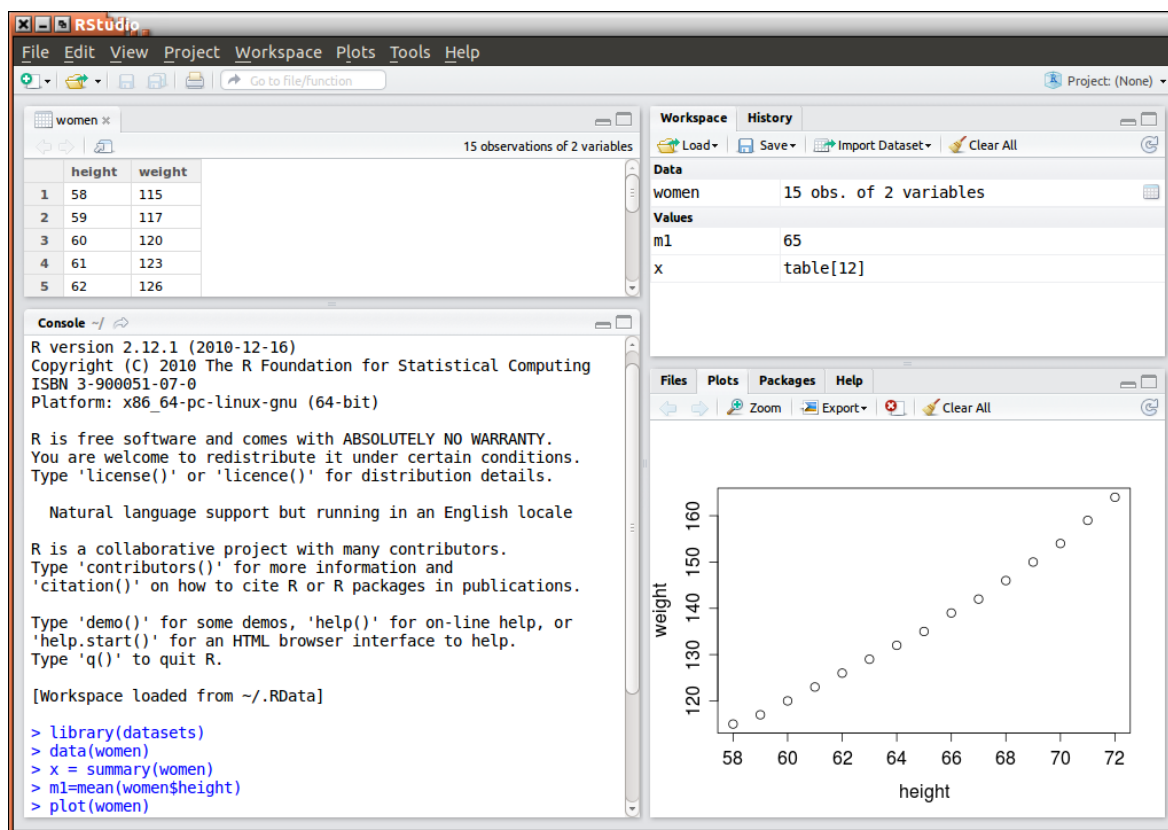
2.4 RStudio

2.4.1 Charakteristika

RStudio je multiplatformní integrované vývojové prostředí (IDE) pro vývoj, spouštění a ladění skriptů psaných pro R. Jde o nástroj, který je zaměřen spíše na pokročilé uživatele prostředí R, kteří se orientují na programování v jazyce R. RStudio je plně integrováno s R (a to včetně konzoly a dokonce grafického výstupu), při práci s ním již není potřeba zvlášť spouštět prostředí R a uživatel pracuje s jedním oknem aplikace (obr. 7 [13]).

2.4.2 Technické informace

RStudio je volně šířený software včetně zdrojových kódů (pod licencí GNU Affero General Public License - varianta GPL), který běží na MS Windows, Linuxu a Mac OS



Obrázek 7. RStudio v linuxu

X. RStudio je programováno převážně v jazyce C++ a za použití okenního toolkitu Qt (Qt framework).

2.4.3 Instalace

Pro stažení instalačního balíku je potřeba navštívit domovské stránky projektu [13]. RStudio není obsaženo v instalačních repositářích Ubuntu, proto je i v tomto případě provést instalaci manuálně, například takto:

```
$ wget http://download1.rstudio.org/rstudio-0.95.265-i386.deb
```

```
$ dpkg -i rstudio-0.95.265-i386.deb
```

Samotné RStudio neobsahuje instalaci prostředí R a proto je nutné nainstalovat R samostatně (pokud možno do standardního adresáře).

2.4.4 Hodnocení

RStudio je skvělý pomocník programátora v jazyku R. Tímto lze stručně RStudio charakterizovat a zároveň naznačit jeho hlavní klady a zápory. Zvýraznění syntaxe jazyka R, podpora ladění, doplňování kódu („code completion“) pro jména funkcí, pro-

měnných i parametrů funkcí nemá zřejmě mezi editory pro R konkurenci. Proto uživatel – programátor, bude s RStudiem pravděpodobně velmi spokojen. Naopak uživatel očekávající podporu pro statistické funkce obsažené v R může být poněkud zklamán, protože RStudio příliš v této oblasti nenabízí.

Klady

- Multiplatformní
- Vysoký stupeň integrace s R
- Komfortní psaní R kódu

Zápory

- Není „statisticky“ orientován
- Není k dispozici česká jazyková varianta
- Obtížně rozšiřitelný

2.5 RKward

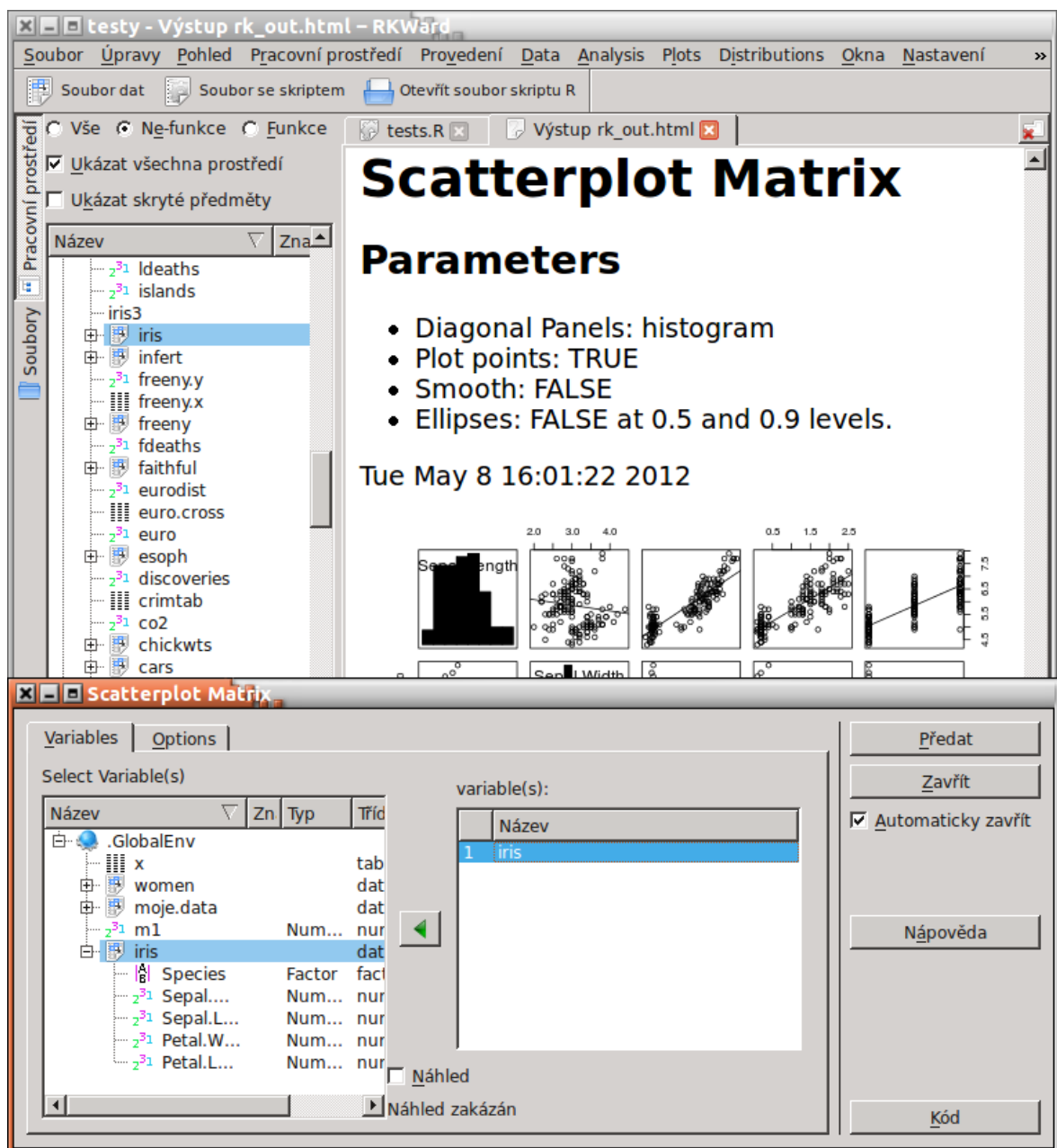
2.5.1 Charakteristika

RKward je kompletní GUI prostředí pro R. Nabízí vysokou míru integrace s prostředím R a jeho zaměření je opravdu široké. Podporuje snad všechny potřebné činnosti, od komfortního vstupu, importu, prohlížení a editace dat, přes přístup k celé řadě statistických funkcí prostřednictvím intuitivního GUI, až po pohodlnou práci s R kódem a to včetně zvýraznění syntaxe a doplňování kódu („code completion“). Aplikace je z části k dispozici i v češtině, což je v oblasti software pro R zcela výjimečné.

Grafický výstup z R je také integrován přímo do prostředí RKwardu. Navíc je zde k dispozici automatický výstup kombinovaného textu a grafiky ve formátu HTML, což může sloužit jako základ pro tvorbu efektních, snadno publikovatelných uživatelských reportů.

2.5.2 Technické informace a instalace

RKward je vyvíjen pro (původně linuxové) desktopové prostředí KDE a to převážně v jazyce C++ a za použití okenního toolkitu Qt (Qt framework). Ačkoli je KDE od



Obrázek 8. RKward v linuxu

roku 2009 portováno i na platformu MS Windows, právě ze závislosti na KDE plynou problémy s dostupností a stabilitou RKward v MS Windows.

2.5.3 Hodnocení

RKward obsahuje některé prvky, které se u jiných GUI pro R nevyskytují (přímý výstup do HTML). Také široký záběr aplikace zřejmě nemá konkurenci. Možná právě tento velký rozsah je důvodem nestability chování aplikace, i když se to většinou týká pouze vizuální stránky aplikace (drobné problémy s překreslováním objektů a podobně).

Otázkou je budoucnost vývoje takto rozsáhlého programu, podle intenzity práce se zdrojovými kódy se zdá, že vývojářská komunita není nijak silná. Celkově aplikace působí trochu nedotaženým dojmem, alespoň v detailech, které trochu důkladnější testování odhalí.

Klady

- Nabízí téměř vše, co lze od GUI pro R očekávat
- Alespoň částečná podpora češtiny

Zápory

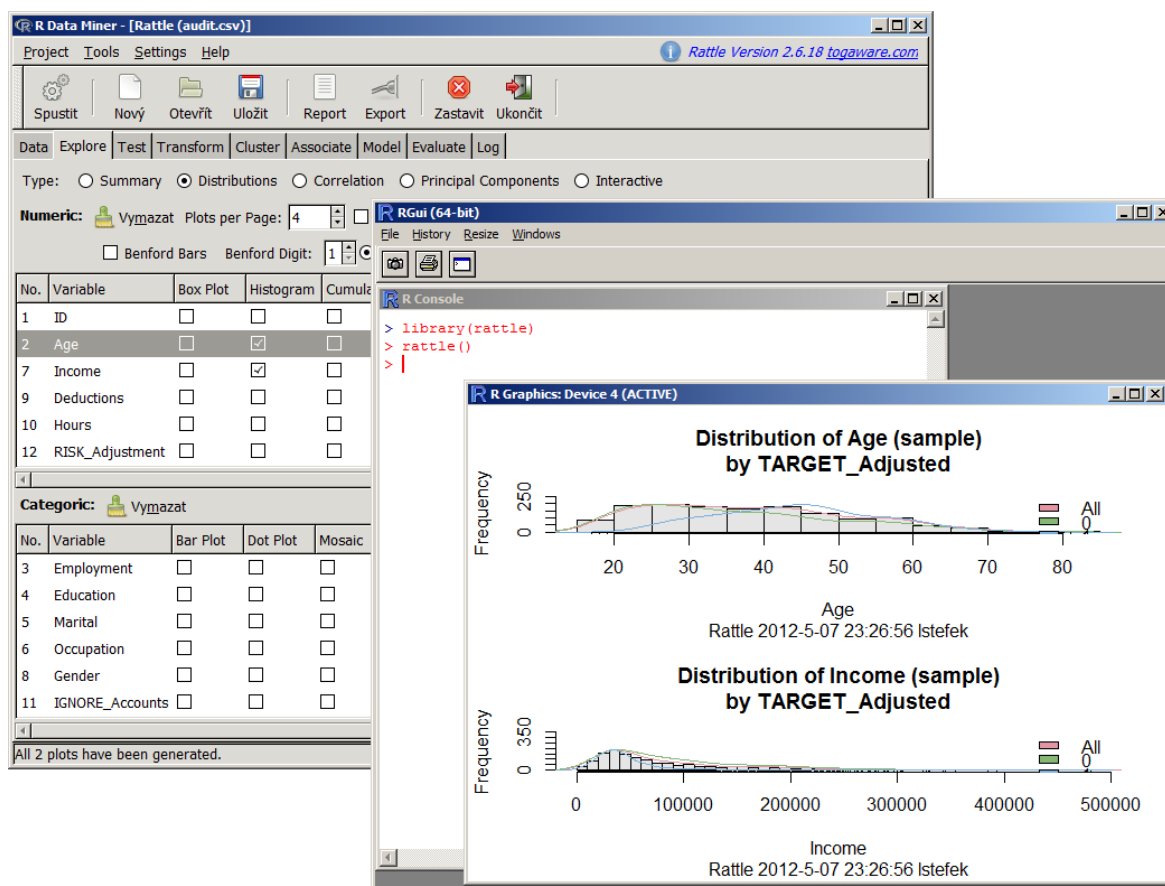
- Komplikovaná rozšiřitelnost a závislost na KDE
- Nestabilita GUI (překreslování)
- Nepřehledný záznam skriptu provedených akcí

2.6 Rattle

2.6.1 Charakteristika

Rattle je zkratkou anglického „the R Analytic Tool To Learn Easily“ – „snadno se naučitelný analytický nástroj pro R“ poskytuje GUI rozhraní pro R, postavené na knihovně GTK+ (tj. balík RGtk2). Podle článku G. J. Williamse v „The R Journal“ [14] byl Rattle vyvinut speciálně pro usnadnění přechodu od základního data miningu, podpořeného nezbytným GUI aparátem, až ke komplexním datovým analýzám pomocí výkonného statistického jazyka R. Rattle staví na poměrně velkém množství dalších podpůrných R balíků, které jsou užitečné pro použití v oblasti datových analýz a data miningu, ale které jsou často obtížně použitelné pro nezkušeného uživatele. Rattle se snaží zde výrazně uživateli pomoci a tak pro započítí práce s Rattle, není znalost R téměř potřeba, ale uživatel je spolu s růstem složitosti datové analýzy, kterou provádí postupně vtažen i do potřebných detailů fungování R v pozadí. Rattle tak představuje vstupní bránu pro použití R jako mocného nástroje pro data mining.

Ve světě je Rattle používáno jednak na mnohých univerzitách při výuce, ale je také rozšířeno mezi analytiky a konzultanty pracujícími v oblasti data miningu a datových analýz. Existuje i komerční podoba Rattle a to v rámci produktů WebFocus firmy Information Builders [16].



Obrázek 9. Rattle: okno aplikace a R konzola

2.6.2 Technické informace

Rattle je naprogramován v jazyce R, staví na funkčnosti mnoha dalších balíků a je distribuován běžným distribučním mechanismem prostředí R. K dispozici jsou i kompletní zdrojové kódy s licencí GNU GPL v2.

2.6.3 Instalace

Velikost instalačního balíku Rattle je přibližně pouze 2MB, je však nutné počítat s potřebou nainstalovat také RGTK a množství dalších balíků (jen pro RGTK to znamená dalších 25MB navíc – více informací o RGTK na straně 49). Vlastní instalace Rattle z R konzoly je obdobná jako pro ostatní R balíky, tedy například takto:

```
> install.packages("rattle")
> library("rattle")
> rattle()
```

2.6.4 Hodnocení

Svým zaměřením patří Rattle mezi nástroje pro reálné použití uživatelem – odborníkem na statistickou datovou analýzu a data mining. Prostředí R jazyka je uživateli částečně skryto a není pro začátky práce s Rattle vyžadováno. Na druhou stranu si Rattle klade za cíl usnadnit postupné proniknutí uživatele do pokročilejšího používání R. Pro práci s jazykem R není Rattle určeno, a proto se doporučuje použití v kombinaci například s RStudiem nebo Tinn-R, které dobře podporují vývoj funkcí pro R. Je ovšem otázkou, zda je tato orientace spíše výhodou či nevýhodou.

Klady

- Multiplatformní
- Je „statisticky“ orientován

Zápory

- Není k dispozici česká jazyková varianta
- Nepodporuje vývoj v R

2.7 RExcel

2.7.1 Charakteristika

RExcel je rozšíření pro tabulkový procesor Microsoft Excel (pro verze 2003, 2007 a 2010). Uživatel pracuje s daty i příkazy R jazyka přímo v uživatelském rozhraní Excelu a pomocí rozšířené kontextové nabídky má možnost přenášet data mezi R a Excelem, spouštět kód v R a přebírat výstup (textový i grafický) z R do sešitu v Excelu.

Rozšíření je k dispozici zdarma, ale pouze pro nekomerční použití. Ačkoli starší verze byly dostupné včetně zdrojových kódů, zřejmý úspěch tohoto produktu přiměl autory k postupnému uzavírání kódu a přechodu ke komerčnímu obchodnímu modelu.

2.7.2 Technické informace a instalace

Ke svému běhu potřebuje několik dalších balíčků, klíčovou komponentou však je statconnDCOM, což je DCOM server, který zprostředkovává veškerou komunikaci s R.

The screenshot shows Microsoft Excel with the following data in cells A1:B22:

x	y
0	100
1	81
2	64
3	49
4	36
5	25
6	16
7	9
8	4
9	1
10	0
11	1
12	4
13	9
14	16
15	25
16	36
17	49
18	64
19	81
20	100

The R console shows the following code:

```

#!rputdataframe mojedata
plot(mojedata)
#!insertcurrentplot
graphics.off()

```

The context menu is open over the plot area, showing options such as 'Run code', 'Get R Value', 'Put R Var', 'Insert Current R Plot', 'Vymazat obsah', 'Filtr', 'Seřadit', 'Vložit komentář', 'Formát buněk...', 'Vybrat z rozevřacího seznamu...', 'Definovat název...', and 'Hypertextový odkaz...'.

Obrázek 10. RExcel – Integrace R a MS Excel

Před vlastní instalací RExcelu je potřeba se rozhodnout, zda budeme používat R souběžně s Excelem (foreground server), nebo ponecháme R běžet na pozadí a konzola R nebude vůbec viditelná (background server) – což je také implicitní volba.

Instalace se provádí tradičně z konzoly a nebyla zcela bezproblémová. Instalační instrukce na domovské stránce produktu se liší od instrukcí, které instalační program vypisuje. Pravděpodobně úspěšný postup instalace je následující:

```

> install.packages("RExcelInstaller")
> installstatconnDCOM()
> library(RExcelInstaller)

```

2.8 RGtknumeric

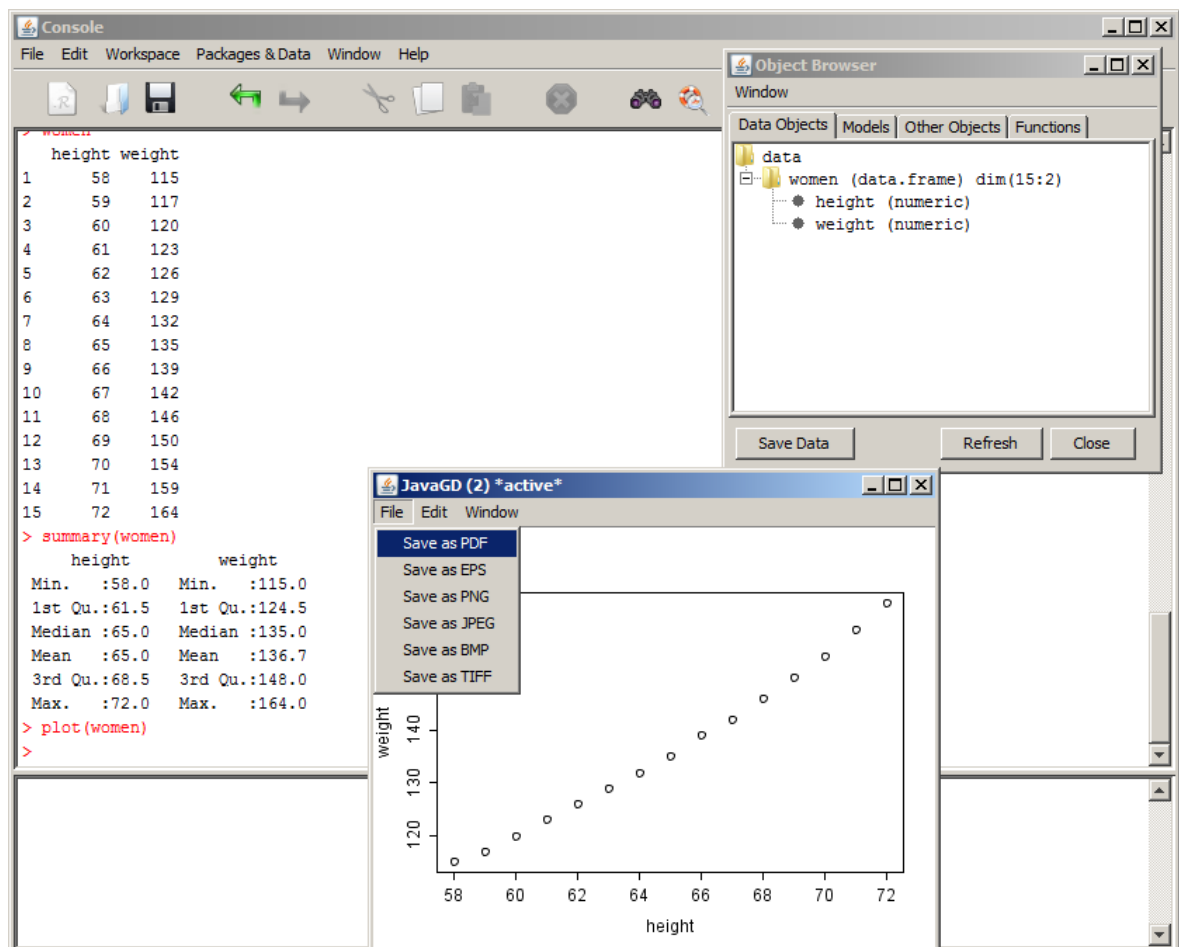
2.8.1 Charakteristika

Tento projekt se svého času pokoušel vyvinout rozhraní mezi R a tabulkovým procesorem Gnumeric, který je standardní součástí linuxového desktopového prostředí GNOME. RGtknumeric lze do jisté míry považovat za obdobu balíku RExcel a dokonce se snažil o užší integraci R a tabulkového procesoru. Bohužel se tento projekt již delší dobu nevyvíjí a je otázkou nakolik jsou zdrojové kódy, které jsou k dispozici ke stažení z <http://www.omegahat.org/RGtknumeric/>, stále ještě použitelné se současnou verzí Gnumeric.

2.9 JGR – Java GUI for R

2.9.1 Charakteristika

Podle poměrně strohých informací na domovské stránce <http://rforge.net/JGR/>, je



Obrázek 11. JGR – Java GUI for R

cílem tohoto projektu vytvořit univerzální a unifikované GUI pro R. První verze JGR byla představena v roce 2004 na konferenci uživatelů R „useR!“ [18]. Současná verze nabízí sice „pouze“ rozhraní na úrovni o něco lepší, než je program RGui na platformě Windows, tj. plně integruje konzolu R (vstup i výstup, včetně grafického), poskytuje editor skriptů, editor dat, prohlížeč proměnných v paměti R, manažer balíčků, ale navíc na tomto základu stavějí některé další projekty a činí JGR velmi zajímavým (Deducer, str. 45 a RGG, str. 50). JGR nabízí také zvýraznění syntaxe R a doplňování kódu.

2.9.2 Technické informace a instalace

Už ze samotného názvu vyplývá, že tento balík je programován v jazyce Java. Pro svůj běh potřebuje nainstalovaný JDK, verze alespoň 1.4 nebo vyšší. Je k dispozici včetně zdrojových kódů, pod licencí GPL.

Instalace a spuštění JGR se provádí opět z konzoly R:

```
> install.packages(c("JGR", "rJava", "JavaGD", "iplots")) # instalace
> library(JGR) # načtení knihovny
> JGR() # spuštění JGR
```

Pro pohodlnější spuštění ve Windows bez nutnosti spouštění R a JGR z konzoly, je ke stažení k dispozici malý exe program (JGR-verze.exe), který sám zajistí spuštění JVM, najde domovský adresář instalace R a spustí prostředí R na pozadí. Pro běžné použití JGR je tento způsob velmi pohodlný.

Klady

- Multiplatformní
- Není „statisticky“ orientován
- Dobrá integrace celého prostředí R

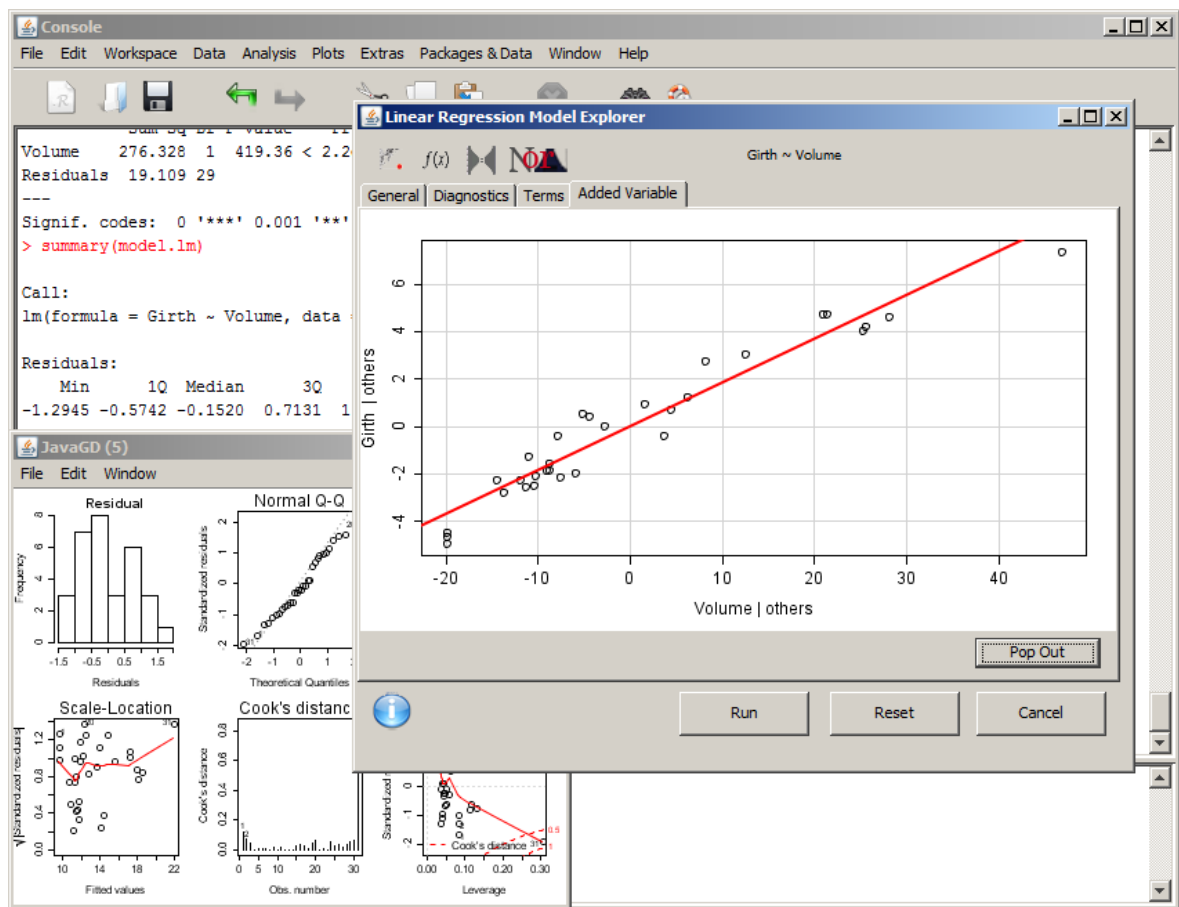
Zápory

- Komunita vývojářů není nijak početná

2.10 Deducer

2.10.1 Charakteristika

Deducer je zdarma dostupná a snadno použitelná alternativa k proprietárním nástro-



Obrázek 12. Deducer – lineární regrese

jmů pro datovou analýzu jako jsou SPSS, JMP nebo Minitab, alespoň podle tvrzení na domovské stránce projektu [17]. Deducer poskytuje přístup k běžným úlohám při manipulaci s daty, k základním analytickým nástrojům a pohodlný způsob editace dat tabulkovou formou. Cílem projektu je:

- poskytnout intuitivní GUI pro R, zejména pro uživatele, kteří se nechtějí zabývat programováním,
- umožnit zkušeným uživatelům R provádět některé typické úlohy pouze několika kliknutími myši.

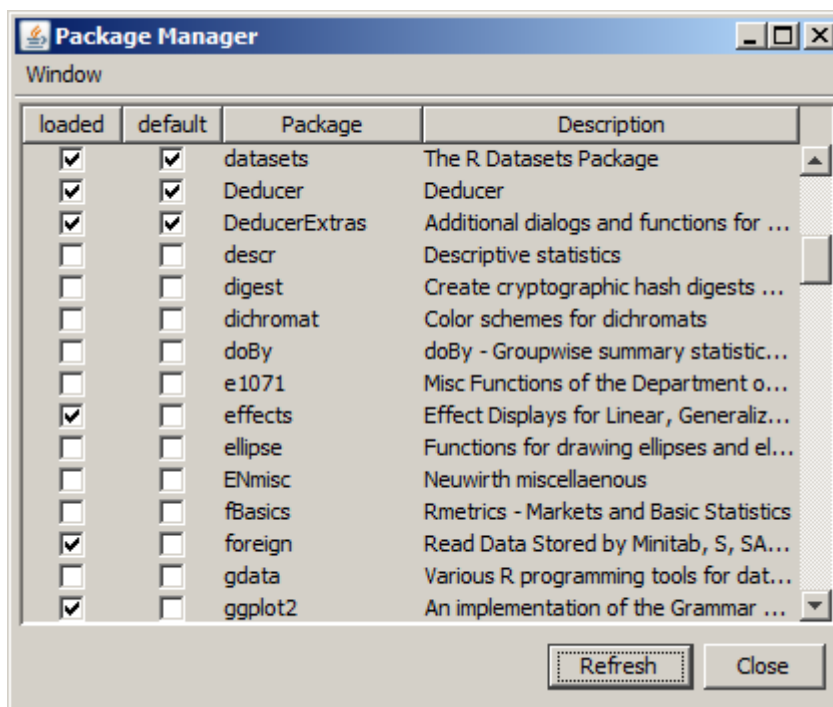
Ačkoli lze Deducer použít i samostatně, byl navržen pro použití spolu s JGR (na straně 44) a instalací obou nástrojů získáme integrované prostředí, jak je vidět na obrázku 12. Po instalaci Deduceru se menu JGR rozroste o položky uvedené v tabulce 4 (je zde vypsána pouze jedna úroveň menu). Navíc k tomuto seznamu balík DeducerExtras přidává další položky zde neuvedené.

Data	Analysis	Packages + Data	Plots
Edit Factor	Frequencies	Data Viewer	Plot Builder
Recode Variables	Descriptives	Object Browser	Import Template
Transform	Contingency Tables		Open Plot
Reset Row Names	One Sample Test		Templates
Sort	Two Sample Test		Interactive
Transpose	K-Sample Test		
Merge Data	Correlation		
Subset	Linear Model		
	Logistic Model		
	Gen. Linear Model		

Tabulka 4. Deducer: přehled položek menu

2.10.2 Technické informace a instalace

Deducer je vyvíjen v jazyce Java, dostupný je opět pod licencí GPL a včetně zdrojo-



Obrázek 13. JGR: Package Manager – jak automaticky spouštět Deducer

vých kódů. Pro běžné používání je nejlepší způsob spouštění nechat si v správci balíčků v JGR Deducer automaticky spouštět (obrázek 13). Instalace se provádí opět z konzoly R:

```
> install.packages(c("Deducer", "DeducerExtras")) # instalace
```

2.11 Red-R

2.11.1 Charakteristika

Red-R přináší do práce s R vskutku netradiční přístup, alespoň v porovnání s ostatními nástroji v rámci této práce. Cílem projektu Red-R je poskytnout uživateli přístup k funkčnosti R úplně bez potřeby mít znalosti programování v R a tohoto cíle se snaží dosáhnout pomocí vizuálního kombinování předpřipravených komponent, jejich propojování a parametrizace – modelování datového toku. Komponenty, které má uživatel k dispozici zajišťují požadované funkce – manipulace s daty, vizualizaci, vstup/výstup a podobně. Jejich propojením uživatel definuje požadovaný datový tok a parametrizuje jej.

2.11.2 Technické informace a instalace

Red-R je multiplatformní a staví na projektu Orange (domovská stránka <http://orange.biolab.si/>) což je nástroj pro vizuální programování, vizualizaci a analýzu. Vývoj probíhá převážně v C/C++ a Pythonu, použit je i grafický toolkit Qt.

2.12 Ostatní – nástroje pro GUI

2.12.1 R-Tcl/Tk

Charakteristika

Balík `tcltk` patří do skupiny nástrojů pro tvorbu GUI pro R [11]. Zpřístupňuje Tcl/Tk a zvláště pak komponenty knihovny Tk pro jejich použití přímo z prostředí jazyka R. Tcl je skriptovací jazyk, pro který byla knihovna Tk původně také vyvinuta. Tk je multiplatformní knihovna komponent (GUI toolkit) pro vytváření grafických uživatelských rozhraní. Samotný jazyk Tcl je sice v balíku `tcltk` použit, ale pouze jako tenká vrstva mezi R a Tk. Balík `tcltk` sám o sobě nepřidává nějaké komponenty speciálně pro R (na rozdíl od `RGtk`). Zjednodušeně lze říci, že `tcltk` „pouze“ zpřístupňuje grafické komponenty Tk do jazyka R a jazyk Tcl je zde kompletně překryt.

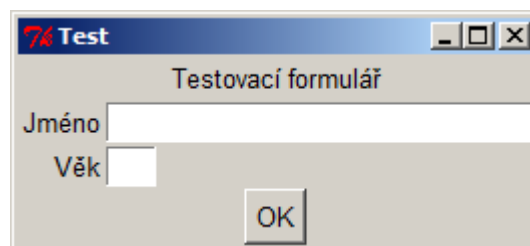
Určitou výhodou může být, že `tcltk` je standardní součástí základní instalace R na platformě MS Windows. Naopak je zde nevýhodou do značné míry rozdílný vzhled grafických komponent. Příkladem použití `tcltk` v reálné aplikaci je R Commander, více informací v kapitole 2.2 na straně 30.

Příklad kódu

```
# tcltk_ex1.R
win <- tktoplevel()
tktitle(win) <- "Test"
popisek <- tklabel(win, text="Testovací formulář")
l.jmeno <- tklabel(win, text="Jméno")
l.vek <- tklabel(win, text="Věk")
e.jmeno <- tkentry(win, width=30)
e.vek <- tkentry(win, width=3)
tkgrid(popisek, columnspan=2)

tkgrid(l.jmeno, e.jmeno)
tkgrid(l.vek, e.vek)
tkgrid.configure(e.jmeno, e.vek, sticky="w")
tkgrid.configure(l.jmeno, l.vek, sticky="e")
tlacitko <- tkbutton(win, text="OK")
tkgrid(tlacitko, columnspan=2)
button.ok.press <- function() {
tkdestroy(win)
}

tkconfigure(tlacitko, command = button.ok.press)
```



Obrázek 14. Výsledek ukázkového příkladu 2.12.1

Technické informace a instalace

Tento balík by měl být nainstalován jako součást základu prostředí R, není potřeba se tedy obávat komplikací jako to občas bývá například u balíku `RGtk`.

2.12.2 RGtk

Charakteristika

Balík `RGtk` patří opět do skupiny vývojových nástrojů pro tvorbu GUI pro R [12]. Zpřístupňuje komponenty multiplatformní knihovny GTK+, která je populární také díky jejímu použití ve velkých projektech jako jsou GNOME nebo GIMP.

Příkladem použití `RGtk` v reálné aplikaci je `RGnumeric`, více informací v kapitole 2.8 na straně 44 nebo `Rattle` v kapitole 2.6 na straně 40.

Příklad kódu

```

require("gWidgets")
require("RGtk2")

#####
# GUI priklad - RGtk2
# Date: 24.1.2012
# L. Stefek
#
guiG <- function() {
  options("guiToolkit"="RGtk2")
  GEv <- new.env()

  GEv$win <- gwindow("Data_Browser",width=300, height=600)
  GEv$grp <- ggroup(horizontal=FALSE, cont = GEv$win)
  GEv$pg <- ggroup(horizontal=TRUE, cont = GEv$grp)

  ## GUI prvky
  GEv$lb1 <- glabel("User_Workspace_Data_Browser", cont=GEv$pg)
  GEv$lb2 <- glabel("Data_Set:", cont=GEv$pg)
  GEv$edt <- gedit( cont=GEv$pg, coerce.with=as.character, handler=NULL)

  GEv$gpg <- gpanedgroup(width = 300,cont=GEv$grp, expand=TRUE)
  GEv$vbr <- gvarbrowser(cont = GEv$gpg, handler=GEv$akce ) # neni zde
  GEv$ntb <- gnotebook(cont = GEv$gpg)

  ## Vystupni textova oblast
  GEv$output <- gtext(width=300,cont=GEv$ntb, label="Output", expand =TRUE)
}
#####
# spusteni:
guiG()

```

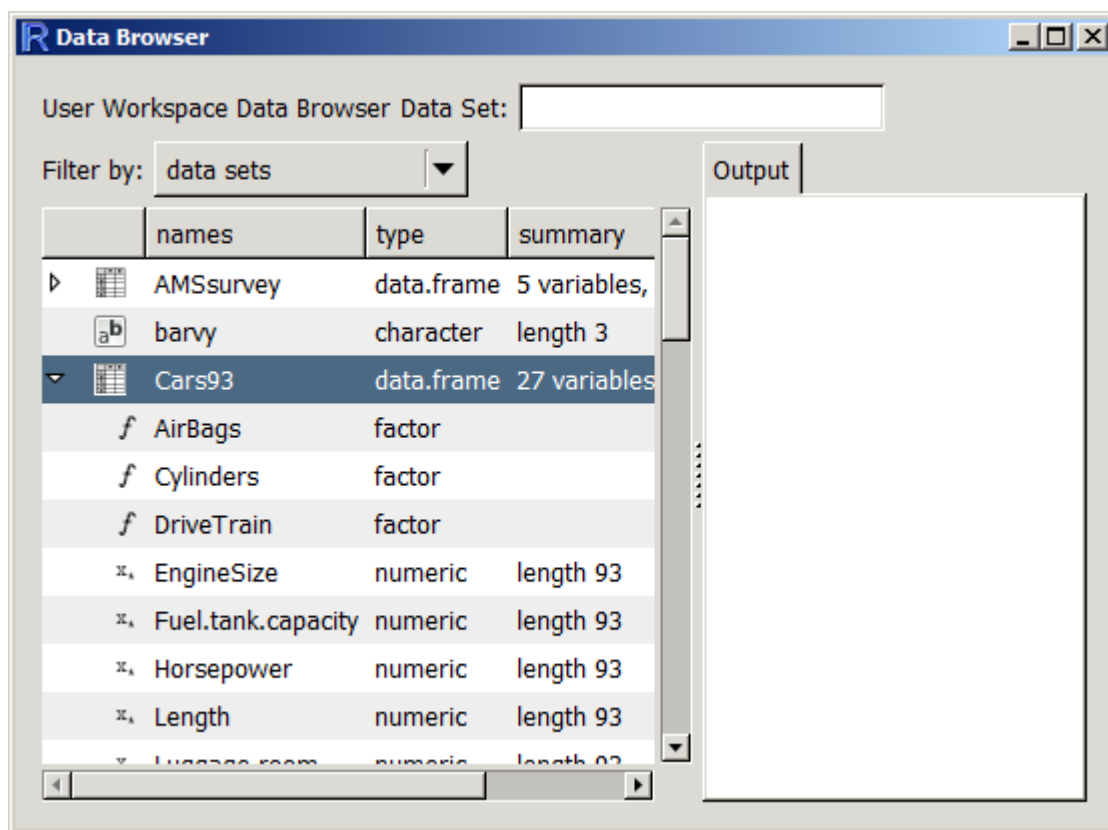
V porovnání s `tcltk` je balík `RGtk` bohatší na GUI komponenty a to jak základní, tak poměrně vysokoúrovňové. V příkladu 2.12.2 to je například komponenta `gvarbrowser`, která samotná zpřístupňuje objekty v paměti aktuálně běžícího R.

Technické informace a instalace

Určitou nevýhodou je nepřítomnost `GTK+` v základní instalaci R na platformě MS Windows a s tím spojená nutnost dodatečné instalace většího množství závislých balíků (> 25MB). V případě instalace na linuxu se lze setkat s problémy s dostupností vzájemně kompatibilních verzí R, `RGtk` a `GTK+` v instalačních repositářích konkrétní linuxové distribuce (například u Ubuntu 11.04).

2.12.3 RGG

RGG je zkratkou pro „R GUI Generator“ a měl by umožňovat snadnou tvorbu grafických uživatelských rozhraní pro R pomocí XML. RGG se skládá z definice rozhraní popsaného v XML souboru a generátoru GUI napsaného v jazyce Java. GUI je genero-



Obrázek 15. Výsledek ukázkového příkladu RGtk 2.12.2

váno za běhu z definovaných grafických značek, výsledek uživatelského vstupu je pak vložen jako kód R do právě interpretovaného skriptu. RGG soubory jsou běžné textové soubory a mohou být vytvořeny pomocí libovolného textového editoru. Aktuální verze RGG je k dispozici, buď jako samostatný program (RGGRunner), nebo jako plug-in pro JGR. RGG je open source projekt pod licencí LGPL a lze je stáhnout volně na <http://rgg.r-forge.r-project.org>.

2.13 Ostatní – editory a rozšíření pro editory

Velkou skupinu programů tvoří specializované editory, respektive rozšíření existujících editorů pro práci s R. V tabulce 5 jsou uvedeny některé z nich. Za zvláštní zmínku stojí editor Tinn-R, který má již některé vlastnosti IDE.

2.14 Ostatní nástroje

Kromě doposud vyjmenovaných nástrojů, které přímo tvoří či souvisí s GUI pro prostředí R, existuje skupina projektů, které nepřímo podporují vývoj uživatelského rozhraní a to zejména pomocí webových technologií. V tabulce 6 jsou uvedeny některé

Název	Platforma	Vlastnosti
Tinn-R	MS Windows	Oblíbený editor, téměř IDE
Rgedit	GNU/Linux	Varianta gedit – jednodušší editor pro linux
TextMate	MAC OS	Oblíbený editor na platformě MAC
Emacs (ESS)	Multiplatformní	Komplexní editor
WinEdt	MS Windows	Syntaxe R
Vim	Multiplatformní	Podpora syntaxe R
jEdit	Multiplatformní	Editor napsaný v Javě. Zvýraznění syntaxe R

Tabulka 5. Některé editory pro R

z nich, spolu s velmi stručnou charakteristikou.

Za zvláštní pozornost stojí první dva z nich. R-DCOM server je na platformě Windows součástí některých dalších projektů. Rserve je multiplatformní implementace TCP/IP serveru, který otevírá možnost přistupovat k R prostřednictvím síťových technologií a architekturou klient/server. Právě Rserve je klíčovou komponentou prototypu RWeb v praktické části (kapitola 4) a je také používána některými dalšími projekty.

Název	Použití a základní vlastnosti
R-(D)COM server	DCOM rozhraní pro R – pro Windows. Od stejných autorů jako RExcel a je jeho nezbytnou součástí (kap. 2.7). Použito i v dalších projektech. Url: http://www.statconn.com/
Rserve	Komponenta, která implementuje TCP/IP server (nově také přes HTTP nebo WebSockets) umožňující klient/server přístup aplikací k R. Podporuje vzdálená připojení, autorizaci přístupu a jeho součástí jsou knihovny pro klientskou část aplikace pro jazyky C/C++ a Java [15].
RStatServer	Neaktivní projekt. Webové rozhraní k R. Url: http://sourceforge.net/projects/rstatserver/ .
RPy	Rozhraní Python – R. Použito v projektu Red-R (kap. 2.11). Url: http://rpy.sourceforge.net/
Rpad	Webové rozhraní k R generované pomocí sady R skriptů. Již zcela neaktivní projekt, ale stále jsou k dispozici zdrojové kódy v SVN na adrese http://rpad.googlecode.com/svn/trunk .
RSOAP	SOAP rozhraní k R. Neaktivní projekt. Url: http://sourceforge.net/projects/rsoap/ .
R.NET.Web	Neaktivní. Spíše ukázka implementace komponenty R-DCOM v prostředí .NET Url: https://www.msbi.nl/dnn/Research/SurvivalAnalysis/RNetWeb/tabid/142/Default.aspx

Tabulka 6. Některé další nástroje pro R

II. PRAKTICKÁ ČÁST

3 ÚVOD DO PRAKTICKÉ ČÁSTI

Požadavky na cílový systém byly v první fázi formulovány zadavatelem poměrně obecně. Některé vlastnosti, zejména související s technickým řešením jednotlivých požadavků, byly vyjasňovány až postupně s nabytými informacemi o reálných možnostech zkoumaných nástrojů a technologií.

Další důležitou okolností byly měnící se priority požadavků zadavatele v průběhu času. Týká se to zejména technické architektury systému. Původní představa, podle které by byla dostačující samostatně běžící GUI aplikace, se postupně změnila a priority získalo řešení postavené na webových technologiích, vzdáleném přístupu a na větším důrazu na spolupráci více uživatelů. Tyto měnící se priority však plně odpovídají měnícím se požadavkům v oblasti trhu, kde zadavatelská firma působí. Proto bylo nutné tyto změněné požadavky akceptovat a přizpůsobit tomu odpovídajícím způsobem, také zaměření této práce.

Cílový systém by měl, dle výchozích požadavků, obsahovat následující moduly:

1. vstup a úprava datových souborů,
2. spouštění aplikací a procedur datové analýzy,
3. prezentace výsledků, podpora tvorby grafických výstupů a dokumentace,
4. případné podpůrné funkce pro ovládání prostředí R.

Jako upřesnění byly dále formulovány následující požadavky na realizované uživatelské rozhraní (výslednou volbou by pravděpodobně byl RGtk2):

- přenos vstupních dat do R s možností jednoduché editace formou tabulky,
- obsluha volání funkcí a plnění jejich argumentů formou formuláře,
- výstupní data zobrazovat graficky i tabulkově,
- podpora prohlížení a editace datových struktur v R prostředí,
- lokální aplikace s využitím některého existujícího toolkitu.

Dalším upřesněním však získaly vysokou prioritu požadavky na:

- vzdálený, víceuživatelský přístup přes webové rozhraní a
- úplné odstínění koncového uživatele od systému R,

které pak již vedly k volbě RServe a později k LabKey.

4 WEBOVÉ UŽIVATELSKÉ ROZHRANÍ PRO R

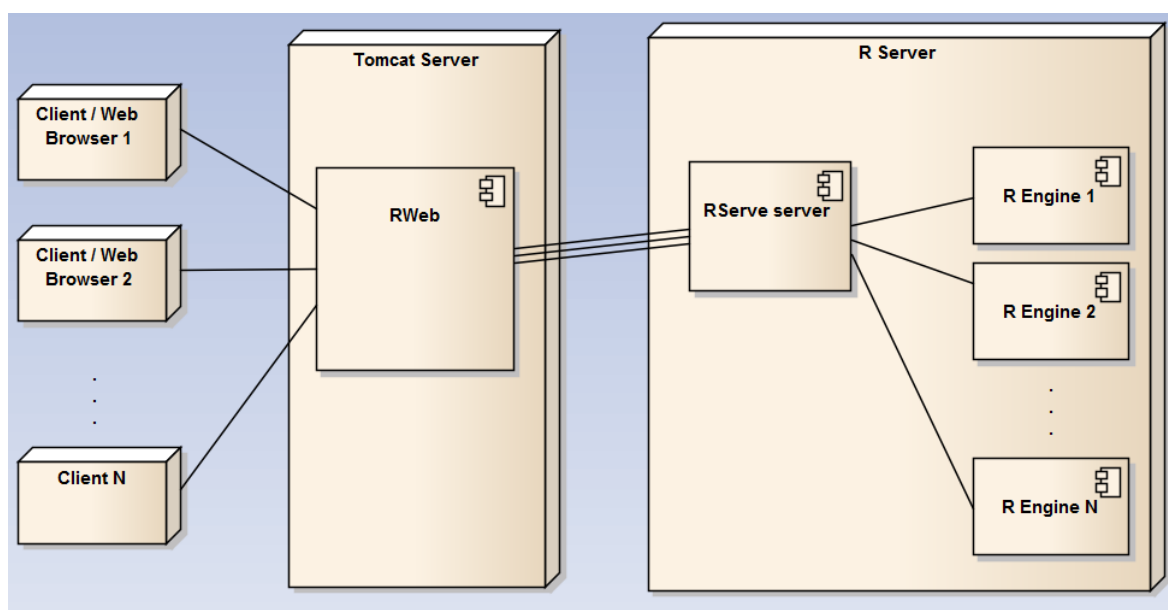
Balík RServe je programová komponenta, která implementuje TCP/IP server umožňující vzdálený, klient/server přístup aplikací k R. Podporuje vzdálená připojení, autorizaci přístupu a jeho součástí jsou knihovny pro klientskou část aplikace pro jazyky C/C++ a Java [15]. RServe je naprogramován v C a je multiplatformní.

Běžící RServe funguje jako server naslouchající na síťovém portu 6311 a přijímá požadavky od klientů. Pro tvorbu klientské části aplikace je k dispozici knihovna funkcí.

4.1 RWeb

RWeb je prototypová aplikace pro spouštění kódu v jazyku R v prostředí webového prohlížeče. Jako webový server jsem zvolil Apache Tomcat, který poskytuje potřebné technologie „Java Servlet Container“ a „JavaServer Pages“, a sám je napsaný v jazyce Java.

Návrh architektury aplikace, na obrázku 16, ilustruje možnou vícevrstvou architekturu, kdy prostředí R může být odděleno od webového serveru. To poskytuje určité možnosti, jednak dodatečnou úroveň zabezpečení a také lepší škálovatelnost. Podobným způsobem může být samozřejmě oddělen i databázový server, což je ale běžné řešení.



Obrázek 16. Náčrt architektury aplikace RWeb

Na následujícím snímku obrazovky (obr. 17) je již existující prototyp aplikace. Umožňuje, na požadavek uživatele, spustit a ukončit spojení s RServe (a tím i vytvářet a rušit instance R), zadat jednoduché výrazy jazyka R do horního textového pole a spustit je tlačítkem Run. Textový výstup vyhodnoceného výrazu je zobrazen v dolním textovém poli. Aplikace také zobrazí výsledný graf (ne zcela automaticky, ale za pomoci R balíku Cairo). V levé části stránky je po každém vyhodnocení výrazu také aktualizován seznam objektů v paměti běžící instance R.

The screenshot shows the Apache Tomcat & Rserve Test application interface. The main area is titled "Execute R commands" and contains a text input field with the R command `summary(cars)`. Below the input field are three buttons: "Stop R session", "Clear form", and "Run". The output of the command is displayed in a text area below the input field, showing summary statistics for "speed" and "dist". To the right of the text area is a scatter plot of "dist" versus "speed".

speed		dist	
Min. :	4.0	Min. :	2.00
1st Qu. :	12.0	1st Qu. :	26.00
Median :	15.0	Median :	36.00
Mean :	15.4	Mean :	42.98
3rd Qu. :	19.0	3rd Qu. :	56.00
Max. :	25.0	Max. :	120.00

Obrázek 17. Obrazovka aplikace RWeb

Aplikace RWeb umožňuje otestovat základní principy komunikace mezi webovou aplikací a R. Struktura této aplikace je velmi jednoduchá a kromě základních technologií (JSP, Java, HTML, CSS, RServe.jar) nepoužívá žádné další komponenty.

4.2 Další vývoj RWeb

Pro další rozvoj této aplikace by však bylo vhodné zvolit některý existující „Java web framework“, který by měl podpořit běžné funkce moderní webové aplikace jako jsou:

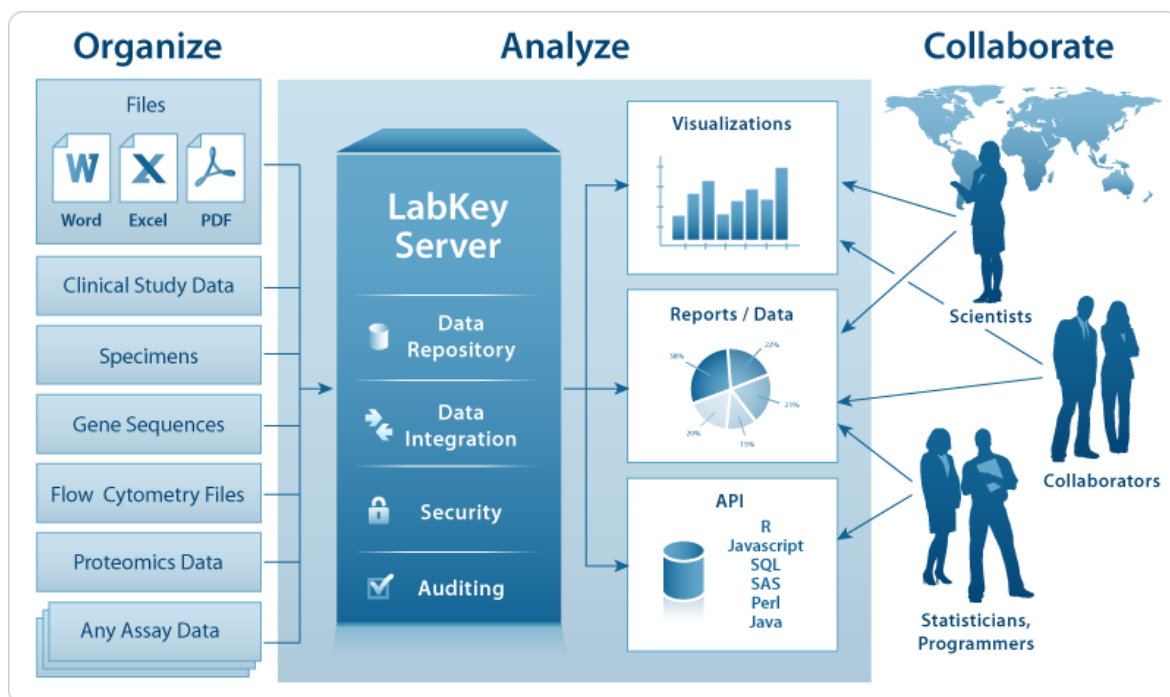
- základní zabezpečení aplikace pomocí autorizace uživatele jménem a heslem,
- správu uživatelů,
- správa sessions,
- vývojový přístup Model – View – Controller,
- podporu potřebných prvků uživatelského rozhraní, zejména editor dat formou editovatelné tabulky a další moderní prvky,
- vestavěnou podporu AJAX technologie, zejména pro dosažení přijatelné uživatelské odezvy v prostředí webového prohlížeče.

V současné době je několik takových, volně dostupných, nástrojů k dispozici. Vhodný „framework“ by pro tuto aplikaci mohl být například Apache Wicket, Vaadin nebo Spring.

5 LABKEY

5.1 Základní informace

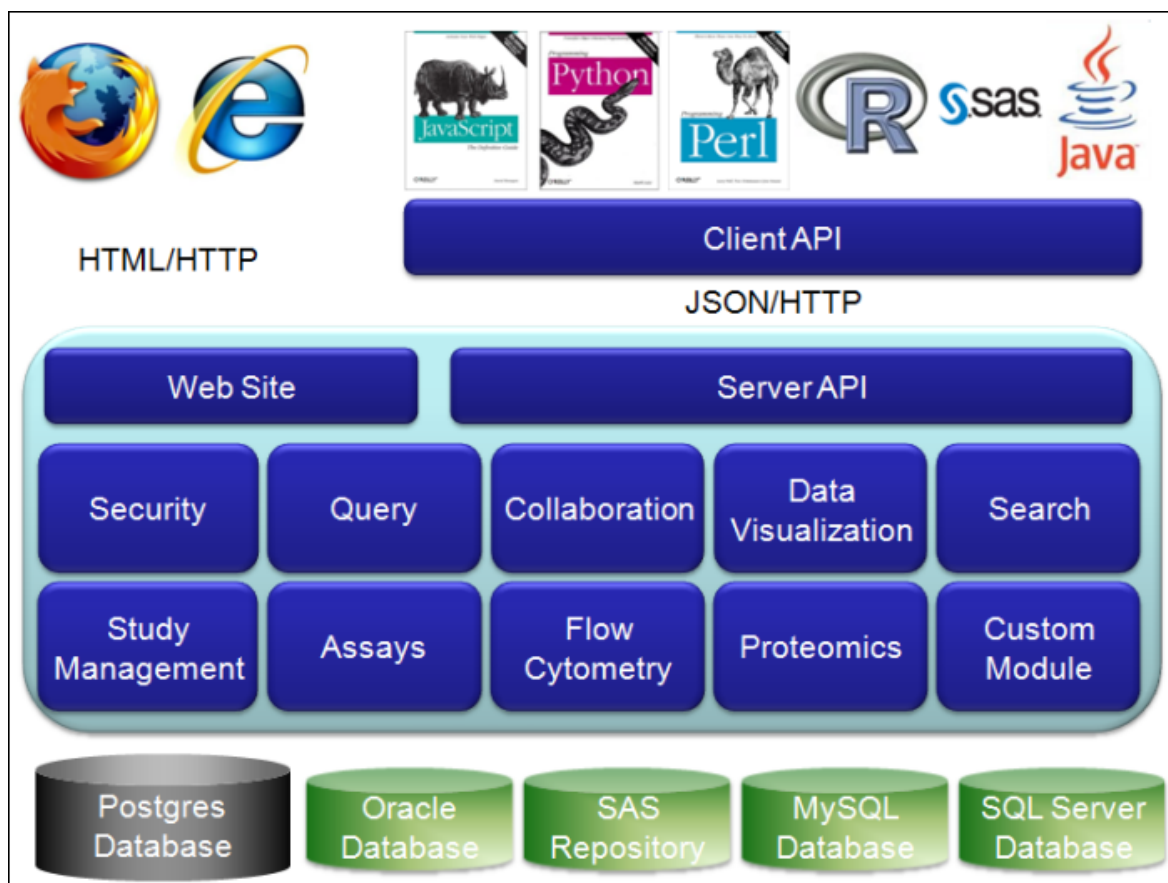
LabKey je volně šířený programový balík, který si klade za cíl pomoci uživatelům organizovat, analyzovat, sdílet biomedicínská data a prezentovat výsledky analýz. Hlavním uživatelským rozhraním je webový prohlížeč, z tohoto pohledu je tedy LabKey zejména webový server. Je vyvíjen skupinou odborníků sdružených v organizaci LabKey Software Foundation a používán několika výzkumnými pracovišti a univerzitami, zejména ve Spojených státech amerických [21]. Licenční podmínky použití tohoto programového produktu jsou příznivé i pro komerční použití, konkrétně se jedná o licenci „Apache Software License“. Přehled o klíčových funkcích LabKey a jeho celkovém



Obrázek 18. LabKey Server

zaměření podává obrázek 18. Architektura systému a přehled hlavních komponent je uveden na obrázku 19, jde ale o poměrně hrubé členění. Při správě systému se pracuje s detailnějším dělením na moduly, například oblast „Colaboration“ zahrnuje moduly Wiki, Issues, Messages a další.

Kromě obecně použitelných modulů (Admin, Audit, Authentication, FileContent, Issues, List, Messages, Portal, Query, Search, Study, Visualization, Wiki) je standardně k dispozici několik specializovaných modulů, zejména pro výzkumné biomedicínské úlohy (Proteomics, Flow Cytometry, DNA Sequencing & Genotyping a mnoho dalších).



Obrázek 19. LabKey Server – klíčové komponenty

Na domovské stránce projektu (<http://www.labkey.org/>) je k dispozici rozsáhlá a podrobná dokumentace, pokrývající vše od instalace, přes tutoriály o používání LabKey, jeho konfiguraci až po vývojářskou dokumentaci a doporučení pro vlastní vývoj modulů.

5.2 Technická architektura LabKey

LabKey je webová aplikace běžící v prostředí aplikačního serveru Apache Tomcat a její součástí je celá řada doplňkových komponent, zejména pro podporu integrace LabKey s okolím. Aplikace je modulární – moduly mají předepsanou strukturu, jsou na sobě relativně nezávislé a umožňují dodatečně rozšiřovat aplikaci o požadovanou funkcionalitu. Modulů existuje několik desítek už v základní distribuci a je možné si také vyvinout vlastní moduly („Custom Module“) nebo rozšířit existující.

LabKey je vyvíjen především v programovacím jazyce Java, používá JSP, Spring Framework, JavaScript a GWT (Google Web Toolkit). Tato kombinace naznačuje zaměření vývoje na moderní technologie, používá AJAX techniky pro dosažení rychlejší

odezvy na uživatelské požadavky a přitom je multiplatformní a podporuje všechny moderní webové prohlížeče.

Jako databázový server lze pro LabKey použít buď PostgreSQL nebo Microsoft SQL Server. Pro účely datové integrace lze pomocí JDBC připojení přistupovat i do jiných databází. Preferovaný databázový server je však PostgreSQL (obr. 19).

5.3 Instalace a úvodní konfigurace

Nejjednodušším způsobem získání funkční, předkonfigurované instalace LabKey, včetně databáze PostgreSQL, webového serveru Tomcat a JRE/JDK je použití instalačního programu pro Windows. Konkrétně pro aktuální verzi LabKey (12.1) to jsou:

- Sun Java Runtime Environment (JRE) 1.6.0-31,
- Apache Tomcat 5.5.33 a
- PostgreSQL 8.3.7 .

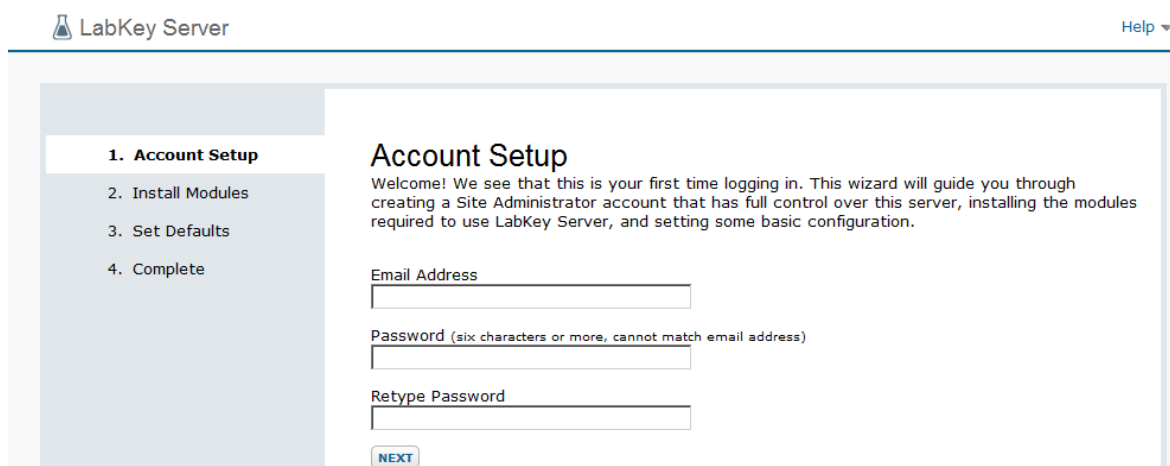
Další možností je manuální instalace pro Windows/Unix/Linux, kdy instalujeme jednotlivé potřebné komponenty zvlášť.

5.3.1 Instalace LabKey ve Windows

Instalace pomocí instalačního programu ve Windows je bezproblémová. Instalační program vytvoří zvláštního Windows uživatele pro PostgreSQL a založí systémovou službu pro automatické spouštění této databáze. Jedinou informací, kterou je potřeba před započítím instalace mít, je nastavení SMTP komunikace pro odchozí poštu. Pokud nenastavíme platný SMTP server a port, nebude LabKey moci poštu odesílat. Případná dodatečná konfigurace je možná, změnu provedeme v konfiguračním souboru `<LabKeyHome>\<apache tomcat>\Catalina\localhost\labkey.xml`, kde nastavíme položky `mail.smtp.host`, `mail.smtp.user` a `mail.smtp.port`.

Jako poslední krok instalace je spuštěn prohlížeč (<http://localhost:8080/labkey/login/initialUser.view>), kde je potřeba v několika krocích provést úvodní nastavení aplikace, jméno a heslo administrátora, instalace modulů a nastavení adresáře pro soubory (obrázek 20).

Po dokončení instalace by měly být databáze i webový server spuštěny, včetně dvou



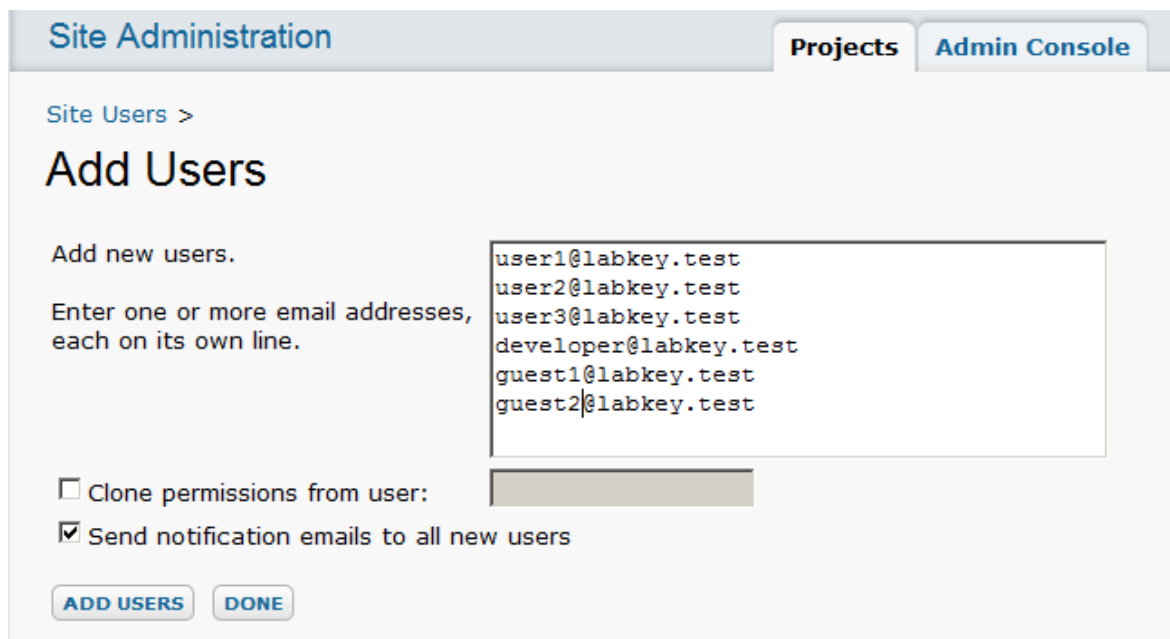
Obrázek 20. Instalace LabKey Server – závěr instalace

automaticky spouštěných služeb (Windows Services) – LabKey Server Apache Tomcat 5.5 a LabKey Server PostgreSQL Database Server.

Pro lokální přístup do aplikace použijeme adresu `http://localhost:8080/labkey/`, případně konkrétní hostname daného počítače.

5.3.2 Administrace LabKey

Během instalace byl vytvořen první uživatel systému, který je zároveň administrá-



Obrázek 21. Administrace LabKey – založení uživatelů

torem celého LabKey serveru. Administrátor může definovat další uživatele systému, přiřazovat jim různé role, definovat skupiny uživatelů a řídit přístup uživatelů a skupin

k projektům.

Existuje zde několik kategorií uživatelů – administrátor, vývojář, uživatel a host (Admin, Developer, User, Guest). Každému uživateli může být přiřazena jedna nebo více rolí. Rolemi jsou – Site Administrator, Developer, Reader, Author, Project Administrator, Folder Administrator a speciální Troubleshooter.

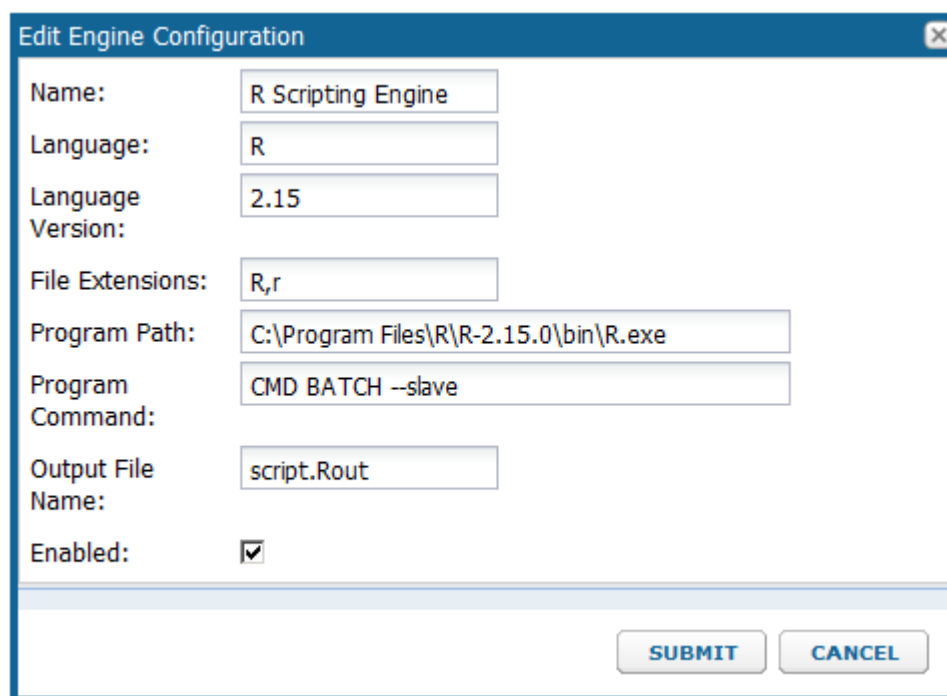
Jako identifikace uživatele slouží jeho e-mailová adresa a pro základní vytvoření uživatele není žádná další informace potřeba. Na obrázku 21 je dialog pro založení několika uživatelů najednou (navigace: Admin → Manage Site → Site Users → Add User).

Další součástí administrace je Admin Console kde se provádí konfigurace celého serveru, některá důležitá nastavení zde jsou:

- **Authentication** – nastavení způsobu uložení a ověřování uživatelů: LDAP, OpenSSO, lokální databáze
- **Email Customization** – úprava textu zasílaných zpráv
- **Files** – nastavení adresáře pro uložené soubory
- **Look and Feel Settings** – úprava vzhledu webového rozhraní, logo, CSS styly, barevné téma stránek
- **Site Settings** – přepnutí aplikace do režimu údržby, nastavení doménového jména a další systémová nastavení
- **Views and Scripting**; – nastavení externích aplikací (Scripting Engines), Perl, R, ECMAScript scripting engine

5.4 Integrace LabKey a R

LabKey obsahuje obecný mechanismus pro integraci externích programů („Scripting Engine“) a takto je možné také integrovat prostředí R do systému. Navíc konkrétně pro integraci s R je v LabKey nachystán celý aparát podpůrných funkcí a jeho použití je celkově snadné a ověřené. Podmínkou používání R se současnou verzí LabKey (12.1) je instalace prostředí R přímo na stejném počítači, a protože R není automatickou součástí instalace LabKey, je potřeba ji provést zvlášť. Na obrázku 22 je pak jeden z kroků konfigurace LabKey, kdy je nutné specifikovat cestu k instalaci R (cesta k tomuto nastavení je Admin Console → Views and Scripting). Jak vyplývá z uvedených parametrů, je použití R v rámci LabKey „dávkové“, to znamená, že pro každé provedení



Obrázek 22. LabKey Server – konfigurace R Scripting Engine

R skriptu se musí vždy inicializovat R sezení, načíst potřebné balíky a data. To má za následek pomalejší odezvu, určitá možná výkonová omezení, bezpečnostní rizika a horší škálovatelnost. Možnou lepší alternativou by zde bylo využití balíku RServe (kap. 2.14) a dokonce byly na požadavek uživatelů, provedeny určité pokusy o zařazení RServe do instalace LabKey, ale nebyly dotaženy do konce a tak zde podpora pro tuto možnost stále chybí. Od té doby se vývoj RServe nezastavil a tak by se tento způsob integrace R a LabKey dal považovat za dobrý námět pro vývoj.

5.5 Použití LabKey – první projekt

Pro organizaci dat se v LabKey používají „projekty“ (Project) a jim podřízené „složky“ (Folder). V podstatě jde o rovnocenné kontejnery, prostřednictvím kterých se zpřístupňují moduly systému, jim odpovídající obsah a definují se přístupová práva uživatelů a skupin.

Při vytváření projektu (navigace: Admin → Manage Site → Create Project) uživatel zvolí typ projektu, což je vlastně určitá předvolená kombinace modulů vhodných pro daný účel. Takovými typy jsou například:

- **Assay** je projekt pro chemické analýzy, které zpracovávají data některých standardizovaných formátů nebo obecná tabulková data,

- **Colaboration** je projekt zaměřený na sdílení informací pomocí modulů Issues, Messages, Wiki, FileContent a dalších,
- **Study** je obecnější studie dat, která pracuje časovou řadou pozorování (Visits) na definované množině prvků (Participant)
- **Custom** umožňuje volně definovat moduly použitelné v projektu

a další, jak je ukázáno na obrázku 23. V některých případech se typ projektu shoduje se jménem modulu, který je pro daný projekt klíčový.

1. Name and Type

2. Users / Permissions

3. Project Settings

Name and Type

Name:

Type:

Assay

Collaboration

Flow

MS1

MS2

Microarray

Study

Custom

Default Tab:

Create a tab for each LabKey module you select. Used in older LabKey installations. Note that any LabKey module can also be enabled in any folder type via Folder Settings.

Choose Modules:

<input type="checkbox"/> ELISpotAssay	<input type="checkbox"/> Experiment	<input checked="" type="checkbox"/> FileContent
<input type="checkbox"/> Flow	<input checked="" type="checkbox"/> Issues	<input checked="" type="checkbox"/> List
<input type="checkbox"/> Luminex	<input type="checkbox"/> Messages	<input type="checkbox"/> Microarray
<input type="checkbox"/> MS1	<input type="checkbox"/> MS2	<input type="checkbox"/> Nab
<input type="checkbox"/> Pipeline	<input checked="" type="checkbox"/> Portal	<input type="checkbox"/> Query
<input checked="" type="checkbox"/> Search	<input checked="" type="checkbox"/> Study	<input checked="" type="checkbox"/> Visualization
<input checked="" type="checkbox"/> Wiki		

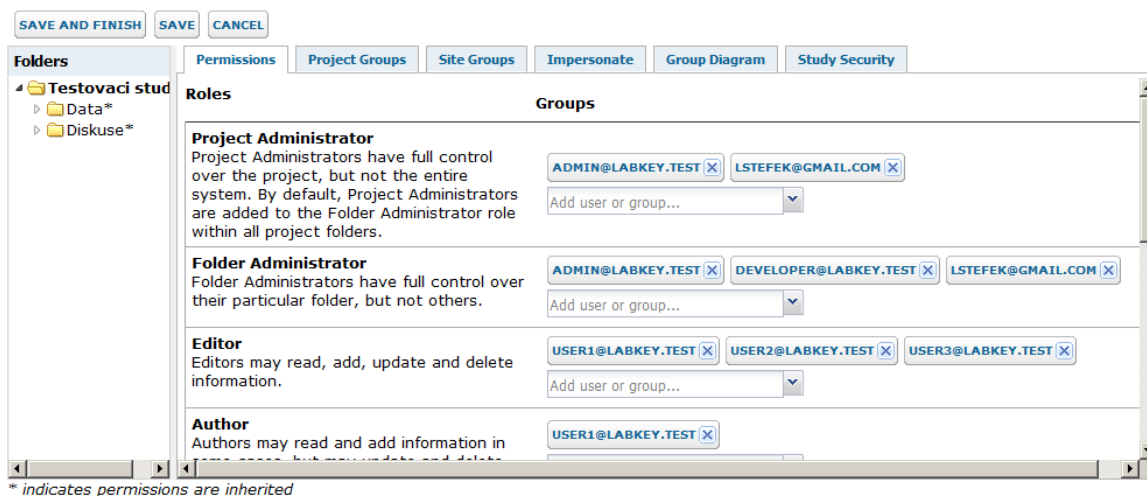
Obrázek 23. Vytvoření projektu, volba jeho typu a modulů

5.5.1 Projekt typu Study

Typem projektu, který by mohl nejlépe pokrývat potřeby definované zadáním, je Study. Prvními kroky u každého projektu v LabKey je vytvoření struktury složek. Ke každé složce definujeme jaký obsah bude zpřístupňovat a tedy jaké prvky jednotlivých modulů (někdy označované také jako portlety) umístíme do té které složky. Dalším důležitým krokem je určení přístupových práv uživatelů a skupin k jednotlivým složkám.

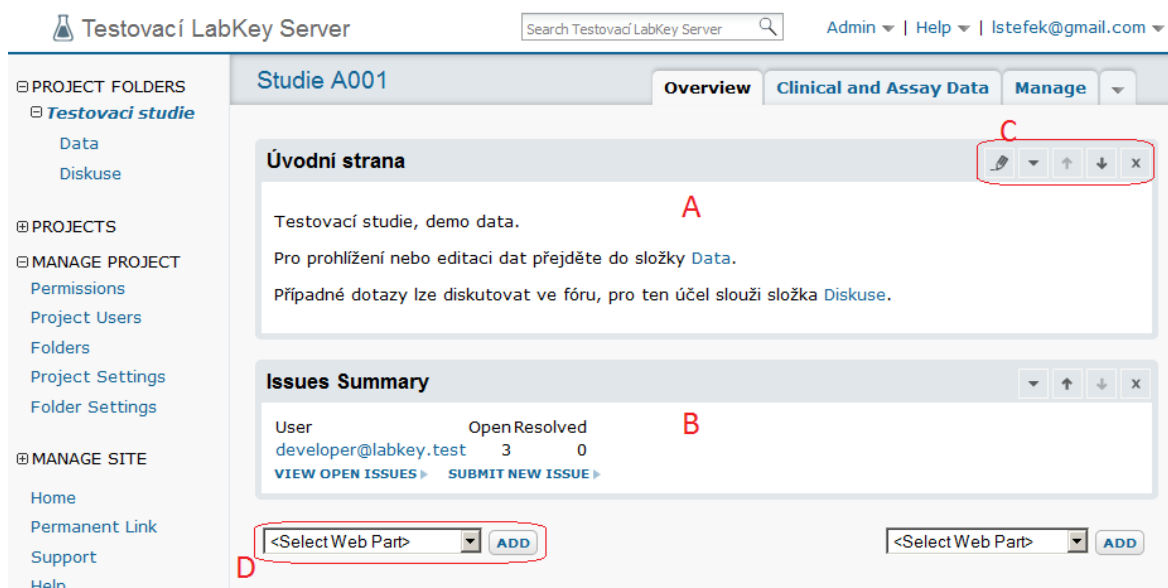
Na příkladu projektu „Testovací studie“ vidíme pod úrovní projekty složky „Data“ a „Diskuse“. Jak projekt, tak obě složky mají typ „Study“. Na obrázku 24 je stránka

pro administraci přístupů k projektu a k jednotlivým složkám. Uživatelé s administrátorskými oprávněními k projektu nebo složce, mohou měnit jejich obsah.



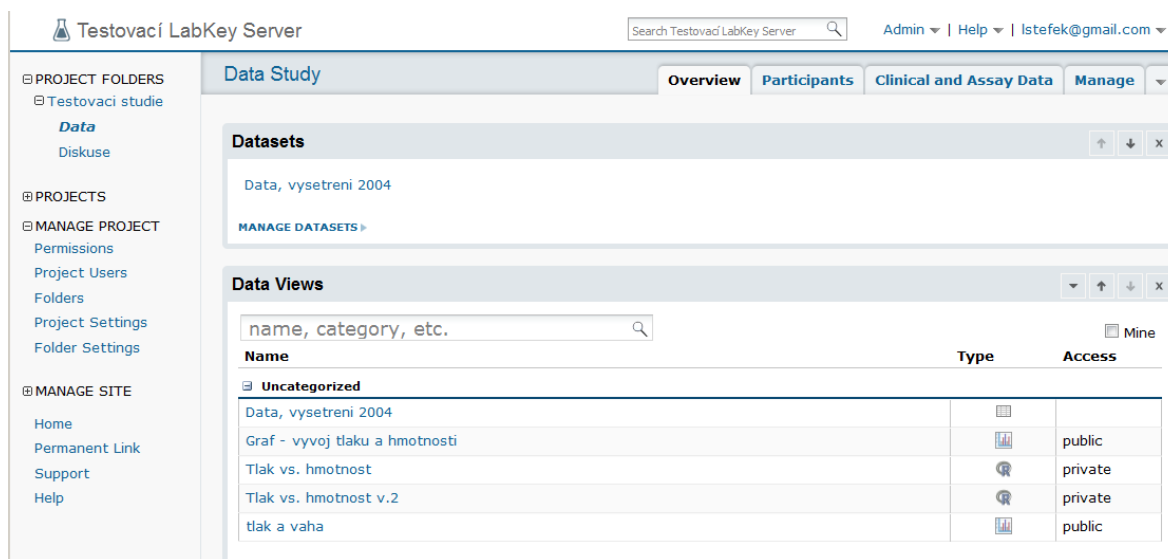
Obrázek 24. Přístupová práva k projektu a složkám

Nastavení obsahu jednotlivých složek může oprávněný uživatel provádět přímo na stránce té které složky. Způsob je podobný jako u některých webových portálů. Na obrázku 25 vidíme obsah složky „Testovací studie“, kde označené části jsou: A – Wiki, prvek s editovatelným HTML obsahem, B – Issues Summary, přehled hlášení z modulu Issues, C – editační prvky přístupné pouze administrátorovi složky a D – možnost přidat další prvek do stránky.



Obrázek 25. Obsah složky Testovací studie

Podobným způsobem je do složky „Data“ umístěn prvek „Datasets“, který spravuje

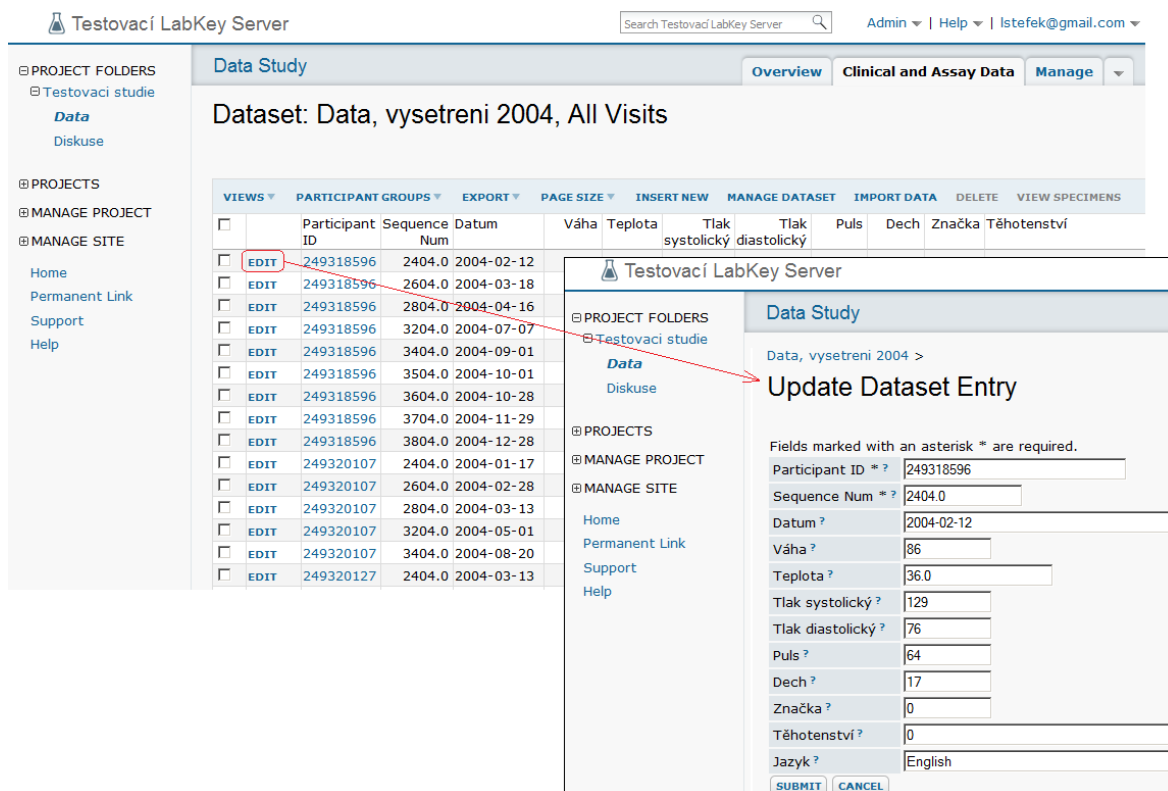


The screenshot shows the LabKey Server interface for a 'Data Study'. The left sidebar contains navigation options like 'PROJECT FOLDERS', 'PROJECTS', and 'MANAGE PROJECT'. The main content area is titled 'Data Study' and includes tabs for 'Overview', 'Participants', 'Clinical and Assay Data', and 'Manage'. Under 'Data Views', there is a search bar and a table listing various data views.

Name	Type	Access
Uncategorized		
Data, vysetreni 2004		
Graf - vyvoj tlaku a hmotnosti		public
Tlak vs. hmotnost		private
Tlak vs. hmotnost v.2		private
tlak a vaha		public

Obrázek 26. Obsah složky Data

data a „Data Views“, který poskytuje různé pohledy na tyto data (obrázek 26). Konkrétně jsou zde vzorová data ze základního lékařského vyšetření, získaná importem ze souboru ve formátu MS Excel. Na obrázku 27 vidíme jakým způsobem jsou zobrazena vlastní data a formulář pro editaci jednotlivého záznamu.



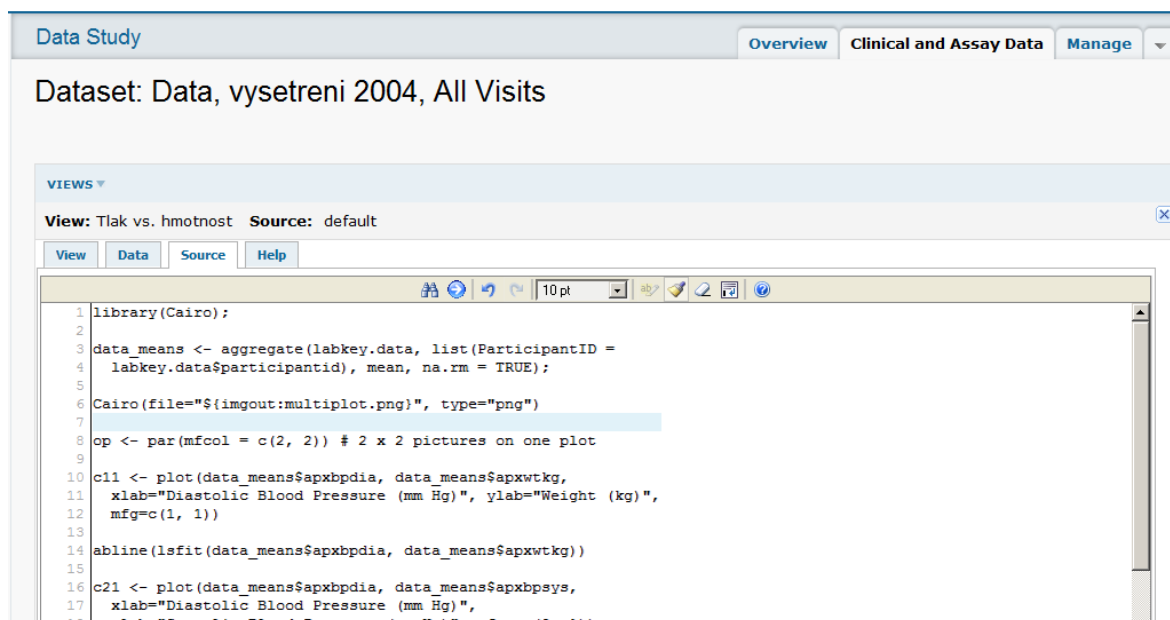
The screenshot shows the 'Dataset: Data, vysetreni 2004, All Visits' view. It features a table with columns for Participant ID, Sequence Num, Datum, Váha, Teplota, and various blood pressure and pulse measurements. An 'EDIT' button is highlighted in the first row. An inset window shows the 'Update Dataset Entry' form, which contains input fields for each column in the table, with asterisks indicating required fields.

Participant ID	Sequence Num	Datum	Váha	Teplota	Tlak systolický	Tlak diastolický	Puls	Dech	Značka	Těhotenství
249318596	2404.0	2004-02-12								
249318596	2604.0	2004-03-18								
249318596	2804.0	2004-04-16								
249318596	3204.0	2004-07-07								
249318596	3404.0	2004-09-01								
249318596	3504.0	2004-10-01								
249318596	3604.0	2004-10-28								
249318596	3704.0	2004-11-29								
249318596	3804.0	2004-12-28								
249320107	2404.0	2004-01-17								
249320107	2604.0	2004-02-28								
249320107	2804.0	2004-03-13								
249320107	3204.0	2004-05-01								
249320107	3404.0	2004-08-20								
249320127	2404.0	2004-03-13								

Obrázek 27. Zobrazení a editace dat

Pro prezentaci dat slouží v LabKey tzv. Data Views. Tyto pohledy mohou být

několika typů, od pouhé změny pořadí, viditelnosti sloupců, uplatnění filtrů a změny pořadí třídění, přes maticové zobrazení, až po zobrazení dat pomocí zpracování v jazyce JavaScript v prohlížeči, nebo zpracováním dat v R. A právě tato možnost je ukázána na následujících dvou obrázcích č. 28 a 29.

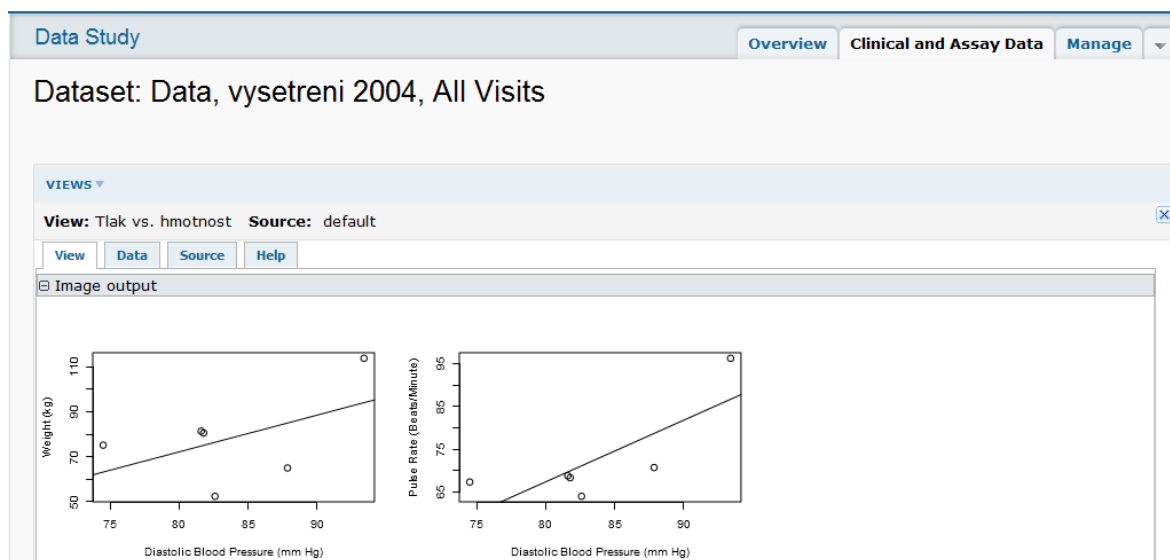


```

1 library(Cairo);
2
3 data_means <- aggregate(labkey.data, list(ParticipantID =
4   labkey.data$participantid), mean, na.rm = TRUE);
5
6 Cairo(file="imgout:multiplot.png", type="png")
7
8 op <- par(mfcol = c(2, 2)) # 2 x 2 pictures on one plot
9
10 c11 <- plot(data_means$apxwpdia, data_means$apxwtkg,
11   xlab="Diastolic Blood Pressure (mm Hg)", ylab="Weight (kg)",
12   mfg=c(1, 1))
13
14 abline(lsfite(data_means$apxwpdia, data_means$apxwtkg))
15
16 c21 <- plot(data_means$apxwpdia, data_means$apxwpsys,
17   xlab="Diastolic Blood Pressure (mm Hg)",
18   ylab="Systolic Blood Pressure (mm Hg)". mfg= c(2, 1))

```

Obrázek 28. Zápis kódu jazyka R v Labkey



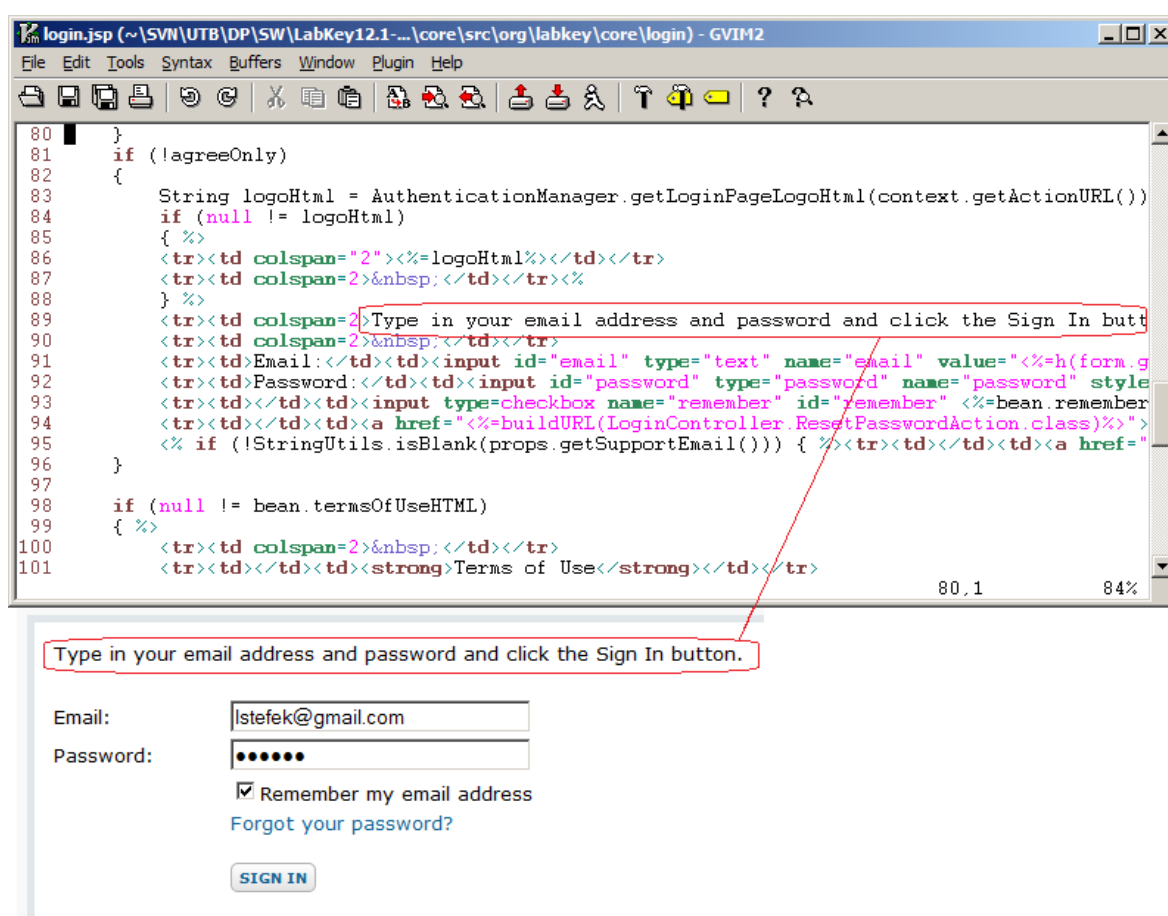
Obrázek 29. Výsledek zpracování dat z LabKey v R

5.6 LabKey – problémy a náměty na vylepšení

Asi jako každý programový produkt má i LabKey některá omezení a chyby. Některé z těch nejvíce viditelných jsou uvedeny v této podkapitole.

5.6.1 Lokalizace LabKey

Provozovat LabKey v jiném, než anglickém jazyku, není momentálně možné. Navíc, jak vyplývá z programového Java a JSP kódu, všechny textové řetězce, které se v aplikaci používají, jsou přímo zapsány v kódu. Není tedy použit žádný mechanismus pro tvorbu vícejazyčných webových aplikací. Je pravděpodobné, že pro současné použití LabKey převážně výzkumnými laboratořemi a univerzitami, nebyla možnost lokalizace aplikace do jiného jazyka vůbec požadována. Na obrázku 30 je příklad části zdrojového kódu ze souboru `login.jsp` a odpovídající část výsledné stránky v aplikaci.



Obrázek 30. Problém lokalizace – všechny texty přímo v kódu

Možným řešením by mohla být implementace doporučení pro tvorbu vícejazyčných webových aplikací vyvíjených v jazycích Java a JSP, které lze nalézt například v kapitole „Internationalizing and Localizing Web Applications“ v dokumentu „The Java EE 5 Tutorial“ [22]. Tato metoda spočívá, pro JSP soubory, v nahrazení anglických textů XML značkou/tagem `message` z JSTL knihovny. Původní text se uloží spolu s klíčem, který tento text jednoznačně identifikuje, do externího textového souboru, tzv. Re-

source Bundle. Alternativní řešení nabízí i samotný framework Spring, ale princip je zcela stejný. Pro texty z Java kódu je princip opět stejný, pouze se místo XML tagu použije volání metody `getString` ze třídy `ResourceBundle`.

Na příkladu na obrázku 30 by bylo nutno nahradit řádek 89 následujícím textem: `<tr><td colspan=2><fmt:message key="type_email_and_pwd"/></td></tr>` a zavést původní text do externího souboru s klíčem `type_email_and_pwd`.

5.6.2 Problémy s diakritikou

LabKey má na několika místech problémy s diakritikou, týká se to zejména názvů datové sady (dataset), kdy aplikace na některých místech chybně interpretuje znakový řetězec v kódování UTF-8 jako řetězec v kódování ISO8859-1. Toto lze vysvětlit pravděpodobně jako lokální chybu v Java kódu, LabKey totiž jinak bez problému pracuje s kódováním UTF-8 jak v databázi, tak na úrovni v HTML stránek.

5.6.3 Způsob editace dat

Způsob, kterým se v LabKey dají upravovat data není příliš pohodlný (obrázek 27). V případech, kdy je potřeba tvořit a editovat data přímo v LabKey, by byl vhodnější jiný způsob, například formou editovatelné tabulky.

5.6.4 Úroveň integrace s R

Způsob, jakým je integrován R do LabKey, není pro některé, zejména větší úlohy, ideální. Dávkový způsob zpracování, spolu s nutností inicializovat celé prostředí R pro každé spuštění „R View“, může být problematický pro větší objemy dat. Určitým zlepšením by mohlo být použití serveru RServe.

5.7 LabKey – závěr

Zdálo by se, že základní požadavky, kladené zadáním této práce, by produkt LabKey neměl mít problémy splnit. Bližší pohled však také odhalil několik slabých stránek tohoto produktu. Je otázkou, nakolik bude obtížné dosáhnout opravy chyb a prosazení potřebných změn v programovém kódu. Ať už vlastními silami, nebo za pomoci komunity stávajících vývojářů, případně zda bude možné se stát aktivním přispěvatelem projektu.

Na druhou stranu, otevřenost, se kterou je LabKey vyvíjen, otevřenost samotného produktu pro rozšiřování, kvalita dokumentace a volba technologií, které jsou pro vývoj používány, však činí projekt LabKey nadmíru zajímavým a perspektivním.

Pro LabKey také mluví jeho používání ve velkých projektech a s tím spojený neustálý vývoj, pravidelnost vydávání nových verzí a množství funkcionality, která zde nebyla blíže přiblížena. Přes velké množství funkcionality, zřejmě díky své modularitě, působí vnitřní struktura produktu a jeho zdrojový kód poměrně srozumitelně. To vše, spolu s již zmíněnou kvalitou dokumentace, dává projektu LabKey dobré předpoklady pro jeho další uplatnění.

ZÁVĚR

Aplikace informačních technologií v oblasti statistické a datové analýzy, navíc v prostředí biomedicíny, lze asi právem považovat za jednu z nejkompexnějších oblastí informatiky. Nejenže se tu setkávají různé obory, statistika, všechny oblasti informatiky, ale nadto, samotná data mají velmi vysokou hodnotu a informace z nich získané, pak hodnotu zlata. Není nutné zdůrazňovat, že získat *správné* informace, nemusí být vůbec snadné a vyžaduje to značnou odbornost ve zkoumané problematice.

Cílem této práce bylo navrhnout vhodnou technologii a realizovat systém pro spouštění aplikačních úloh se systémem R.

V první, teoretické části této práce, jsou proto uvedeny dvě kapitoly, shrnující jednak základní vlastnosti prostředí R a také jednotlivá existující grafická uživatelská rozhraní tohoto systému. Pokud by měly být tyto kapitoly použitelné samostatně, jistě by zasloužily rozšířit, ale pro potřeby této práce již poskytují určitý základní, snad také dostačující přehled.

V praktické části jsou uvedeny dvě hlavní kapitoly (kap. 4 a 5), které prezentují dvě možné cesty k dosažení cílů této práce. A zatímco aplikace RWeb zůstala vývojově ve fázi prototypu, druhá cesta, použití existujícího softwarového produktu LabKey se dočkala podrobnějšího rozpracování. LabKey sice nelze ani zdaleka zařadit do kategorie produktů „GUI pro R“, ale vzhledem k aktuálním požadavkům zadavatele, jej lze doporučit k postupnému nasazení.

ZÁVĚR V ANGLIČTINĚ

Application of Information Technology in the area of statistical data analysis, especially for biomedical research, can be considered as one of the most complex areas of Computer Science. This is a meeting point of different disciplines including Statistics and Information Technology, but moreover, the data itself has a very high value and information derived from it is then even more valuable. It is not necessary to emphasize that to get correct information out of the data is not an easy task at all.

The aim of this thesis was to choose and implement appropriate technology and also to design system for the execution of application tasks in the R.

In the theoretical part of this thesis there are presented two chapters summarizing the basic features of R and also summarizing various existing graphical user interfaces for R. In case these chapters would be used separately they would need some more detailed information to include, but for the purpose of this thesis, these chapters already provide sufficient overview.

The practical part contains two main chapters (chapter 4 and 5) representing two possible ways to achieve the objectives of this thesis. The first one presents the application RWeb which is rather a prototype. The second one is evaluating an existing software product LabKey and it has been elaborated in more detail. Although LabKey don't fit into category "GUI for R", it is the product which could help to resolve current requirements of the customer.

Reference

- [1] *The R Journal* [online casopis]. 2009- [cit. 2012]. Dostupné z: <http://journal.r-project.org/>. ISSN: 2073-4859.
- [2] *R News* [online casopis]. 2001-2008 [cit. 2012]. Dostupné z: <http://journal.r-project.org/>. ISSN: 1609-3631.
- [3] DALGAARD, Peter. *Introductory Statistics with R*. New York: Springer, 2002. ISBN 0387954759.
- [4] The R Foundation. *The R Project for Statistical Computing* [online]. [cit. 2012]. Dostupné z: <http://www.r-project.org/>.
- [5] The R Foundation. *The Comprehensive R Archive Network* [online]. Dostupné z: <http://cran.r-project.org/>.
- [6] TIBCO Software Inc. *TIBCO Spotfire S+* [online]. Dostupné z: <http://spotfire.tibco.com/products/s-plus/statistical-analysis-software.aspx>
- [7] DROZD, P. *Cvičení z biostatistiky: Základy práce se softwarem R*. [online] Ostrava: 2007. 111 s. ISBN 978-80-7368-433-4. [cit. 2012-05-10]. Dostupné z: <http://cran.r-project.org/doc/contrib/CviceniR1.pdf>
- [8] KONEČNÁ, Kateřina. *Výuka jazyka R* [online]. Brno, 2010 [cit. 2012-05-14]. Dostupné z: http://is.muni.cz/th/270073/prif_b/Bakalarska_prace.pdf. Bakalářská práce. Masarykova Univerzita.
- [9] Fox, John. The R Commander. *Journal of Statistical Software*. A Basic-Statistics Graphical User Interface to R [online]. 2005, roč. 14, č. 9. ISSN: 1548-7660.[cit. 2012-05-20]. Dostupné z: <http://www.jstatsoft.org/>.
- [10] DALGAARD, Peter. A Primer on the R-Tcl/Tk Package. *R News* [online]. 2001, č. 3. [cit. 2012-05-20]. Dostupné z: <http://journal.r-project.org/>. ISSN: 1609-3631.
- [11] DALGAARD, Peter . The R-Tcl/Tk interface. In: *Proceedings of the 2nd International Workshop on Distributed Statistical Computing*, March 15-17, 2001, Technische Universität Wien, Vienna, Austria, 2001. ISSN 1609-395X. Dostupné z: <http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/>.

- [12] ROBISON-COX, James. Putting RGtk to Work. In: *3rd International Workshop on Distributed Statistical Computing* [online]. TUW, Vienna, Austria, 2003 [cit. 2012-05-05]. ISSN 1609-395X. Dostupné z: <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/RobisonCox.pdf>
- [13] *The RStudio Integrated Development Environment* [online]. Dostupné z: <http://www.rstudio.org/>
- [14] Williams, G.J. Rattle: A Data Mining GUI for R. *The R Journal*. Vol. 1, pages 45–55, december 2009. [cit. 2012-05-20]. Dostupné z: http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf. ISSN: 2073-4859.
- [15] URBANEK, Simon. Rserve – A Fast Way to Provide R Functionality to Applications. In: *3rd International Workshop on Distributed Statistical Computing* [online]. TUW, Vienna, Austria, 2003 [cit. 2012-05-15]. ISSN 1609-395X. Dostupné z: <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Urbanek.pdf>
- [16] Information Builders. *WebFOCUS RStat: Predict the Future and Make Effective Decisions Today*. [online]. New York, 2010. [cit. 2012-05-20]. Dostupné z: http://www.informationbuilders.com/products/webfocus/pdf/RStat_FS.pdf
- [17] *Deducer: A GUI for R - Deducer Manual* [online]. 2012, April 16, 2012 [cit. 2012-05-13]. Dostupné z: <http://www.deducer.org>
- [18] JGR: JAVA GUI FOR R. *Statistical Computing & Statistical Graphics* [online]. 2005, roč. 16, č. 2, str. 9–12 [cit. 2012-05-13]. Dostupné z: <http://stat-computing.org/newsletter/issues/scgn-16-2.pdf>
- [19] *Red-R: visual programming for R* [online]. Sep. 21, 2009 [cit. 2012-05-13]. Dostupné z: <http://www.red-r.org/documentation>
- [20] VISNE, Ilhami, Erkan DILAVEROGLU, Klemens VIERLINGER, Martin LAUSS, Ahmet YILDIZ, Andreas WEINHAEUSEL, Christa NOEHAMMER, Friedrich LEISCH a Albert KRIEGNER. RGG: A general GUI Framework for R scripts. *BMC Bioinformatics* [online]. 2009, roč. 10, č. 1 [cit. 2012-05-12]. ISSN 1471-2105. DOI: 10.1186/1471-2105-10-74. Dostupné z: <http://www.biomedcentral.com/1471-2105/10/74>

- [21] NELSON, Elizabeth K, Britt PIEHLER, Josh ECKELS, Adam RAUCH, Matthew BELLEW, Peter HUSSEY, Sarah RAMSAY, Cory NATHE, Karl LUM, Kevin KROUSE, David STEARNS, Brian CONNOLLY, Tom SKILLMAN a Mark IGRA. LabKey Server: An open source platform for scientific data integration, analysis and collaboration. *BMC Bioinformatics* [online]. 2011, roč. 12, č. 1 [cit. 2012-05-12]. ISSN 1471-2105. DOI: 10.1186/1471-2105-12-71. Dostupné z: <http://www.biomedcentral.com/1471-2105/12/71>
- [22] JENDROCK, Eric, Jennifer BALL, Debbie CARSON, Ian EVANS, Scott FORDIN a Kim HAASE. *The Java EE 5 Tutorial: For Sun Java System Application Server 9.1* [online]. September 2010. [cit. 2012-05-17]. Dostupné z: <http://docs.oracle.com/javaee/5/tutorial/doc/index.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML
CLI	Command Line Interface - rozhraní příkazového řádku
DCOM	Distributed Component Object Model
GNU	GNU's Not Unix! - a recursive acronym
GNU GPL	GNU General Public License
GUI	Graphical User Interface - grafické uživatelské rozhraní
GWT	Google Web Toolkit
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JDK	Java Development Kit
JRE	Java Runtime Environment
JSP	Java Server Pages
JSTL	JavaServer Pages Standard Tag Library
KDE	The K-Desktop Environment
LDAP	Lightweight Directory Access Protocol
LGPL	Lesser General Public License
.NET	The .NET Framework je vývojová platforma od firmy Microsoft
SVN	Verzovací systém Apache Subversion
Tcl	Tool Command Language
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

Seznam obrázků

Obr. 1. R ve Windows	16
Obr. 2. R v linuxovém terminálu	17
Obr. 3. Příklady některých typů grafů v R	27
Obr. 4. R Commander: Hlavní okno aplikace	31
Obr. 5. Dialogová okna při instalaci R Commanderu	33
Obr. 6. Poor Man's GUI – pmg.....	35
Obr. 7. RStudio v linuxu	37
Obr. 8. RKward v linuxu.....	39
Obr. 9. Rattle: okno aplikace a R konzola.....	41
Obr. 10. RExcel – Integrace R a MS Excel.....	43
Obr. 11. JGR – Java GUI for R.....	44
Obr. 12. Deducer – lineární regrese	46
Obr. 13. JGR: Package Manager – jak automaticky spouštět Deducer	47
Obr. 14. Výsledek ukázkového příkladu 2.12.1	49
Obr. 15. Výsledek ukázkového příkladu RGtk 2.12.2.....	51
Obr. 16. Náčrt architektury aplikace RWeb	55
Obr. 17. Obrazovka aplikace RWeb	56
Obr. 18. LabKey Server	58
Obr. 19. LabKey Server – klíčové komponenty	59
Obr. 20. Instalace LabKey Server – závěr instalace	61
Obr. 21. Administrace LabKey – založení uživatelů.....	61
Obr. 22. LabKey Server – konfigurace R Scripting Engine.....	63
Obr. 23. Vytvoření projektu, volba jeho typu a modulů	64
Obr. 24. Přístupová práva k projektu a složkám	65
Obr. 25. Obsah složky Testovací studie	65
Obr. 26. Obsah složky Data	66
Obr. 27. Zobrazení a editace dat	66
Obr. 28. Zápis kódu jazyka R v Labkey	67
Obr. 29. Výsledek zpracování dat z LabKey v R	67
Obr. 30. Problém lokalizace – všechny texty přímo v kódu.....	68

Seznam tabulek

Tab. 1. Základní operátory v R.....	19
Tab. 2. Základní matematické funkce v R.....	19
Tab. 3. Operátory a funkce pro manipulaci s maticemi	21
Tab. 4. Deducer: přehled položek menu.....	47
Tab. 5. Některé editory pro R.....	52
Tab. 6. Některé další nástroje pro R	52