

University of Žilina

2nd International Conference on Open Source tools in Computer Science university education Conference Proceedings

Rudolf Blaško (Ed.)
University of Žilina

Project: Innovative Open Source Courses
for Computer Science Curriculum



29. – 30. 6. 2022



ŽILINSKÁ UNIVERZITA
V ŽILINE

Reviewer: prof. RNDr. Vojtech Bálint, CSc., RNDr. Michal Kaukič, vCSc.,
doc. Ing. Miroslav Kvaššay, PhD., University of Žilina, Ing. Pavel Stríž, Ph.D.,
CSTUG Praha

Project: Innovative Open Source Courses for Computer Science Curriculum

© University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia

ISBN 978-80-554-1941-1



Open Access. This book is licensed under the terms of the Creative Commons
Attribution-ShareAlike 4.0 International License, CC BY-SA 4.0

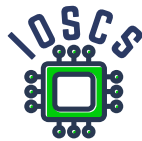
(<https://creativecommons.org/licenses/by-sa/4.0/>)

University of Žilina

2nd International Conference on Open Source tools in Computer Science university education Conference Proceedings

Rudolf Blaško (Ed.)
University of Žilina

Project: Innovative Open Source Courses
for Computer Science Curriculum



29. – 30. 6. 2022



ŽILINSKÁ UNIVERZITA
V ŽILINE

Project: Innovative Open Source Courses for Computer Science Curriculum



This material was created as one of the activity of the project “Innovative Open Source Courses for Computer Science Curriculum”, funded by the Erasmus+ grant nr. 2019-1-PL01-KA203-065564.

Project information

Project was implemented under the Erasmus+.

Name: “Innovative Open Source courses for Computer Science curriculum”

Nr: 2019-1-PL01-KA203-065564

Key Action: KA2 – Cooperation for innovation and the exchange of good practices

Action Type: KA203 – Strategic Partnerships for higher education

Consortium

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Mendelova Univerzita v Brně

Žilinská Univerzita v Žiline

Erasmus+ Disclaimer

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Copyright Notice

This content was created by the IOSCS consortium: 2019–2022.

The content is Copyrighted and distributed under Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).



Co-funded by the
Erasmus+ Programme
of the European Union

Obsah

Foreword	5
Radosław Maciaszczyk Project information	7
Ondřej Vencálek Statistika a jiné zbraně	12
Jiří Rybička Testy v kurzu Nástroje Open source pro zpracování textů	17
Rudolf Blaško Integration with wxMaxima	26
Mirosław Łazoryszczak GNU Radio and SDR platforms	41
Marcel Puškarik Interaktívna webová aplikácia pre výber a štatistické spracovanie otvorených datasetov	48
Radosław Maciaszczyk Android – Architecture Pattern	56
Ivan Baroňák, Michal Vančo Rocket.chat	62
Eva Rakovská Prečo učím CLIPS, aj keď sa to zdá mnohým staromódne	63
Aleš Kozubík Pravdepodobnosť a štatistika v R Základy grafiky	70

Foreword

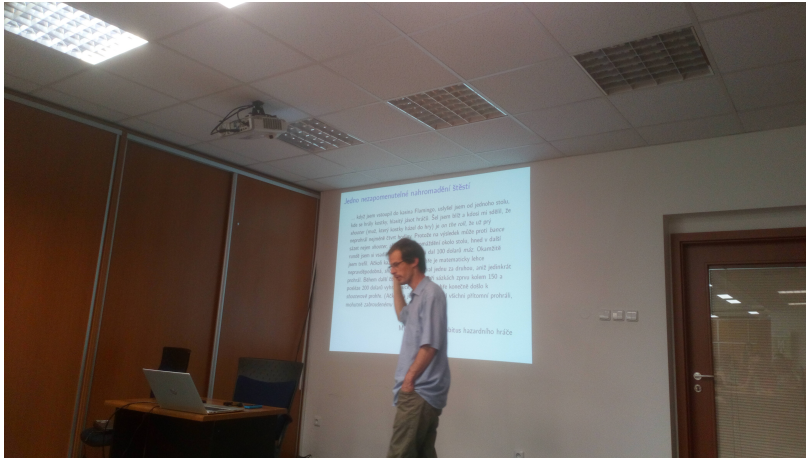
Multiplier events – conferences are one of the main activities of the project “Innovative Open Source Courses for Computer Science Curriculum”. The second of the two planned conferences took place on June 29 and 30, 2022 in Information centre at Faculty of Management sciences and Informatics of University of Žilina. The working languages of the conference were English, Czech and Slovak.



A “2nd International Conference on Open Source tools in Computer Science university education” included papers focused on project information, but also a section of papers devoted to more general aspects of open source software – Rocket.Chat on Ubuntu, Programming teaching by PygameZero, NodeRED tool for internet of things and statistical analyses by open source software.

Due to major restrictions in the ongoing Covid-19 pandemic, as many speakers and listeners as would have been appropriate could not participate in the conference. Nevertheless, it was possible to reach the number of circa 40 participants and give the conference the necessary weight.

The conference was divided into two days. On the first day, contributions related to general information about the project and contributions from the extended context of open source programs were heard, while on the second day contributions focused on clarifying the concepts of individual courses that are to be created within the project.



This proceedings summarizes all contributions on the conference. The main section contains the contributions of the first day, the second section called concentrates the contributions presented on the second day.

We have attached photo to remind you of the atmosphere of the conference.

Jiří Rybička

Project information

Radosław Maciaszczyk

Abstract.

Open-source software provides an alternative to commercial software, but awareness of this alternative is low. The project "Innovative Open Source courses for Computer Science curriculum" represents a confident response to the popularisation of open source software in teaching higher education subjects. This article presents the results of an evaluation of the summer school courses developed as part of this project.

Abstrakt.

Oprogramowanie otwarte źródłowe stanowi alternatywę dla oprogramowania komercyjnego, jednak świadomość istnienia tej alternatywy jest dość niska. Projekt "Innovative Open Source courses for Computer Science curriculum" stanowi pewną odpowiedź na popularyzację oprogramowania open-source w kształceniu w szkolnictwie wyższym. W niniejszym artykule przedstawiono wybrane wyniki ewaluacji kursów szkoły letniej opracowanych w ramach tego projektu.

Abstrakt.

Open source softvér poskytuje alternatívu ku komerčným softvéru, ale povedomie o tejto možnosti je nízke. Projekt „Innovative Open Source courses for Computer Science curriculum“ predstavuje vhodnú odpoveď na popularizáciu Open source softvéru s otvoreným zdrojovým kódom vo výučbe predmetov vysokoškolského vzdelávania. Tento článok predstavuje výsledky hodnotenia kurzov letných škôl uskutočnených v rámci tohto projektu.

Abstrakt.

Open source software poskytuje alternativu ke komerčnímu softwaru, ale povědomí o této možnosti je nízké. Projekt „Innovative Open Source courses for Computer Science curriculum“ představuje vhodnou odpověď na popularizaci Open source softwaru s otevřeným zdrojovým kódem ve výuce předmětů vysokoškolského vzdělávání. Tento článek představuje výsledky hodnocení kurzů letních škol uskutečněných v rámci tohoto projektu.

Key words: Open Source, introduction, Computer Science.

Introduction

The project “Innovative Open Source courses for Computer Science curriculum” – (IOSCS) is funded under EU Erasmus+ Programme in Key-Action 2 “Cooperation for innovation and the exchange of good practices”, Action Type – “Strategic Partnerships for higher education”. Project duration is from 1st of September 2019 until 31st of December 2022. The project is coordinated by West Pomeranian University of Technology in Szczecin (Poland) and is implemented in partnership with Mendel University in Brno (Czech Republic) and University of Žilina (Slovak Republic).

Consortium:

- Zachodniopomorski Uniwersytet Technologiczny w Szczecinie – ZUT (PL),
- Mendelova univerzita v Brně – MENDELU (CZ),
- Žilinská univerzita v Žiline – UNIZA (SK).

Teams’ leaders:

- ZUT dr inż. Radosław Maciaszczyk
- MENDELU doc. Ing. Jiří Rybička. Ph.D.
- UNIZA RNDr. Aleš Kozubík, PhD.

Main goal of the project is to develop 6 study courses, based on Open Source approach. The courses will cover fields of programming, mathematics, operating systems, embedded systems, systems controlled by computer and other hardware and/or software applications.

The results of IOSCS project will be:

- descriptions of the courses,
- materials necessary for lectures and practical classes,
- handbook composed of the materials necessary to follow the courses by the students.

Selected courses:

- Mathematical Analysis supported by wxMaxima [UNIZA].
- Probability and statistics with programming in R [UNIZA].
- Open Source tools for text processing [MENDELU].

- Programming language Lua [MENDELU].
- Wireless Signal Processing in GNU Radio Environment [ZUT].
- Mobile Application Development [ZUT].

1 Summer intensive courses

The courses are to be tested during two summer intensive courses. The first summer school was held in Žilina, and the second was in Brno. Each lasted five working days. For four days, students studied courses of their choice. The last day was dedicated to evaluation and assessment.

1.1 Summer school in Žilina

The first summer intensive school took place in Žilina from 6. 9. – 10. 9. 2021. It was the period of the pandemic and there were restrictions on the organisation of meetings. There were 25 students in attendance, 10 students from Szczecin, 11 students from Brno (2 remotely) and 4 students from Žilina. The classes were taught by five lecturers. Each course was designed for 10 h of classes, with students declaring participation in 2 course.

Choice of courses:

- Open Source tools for text processing – 8 students,
- Programming language Lua – 10 students,
- Mobile Application Development – 10 students,
- Mathematical Analysis supported by wxMaxima – 5 students,
- Probability and statistics with programming in R – 10 students,
- Wireless Signal Processing in GNU Radio Environment – 7 students.

At the end of the course, progress was discussed and students gave feedback on the courses, an evaluation of the course was made. Students answered the following questions:

1. What should be improved in the materials?
2. Was the material presented sufficient to understand the issues?
3. What should be supplemented?
4. To what extent have you improved your knowledge and skills during the course?

5. Have you found open-source tools useful and inspiring?
6. Rate the quality of the prepared material (1–6 where 6 is the best).

In response to the first question, there were issues related to increasing the time for the material, making the final results available for comparison, or increasing the readability of some of the material (screenshots). For the second question, most answers indicated the materials' readability. The courses' tools and scope were new material for most students, so the third question was often answered with a lack of opinion. In the fifth, students indicated that the material allowed them to increase their level of knowledge. In response to question five regarding knowledge of open-source tools, the most frequent response was lack of knowledge. For question six, the vast majority of respondents answered in the affirmative.

In the last question, students evaluated the prepared materials; the average of all responses was: 5.065. The responses received were discussed at meetings in the consortium. Some of the comments received were incorporated into the materials. These were reviewed during the second summer school.

1.2 Summer school in Brno

The second summer intensive school took place in Brno from 30. 5. – 3. 6. 2022. 43 students participated in this edition, including 12 from Szczecin, 12 from Žilina and 19 from Brno. Six lecturers taught classes. Identical to the previous edition, students made their choice of courses:

- Open Source tools for text processing – 7 students,
- Programming language Lua – 14 students,
- Mobile Application Development – 20 students,
- Mathematical Analysis supported by wxMaxima – 7 students,
- Probability and statistics with programming in R – 18 students,
- Wireless Signal Processing in GNU Radio Environment – 10 students.

In the end, students wrote a knowledge test in each subject and answered evaluation questions. The average rating of the courses in this edition is higher and was: 5.27. It is also noticeable that the number of comments on individual courses has decreased. However, students pointed out that the presentation of assignments with application examples or the extension of the essential topics should be increased. According to the lecturers, the collected evaluation results are satisfactory, while the detailed answers will

help to improve the prepared materials. Some of the material was improved after the first summer school. This can be seen in the outcome of the second summer school, where, despite a much higher number of students, the evaluation obtained is higher. The comments from the second summer school will improve the existing materials and will also be taken into account in creating the book of materials for the individual courses. The student's course choices are also worth noting. Based on the two editions of the summer school, the most popular course was Mobile Application Development, followed by Probability and statistics with programming in R. In contrast, the most miniature popular courses were Mathematical Analysis supported by wxMaxima and Wireless Signal Processing in GNU Radio Environment. There is less interest in the underlying issues, so it is important to introduce attractive methods of presenting them. It is also important to use additional tools to facilitate the course, such as those used in the project. Given that the courses are intended for students of Computer Science, and the tools used are open source, it can be hoped that there will be students interested in developing these tools, whether as part of their classes, study groups or interests. The final point is that the summer school students going into the course were not familiar with open-source tools. The summer schools have created interest and increased knowledge of open-source software.

2 Conclusions

The summer schools showed that knowledge of open-source software is rare. It also showed that projects such as this are needed and that it is worth putting more effort into spreading knowledge in this area. Particularly among students studying computer science. At the same time, the summer schools allowed students to evaluate the courses. The evaluations received are satisfactory, while critical comments will allow the content presented to be improved.

Radosław Maciaszczyk, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin, ul. Żołnierska 49, 71-210 Szczecin, Poland, e-mail: rmaciaszczyk@zut.edu.pl

Statistika a jiné zbraně

Ondřej Vencálek

Abstrakt.

Statistika se dá velmi dobře zneužít k demagogii, navíc závěry jsou přece podloženy údaji. Se statistikou je to stejné, jako s jakýmkoliv jiným nástrojem. Záleží na tom, kdo ho drží v ruce a jak s ním umí zacházet. Cílem tohoto příspěvku je ukázat několik příkladů chybných úsudků vyvozených z pozorování reality a zdánlivě podložených objektivními údaji.

Abstract.

Statistics can very well be misused for demagoguery, moreover, the conclusions are supported by data. It's the same with statistics as with any other tool. It depends on who is holding it and how they can handle it. The goal of this post is to show several examples of erroneous judgments derived from observation of reality and seemingly supported by objective data.

Klíčová slova: Statistika, histogram, graf.

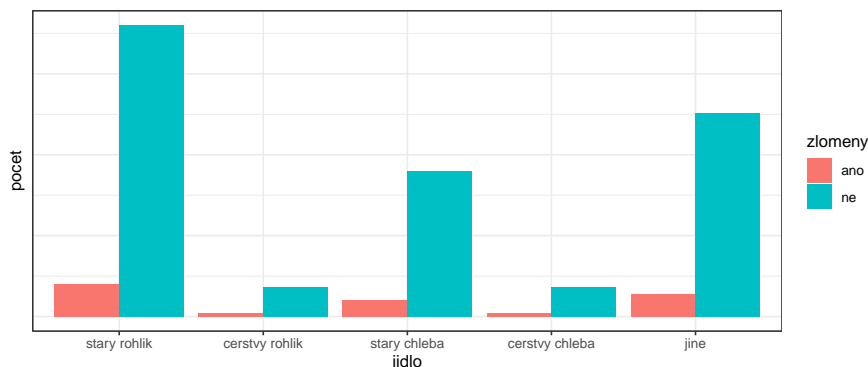
1 Úvod – statistika jako zbraň

„Statistika se dá velmi dobře zneužít k demagogii, navíc závěry jsou přece podloženy údaji, že? ... Se statistikou je to stejné, jako s jakýmkoliv jiným nástrojem. S nožem si můžete stejně tak ukrojit chleba, jako ho bodnout někomu do zad. Záleží pochopitelně na tom, kdo ho drží v ruce a jak s ním umí zacházet.“ Těmito slovy varovali před zneužitím statistiky autoři publikace *Moderní metody zpracování dat*, [3]. Ačkoliv od jejího vydání v roce 1992 uplynulo letos již třicet let, toto varování neztrácí na své aktuálnosti.

Cílem tohoto příspěvku je ukázat několik příkladů chybných úsudků vyvozených z pozorování reality a zdánlivě podložených „objektivními“ údaji.

2 Příklad ulomeného zubu

Představme si zkušeného zubaře, který shrne svá pozorování do následujícího tvrzení: „Lidé si nejčastěji vyloží zub na včerejším (starém) rohlíku.“ Toto tvrzení by mohl podpořit daty – každého pacienta, který přichází s „ulomeným zubem“ by se zubař zeptal, jakým způsobem k ulomení došlo, a odpověď by zaznamenal. Poté by vyhodnotil, zda opravdu nejčastější příčinou ulomení zubu byla konzumace starého rohlíku. Pokud by tomu tak



Obrázek 1: Fiktivní záznamy o nebezpečnosti konzumace různých druhů pečiva.

skutečně bylo, mohli bychom snadno učinit následující závěr v podobě praktického doporučení: „Chcete-li se vyhnout potížím s ulomenými zuby, nejezte staré rohlíky.“ Takový závěr učiníme, pokud si budeme myslet, že konzumace starých rohlíčků je pro náš chrup nebezpečnější než konzumace čerstvých rohlíčků či jiného pečiva. Takový závěr však z výše uvedených dat nevyplývá. To, že je nejčastější příčinou ulomení zubu konzumace starého rohlíku, může totiž znamenat jen to, že hodně lidí snídá staré rohlíky. Představme si, že konzumace starých rohlíčků je stejně nebezpečná jako konzumace jiného pečiva, ale že nejvíce lidí jí právě starý rohlík. Těto situaci odpovídá (fiktivní) graf 1.

Poměr výšky červeného sloupce ku výšce modrého sloupce je ve všech případech (pro všechny typy pečiva) stejný, tedy konzumace jednotlivých druhů pečiva je stejně nebezpečná. Zubař ve skutečnosti pozoruje jen případy, kdy došlo k ulomení zubu, vidí tedy jen červené sloupce. Z nich nejvyšší je opravdu ten odpovídající případům, kdy došlo k vylomení zubu na starém rohlíku. Důvodem v tomto případě ovšem nebyla vyšší nebezpečnost tohoto pečiva, ale vyšší konzumace (i modrý sloupec odpovídající konzumaci starého rohlíku je nejvyšší mezi modrými sloupci). Z pozorování pouze červených sloupců nelze nic říci o nebezpečnosti konzumace jednotlivých typů pečiva.

3 Případ zasažených letadel

Přidejme ještě jeden slavný příklad toho, že neméně důležité než to, co data obsahují, je také to, co data neobsahují.

Během druhé světové války působila na Columbia University v New Yorku

výzkumná statistická skupina, která měla mimo jiné za úkol analyzovat data o vojenských letadlech, která během operací utrpěla zásah. Cílem analýz bylo zjistit, které oblasti jsou zasaženy nejčastěji. Tyto oblasti měly být následně důkladněji opancérovány.

Bylo zjištěno, že nejvíce zásahů se nachází v oblasti křídel a ocasních ploch, naopak málo zásahů bylo zjištěno v oblasti motoru. Teprve Abraham Wald upozornil na to, že právě oblast motoru je zřejmě kritická, neboť letadla zasažená do této části se nevrátila na základnu a nemohla být proto zařazena do analýzy. Wald vlastně upozornil na tzv. zkreslení přežití (survivorship bias), viz [2]

4 Případ průměrného věku doktorů

Další příklad toho, jak lze statistiku (ať už záměrně či z pouhé nevědomosti) zneužít k demagogii nabízí kniha Tomáše Kašpara *Krutá hra o tvé zdraví* [4]. Autor v ní uvádí: „Naším hlavním zdrojem informací o výživě a zdraví jsou obvykle lékaři. Jich se ptají novináři na názor, když vznikne nějaký spor. Nikdo se však již nedívá na to, že většina lékařů žije nezdravým životním stylem a že se dožívají v průměru nejkratšího věku (průměrný věk lidí v USA je 75,5 let, průměrný věk lékařů v USA je jen 58 let – o 17 let méně!) [...] Od koho se chcete učit být zdraví: Od zdravých, dlouho žijících lidí, nebo od lidí s průměrně nejkratším životem?“

Zde jsou porovnávány dva nesrovnatelné údaje: prvním je tzv. *střední délka života* neboli naděje dožití při narození. Tento ukazatel říká, kolik let v průměru ještě prožije právě narozený jedinec, pokud by po zbytek jeho života byla úmrtnost stejná jako v roce, pro který je tento ukazatel počítán. V České republice byla v roce 2020 hodnota tohoto ukazatele přibližně 75,3 let pro muže a 81,4 let pro ženy, [5]. V USA byla hodnota tohoto ukazatele (souhrně bez rozlišení podle pohlaví) přibližně 77 let¹. Při výpočtu tohoto čísla vycházíme vždy z věku, ve kterém došlo k úmrtí.

Druhým údajem je průměrný věk lékařů. Ten spočteme tak, že vezmeme všechny žijící (!) lékaře, u každého zjistíme jeho věk a tyto hodnoty průměrujeme. V roce 2020 bylo v USA asi 1,02 miliónu lékařů. Jejich průměrný věk byl 51,7 let, [9]. Toto číslo není nijak překvapivé. Lékařů mladších 30 let je poměrně málo (necelá 2 %) – jsou to lékaři těsně po vystudování. Na druhé straně praktikujících lékařů ve věku nad 70 let je necelých 9 %. Mnoho lékařů odchází již dříve do důchodu. Poznamenejme, že průměrný věk v České republice, pokud uvažujeme všechny obyvatele, byl v roce 2020 pro muže 41,1 roků a pro ženy 44,0 roků [5].

¹<https://data.worldbank.org/indicator/SP.DYN.LE00.IN>

Rozdílu mezi výše uvedenými ukazateli je věnována kapitola *Naděje dožití a průměrný věk* knihy Terezy Košťákové *O složitém jednoduše* [6]. Zájemce odkazujeme právě na tuto publikaci vydanou Českým statistickým úřadem a volně dostupnou na jeho stránkách.

5 Případ konfirmačního zkreslení

Výše popsanou chybnou argumentaci, jíž chtěl autor přesvědčit své čtenáře o tom, že doktoři (lékaři) nejsou dobrými rádci ohledně zdraví, můžeme považovat za názornou ukázkou tzv. konfirmačního zkreslení. Jde o situaci, kdy svůj stávající názor na lékaře autor podpořil „daty“, aniž by se kriticky zamyslel nad nesmyslností srovnání, které provádí.

O tendenci přeceňovat „důkazy“ svých stávajících stanovisek uvažoval již před více než 400 lety filosof Francis Bacon, když ve svém díle *Nové organon* [1] napsal: „Jakmile člověk zaujme nějaké stanovisko, sbírá všechny důkazy, které ho potvrzují, a i když důkazy svědčící o opaku mohou být častější a závažnější, buď si jich nevšimne nebo je zavrhne, aby víra v zaujaté stanovisko nebyla otřesena.“

Leonard Mlodinow ve své knize *Život je jen náhoda* [8] poznamenává: „Když jsme v zajetí nějaké iluze – nebo prostě když máme nějaký nový nápad – tak místo abychom hledali možnost, jak dokázat, že naše představa neplatí, obvykle se pokoušíme dokázat, že je správná. Psychologové došli k názoru, že jsme zatíženi na hledání souhlasu s našimi názory...“. Mlodinow dále cituje studii [7], v níž experimentátoři zkoumali skupinu studentů, z nichž někteří byli přesvědčeni o správnosti trestu smrti a druzí o opaku. Všichni studenti obdrželi ke studiu stejný odborný materiál o „efektivnosti“ trestu smrti. Polovina studií ukazovala na odstrašující efekt, druhá polovina jej popírala. Výzkumníci studenty upozornili na možné slabiny v logice studií. Studenti měli jednak posoudit kvalitu studií, a následně posoudit, do jaké míry studium problematiky ovlivnilo jejich názor na trest smrti. Ukázalo se, že studenti lépe hodnotili studie, které potvrzovaly jejich původní stanovisko. Jak zastánci, tak odpůrci trestu smrti uváděli, že to, co četli, je utvrdilo v jejich přesvědčení.

6 Závěr

Východiskem tohoto příspěvku je varování před zneužíváním statistiky k demagogii. Hanousek s Charamzou dokonce přirovnali statistiku k noži – užitečnému nástroji v rukou jedněch, ovšem nebezpečné zbraně v rukou jiných.

V tomto příspěvku jsme na několika příkladech ukázali, jakých chyb se člověk může dopustit při ledabylé či záměrně demagogické interpretaci dat. Zmínili jsme zejména potřebu přemýšlet nejen nad tím, co je v datech obsaženo, ale také nad tím, co v datech z různých důvodů obsaženo není (tzv. zkreslení přežití). Dále jsme upozornili na tzv. konfirmační zkreslení – tendenci vyhledávat data podporující naše stávající názory a naopak přehlížet data, která tyto názory zpochybňují.

Věříme, že povědomí o existenci výše uvedených zkreslení může čtenáři napomoci k tomu, aby se těchto zkreslení vyvaroval, byť jde o úkol velice obtížný.

Reference

- [1] BACON, Francis. *Nové organon*. Vyd. 1. Praha: Svoboda, 1974. 249 s.
- [2] CASSELMAN, Bill *The Legend of Abraham Wald*, 2016 <http://www.ams.org/publicoutreach/feature-column/fc-2016-06>
- [3] HANOUSEK, Jan a Pavel CHARAMZA. *Moderní metody zpracování dat: matematická statistika pro každého*. Praha: Grada, 1992. Educa '99.
- [4] KAŠPAR, Tomáš. *Krutá hra o tvé zdraví: budeš figurkou, nebo hráčem? : jak vyhrát radost ze života, získat kontrolu a vrátit si zdraví*. Hradec Králové: Akademie úspěchu, 2016. ISBN 978-80-906020-3-8.
- [5] kolektiv autorů: *Česko v číslech 2021*. Praha: ČSÚ 2021. ISBN: 978-80-250-3163-6.
- [6] KOŠŤÁKOVÁ, Tereza. *O složitém jednoduše, aneb, Nebojte se statistiky, nekouše*. Praha: Český statistický úřad, 2019. ISBN 978-80-250-2908-4.
- [7] LORD, Charles G., ROSS, Lee, LEPPER, Mark R. *Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence*. Journal of personality and social psychology, 1979, 37.11: 2098.
- [8] MLODINOW, Leonard. *Život je jen náhoda – jak náhoda ovlivňuje naše životy*. Praha: Slovart, 2009. ISBN: 978-80-7391-259-8.
- [9] YOUNG, Aaron, a kol. *FSMB Census of Licensed Physicians in the United States, 2020*. Journal of Medical Regulation, 2021, 107.2: 57–64.

Ondřej Vencálek, Česká statistická společnost, Na padesátém 81,
100 82 Praha 10, Česká republika, e-mail: ondrej.vencalek@upol.cz

Testy v kurzu Nástroje Open source pro zpracování textů

Jiří Rybička

Abstract.

V rámci projektu Innovative Open Source Courses for Computer Science Curriculum je řešen kurz zabývající se zpracováním textů pomocí počítače. V pilotním běhu v rámci aktivit projektu byly aplikovány vybrané kapitoly kurzu a v závěru výuky byl realizován stručný test, jehož cílem bylo zjistit impakt dané výuky i možnosti konstrukce podobných testů rozšířené na kompletní obsah kurzu.

Abstrakt.

It is being solved within the project Innovative Open Source Courses for Computer Science Curriculum computer-based word processing course. In the pilot run as part of the project activities, selected chapters of the course were applied and at the end of the lesson a short test was implemented, the aim of which was to determine the impact of the given lesson and the possibility of constructing similar tests extended to the complete content of the course.

Klíčová slova: Open source, zpracování textů, \TeX , didaktický test, testové otázky

1 Úvod

Výukový kurz v oblasti zpracování textů pomocí programového vybavení osobních počítačů je jedním z hlavních výstupů projektu Innovative Open Source Courses for Computer Science Curriculum a jeho konstrukce (viz Rybička, 2021a) je založena na řešení tří hlavních fází: návrhu sylabu a základní koncepce, zkušebního běhu v intenzivně řešené výuce v rámci tzv. letních škol a na závěr ve vytvoření kompletního výukového materiálu – prezentací použitelných ve výuce a odpovídající učebnice v knižním formátu. Materiály vycházejí z typografických a technologických zásad (Beran, 2012; Kočíčka a Blažek, 2000; Rybička, 2021).

Důležitou možností zpětné vazby jsou ze zmíněných fází možnosti zkušebního běhu ve dvou letních školách. Toho bylo využito pro uplatnění didaktických testů v závěru druhé z letních škol. Článek se zabývá konstrukcí těchto testů, výběrem použitých otázek a zjednodušeným vyhodnocením výsledků získaných na zkušebním vzorku jedné studijní skupiny.

2 Materiál a metody

Didaktické testy obecně mají za cíl získat informaci o tom, jaké znalosti a dovednosti mají v dané chvíli testované osoby. Pokud je obsahem didaktického testu určený okruh učiva (předmět, oblast), pak lze zjistit, jak se respondent v tomto okruhu orientuje. Pokud bychom měli usoudit, jaký vliv měla na danou osobu proběhlá výuka, museli bychom použít dvojici testů – jeden aplikovat na začátku jako test *vstupní*, druhý pak provést po skončení sledované pedagogické aktivity jako *výstupní*. Jejich porovnáním pak lze určit, jaké znalosti a dovednosti byly u testovaných osob získány, přičemž se automaticky předpokládá, že daná pedagogická aktivita (výuka, laboratoř, exkurze, praxe atd.) je jediným zdrojem výukového obsahu.

V případě didaktických testů provedených v rámci kurzu týkajícího se principů a technologie zpracování textů byla využita pouze varianta výstupního testu, kde byl stanoven předpoklad, že tuto oblast testované osoby na začátku výuky nemají ve svém okruhu znalostí a dovedností. Tento zjednodušený předpoklad lze podpořit argumentem, že šlo o osoby, jejichž studijní profil je přibližně znám a nevyskytuje se v něm předmět s podobnou problematikou. Nelze tedy předpokládat, že by se testované osoby již dříve systematicky touto oblastí zabývaly.

2.1 Konstrukce testových otázek

S cílem vytvořit jednoduchou konstrukci byly zvoleny uzavřené otázky typu „1 z N“, kdy byly správné odpovědi doplněny dvěma až třemi distraktory. Pro obsahový základ byly využity jednotlivé kapitoly probírané ve výuce na letní škole. Byly záměrně vystřídány otázky klasifikované jako velmi snadné i obtížnější. Celkem test obsahoval 20 otázek. Jako technologie vlastního testování byly zvoleny Google Forms, a to především pro možnost jednoduché distribuce i sběru výsledků. Test neměl časový limit.

Jednotlivé otázky stručně představíme a uvedeme motivaci k jejich zařazení do testu. Pro usnadnění orientace v tomto textu je vždy první varianta správnou odpovědí, další varianty jsou pak zvolenými distraktory. Ve skutečném testu byly varianty v různém pořadí.

1. What is an important part of the effective document processing principle?
 - Detection of document elements.
 - Knowledge of all menus of the used program.
 - Knowledge of all document file formats.
 - Detection of all font sizes, shapes, and types.

Základní koncepční otázka reprezentující pochopení obecného přístupu ke zpracování dokumentů. Distraktory se soustředují na nepodstatné technologické aspekty.

2. Which system is an Open Source category?

- Libre Office Writer
- Microsoft Word
- Adobe InDesign

Otázka směřující na zaměření projektu – reflexe diskuse o možnostech využití open source systémů a jejich licencovaných alternativ. Zde byla využita pouze trojice odpovědí.

3. What is a basic principle of $\text{T}_{\text{E}}\text{X}$ system?

- Source text is compiled into PDF (or DVI) file format.
- User interface displays always the final form of document.
- Each formatting command is chosen from interactive menu.
- User can edit the final form of document.

Otázka směřující na dva podstatné rysy: systém nemá uživatelské rozhraní a úpravy dokumentu nejsou prováděny v jeho finálním náhledu.

4. What is a structured markup?

- It is a system of tags with information about the meaning of document element.
- It is a system of tags with information about the shape of document element.
- It is a system of tags which create a nested structure.
- It is a system of all tags of document.

Otázka vede k ověření základního pochopení principu oddělení konstrukce dokumentu od jeho vizuální formy. Jde o koncepční přístup k přípravě a modifikaci dokumentu.

5. Choose a system which represents a **structured markup**:

- Microsoft Word with named formatting styles
- Libre Office with all elements in Normal style
- HTML document with tag `` or `<I>`
- $\text{T}_{\text{E}}\text{X}$ like system with `\kern` or `\bfseries` command

V otázce je zjišťováno pochopení realizace strukturního značkování v různých diskutovaných systémech – obecný princip má různé podoby a důležitým výsledkem kurzu je schopnost najít v libovolném systému realizaci strukturních značek.

6. Which TeX command can define a **new macro**:

- `\def`
- `\newmacro`
- `\definecommand`
- `\create`

Ověření detailní znalosti konstrukce strukturních značek v systémech na bázi $\text{T}_{\text{E}}\text{X}$.

7. What is an **active character** in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ source file?

- It is any character with a different meaning than an ordinary character.
- It is any alphabetic character.
- It is any non-alphabetic character.
- It is only braces and punctuation characters.

Otázka směřuje na pochopení základních elementů zdrojového textu v systémech na bázi $\text{T}_{\text{E}}\text{X}$.

8. Which way of marking is optimal for **people's names**?

- small caps
- bold face
- italics
- underline

Jde o otázku z oblasti typografických zásad a pravidel, která jsou důležitá v každém systému a která jsou motivem pro tvorbu strukturních značek.

9. Which font size in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is default?

- `\normalsize`
- `\footnotesize`
- `\small`
- `\large`

V této otázce jde o strukturní značku pro velikost základního písma.

10. Which is the default font category for **printed** documents?

- Roman (serif)
- Monotype
- Sans serif

Otázka směřuje na typografické zásady. Neobsahuje čtyři odpovědi, protože čtvrtá kategorie v nastavení písem neexistuje.

11. How many **mandatory parameters** can have a command in LaTeX?

- 9
- 1
- 5

Otázka směřuje na základní možnosti tvorby vlastních příkazů, což je pro konstrukci strukturních značek podstatná technologie.

12. Which command in L^AT_EX is used for page break?

- `\newpage`
- `\\`
- `\par`

Otázka ověřuje detailní prvek sázecího systému.

13. Which is the **default margin** of standard paragraph?

- 0 pt
- 10 pt
- 1 em

Odpověď na otázku ověřuje znalost implicitního chování systému.

14. Which is the optimal size of **paragraph indent** when typeset in A4 format?

- 2 em
- 1 em
- 5 em

Otázka ověřuje typografické zásady jednoho z parametrů odstavcové sazby.

15. The useful command in TeX that allows you to efficiently **transfer hyphens** at line breaks is called:

- `\discretionary`
- `\hyphenbreak`
- `\penalty`

Otázka byla zařazena jako obtížnější – ověřuje detailní znalost principu, který je velmi užitečný, ale v jiných systémech nedostupný.

16. What is the name of predefined command for the **highest section** in L^AT_EX document class **article**?

- `\section`
- `\chapter`
- `\part`

Otázka ověřuje znalost základního určení předdefinovaných dokumentních tříd.

17. Which is the default **page style** in L^AT_EX?

- `plain`
- `headings`
- `empty`

Detailní otázka na výchozí parametry stránkového designu.

18. Which package is used to set **page parameters**?

- `geometry`
- `page`
- `setpage`

Otázka se zabývá jedním z velmi rozšířených nadstavbových balíčků pro stránkový design.

19. Which type of file is being compiled as a **source**?

- text file (e.g. `*.tex`)
- font file (e.g. `*.otf`)
- DVI file
- PDF file

Koncepční otázka technologického principu práce \TeX ového systému.

20. Which symbol is predefined for a **non-breaking space**?

- \sim (tilde)
- % (percent)
- ! (exclamation mark)

Otázka směřuje na jeden z nejpoužívanějších speciálních znaků.

3 Výsledky

Z 16 vyslaných žádostí o vyplnění testu se vrátilo 13 vyplněných testů. Nebyla stanovena přesná doba, kdy měli respondenti test vyplňovat, bylo pouze nutné test absolvovat v průběhu cca 3 dní. Průměrná doba vyplnění testu byla 22:07 minut.

Tabulka 1: Vyhodnocení jednotlivých otázek testu

Ot. č.	Správně	D1	D2	D3
1	8	0	0	5
2	12	0	1	–
3	12	0	0	1
4	5	4	2	2
5	3	1	3	6
6	11	0	2	0
7	7	0	4	2
8	8	0	5	0
9	13	0	0	0
10	9	4	0	–
11	7	5	1	–
12	9	2	2	–
13	4	5	4	–
14	8	4	1	–
15	4	1	8	–
16	11	2	0	–
17	10	1	2	–
18	9	3	1	–
19	12	0	0	1
20	13	0	0	–

4 Diskuse

Na základě uvedených výsledků můžeme uvést několik postřehů k jednotlivým otázkám i k testu jako celku:

- Pouze dvě otázky byly odpovězeny vždy správně – velmi jednoduchá otázka týkající se názvu stupně základního písma a otázka na nezlomitelnou mezeru.
- Koncepční otázka na dokumentní prvky (1) měla 5 voleb distraktoru týkajícího se orientace na vizuální stránku dokumentu. Je pravděpodobné, že si tito respondenti neuvědomují zásadní rozdíl v přístupu k dokumentu: vizuální parametry jsou pro ně stále primární, tj. nedoceňují princip strukturních informací.
- Podobně neúspěšná byla otázka na strukturní značkování (4). Správných odpovědí nebyla ani polovina, což svědčí o nízkém stupni pochopení této koncepce.
- Ve stejné rovině lze chápat i otázku 5 diskutující systém strukturních značek v různých systémech. Správných odpovědí není ani čtvrtina. Na druhé straně je pravda, že systém MS Word není v daném kurzu brán jako vzor dobré praxe, což mohlo uvažování respondentů výrazně ovlivnit.
- Obtížná otázka 7 dává nepřekvapivé výsledky – bez větší praxe a vzhledu do principu atomů zdrojového textu lze jen těžko přesně vybrat význam aktivního znaku.
- Typografické pravidlo o zdůrazňování jmen (otázka 8) není všeobecně známo a respektováno, je přirozené, že respondenti volili distraktor běžného zdůraznění.
- V otázce 10 šlo o volbu serifového písma, které je podle typografických zásad vhodnější pro tištěné dokumenty. Opět lze vidět ovlivnění nedobrou praxí, kdy je jako základní písmo univerzálně voleno bezserifové (a předdefinované v MS Wordu) písmo.
- V technologické otázce 11 šlo o znalost jisté hranice při tvorbě marker (max. 9 parametrů), což ovšem v praktické části výuky není často diskutovaný případ – lze vidět volby distraktorů častých případů.
- Velmi četné nesprávné volby u otázky 13 jsou pravděpodobně způsobeny nepřesným pochopením otázky – jde o častou záměnu okraje odstavce a okraje stránky.

- U otázky 15 jde o znalost příkazu, který je frekventován pouze v některých jazycích (zejména čeština) a představuje technologickou možnost nedostupnou v jiných systémech. Důkladná znalost příkazů systému není dosažitelná v krátkém intenzivním kurzu, proto není vysoká četnost nesprávných odpovědí na tuto otázku přílišným překvapením.
- Podobně technologická otázka 18 nemá vysokou správnost, což lze z uvedených důvodů opět vysvětlit.

5 Závěr

Celkově lze na základě těchto výsledků vyhodnotit impakt provedeného kurzu jako relativně dostatečný, ale zároveň lze konstatovat, že koncepční prvky, jako je princip strukturního značkování a s tím spojené efektivní zacházení s dokumentem, nejsou v kurzu patrně dostatečně vysvětleny, opakovány a upevňovány. Protože se jedná o zásadní přínos kurzu, je zcela nezbytné věnovat této části více pozornosti ve výuce, uvést více příkladů a propojit je s dalšími technologičtějšími partiiemi.

Reference

- [1] Beran, V. et al., *Aktualizovaný typografický manuál*. Praha: Kafka design, 2012.
- [2] Kočíčka, P., Blažek, F., *Praktická typografie*. Praha: CP Books, 2000.
- [3] Rybička, J., *Zpracování textů v programu Word*. [online] [vid. 20. 11. 2021] Dostupné na <https://akela.mendelu.cz/~{}rybicka/prez/zpract/odbprace/fastword2.docx>
- [4] Rybička, J., Nástroje open source pro zpracování textů. In *Sborník příspěvků z konference Open Source nástroje ve výuce informatiky* Brno, 2021a, s. 91–99.

Jiří Rybička, Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: rybicka@mendelu.cz

Integration with wxMaxima

Rudolf Blaško

Abstract.

The paper is devoted to the preparation of a basic course of mathematical analysis supported by the open source program wxMaxima for symbolic and numerical calculations. This post deals with the basics of integration.

Abstrakt.

Príspevok je venovaný príprave základného kurzu matematickej analýzy podporovanej open source programom wxMaxima na symbolické aj numerické výpočty. Príspevok sa zaoberá základmi integrovania.

Key words: Open Source, calculus, wxMaxima, integration.

The Indefinite Integral

We motivated the introduction of the concept of derivative by the task of determining the instantaneous speed of a mass point, which moves in a straight line. We can reverse the problem and look for the trajectory of the mass point provided we know its instantaneous velocity at the given time.

$I \subset R$ is an open interval, the function $F(x)$, $x \in I$ is called a **primitive function (antiderivative)** to the function $f(x)$, $x \in I$ on the interval I , if the derivative $F'(x)$ exists for all $x \in I$ and $F'(x) = f(x)$ holds for all $x \in I$.

$F(x)$ is primitive to $f(x)$ on the interval I , $c \in R$ (constant).

$\Rightarrow G(x) = F(x) + c$ is primitive to the function $f(x)$ on the interval I .

It follows from the definition of a primitive function, that F is continuous on the interval I . All primitive functions to a given function $f(x)$, $x \in I$ on the interval I differ from each other by a constant and form the set $\{F(x) + c, c \in R\}$, (where F is any of the primitive functions). This set is called **indefinite integral of the function f on the interval I** and is denoted

$$\int f(x) dx = \{F(x) + c, x \in I, c \in R\} = F(x) + c, x \in I, c \in R.$$

$f(x)$, $x \in I$ is continuous on the interval I . \Rightarrow There exists $\int f(x) dx$.


```
(%i1) 'integrate(1/(1+x^2), x)
```

```
(%o1)  $\int \frac{1}{x^2+1} dx$ 
```

The `integrate` command is used to integrate in the wxMaxima.

```
(%i1) integrate(1/(1+x^2), x);
```

```
(%o1) atan x
```

```
(%i2) diff(%, x);
```

```
(%o2)  $\frac{1}{x^2+1}$ 
```

```
(%i1) f(x):=1/(1-x^2); integrate(f(x), x);
```

```
(%o1)  $\frac{1}{1-x^2}$ 
```

```
(%o2)  $\frac{\log(x+1)}{2} - \frac{\log(x-1)}{2}$ 
```

Differentiation and integration are inverse operations on the interval I . F is primitive to the function f on the interval I , $c \in R$, then for all $x \in I$:

$$\left[\int f(x) dx \right]' = [F(x) + c]' = f(x), \quad \int F'(x) dx = \int f(x) dx = F(x) + c.$$

$$\bullet \int \cot x dx = \int \frac{\cos x}{\sin x} dx = \int \frac{[\sin x]'}{\sin x} dx = \ln |\sin x| + c, \\ x \in R - \{k\pi, k \in Z\}, c \in R.$$

$$\bullet \int |x| dx = \begin{cases} \int x dx = \frac{x^2}{2} + c = \frac{x \cdot x}{2} + c = \frac{x|x|}{2} + c, & \text{for } x \geq 0, \\ \int (-x) dx = -\int x dx = -\frac{x^2}{2} + c = \frac{x|x|}{2} + c, & \text{for } x < 0. \end{cases}$$

```
(%i1) integrate(cot(x), x);
```

```
(%o1) log(sin x)
```

```
(%i2) integrate(abs(x), x);
```

```
(%o2)  $\frac{x|x|}{2}$ 
```

Decomposition method.

$$\int [af(x) + bg(x)] dx = a \int f(x) dx + b \int g(x) dx \text{ for } a, b \in R.$$

- $\int \frac{dx}{\sin^2 x \cos^2 x} = \int \frac{\sin^2 x + \cos^2 x}{\sin^2 x \cos^2 x} dx = \int \left[\frac{1}{\cos^2 x} + \frac{1}{\sin^2 x} \right] dx = \tan x - \cot x + c,$
 $x \in R, x \neq \frac{k\pi}{2}, k \in Z, c \in R.$
- $\int \frac{(x-1)^2}{x} dx = \int \frac{x^2 - 2x + 1}{x} dx = \int \left[x - 2 + \frac{1}{x} \right] dx = \frac{x^2}{2} - 2x + \ln |x| + c,$
 $x \in R - \{0\}, c \in R.$

```
(%i1) integrate(1/(sin(x)^2*cos(x)^2),x);
(%o1) tan x - 1/tan x
(%i2) integrate((x-1)^2/x,x);
(%o2) log x + (x^2-4x)/2
```

Method per partes. u, v, u', v' are continuous on the interval I .

$$\int u(x) v'(x) dx = u(x) v(x) - \int u'(x) v(x) dx, \quad x \in I.$$

- $\int x \cos x dx = \left[\begin{array}{l} u = x \\ v' = \cos x \end{array} \middle| \begin{array}{l} u' = 1 \\ v = \sin x \end{array} \right] = x \sin x - \int \sin x dx = x \sin x + \cos x + c,$
 $x \in R, c \in R.$

```
(%i1) u:x; v:integrate(cos(x),x);
(u) x
(v) sin x
(%i3) u*v-integrate(v,x);
(%o3) x sin x + cos x
(%i4) integrate(x*cos(x),x);
(%o4) x sin x + cos x
```

- $\int \ln x dx = \left[\begin{array}{l} u = \ln x \\ v' = 1 \end{array} \middle| \begin{array}{l} u' = \frac{1}{x} \\ v = x \end{array} \right] = x \ln x - \int dx = x \ln x - x + c,$
 $x \in (0; \infty), c \in R.$

We can use the per partes method several times in a row, but we must be careful to stick to it they did not return to the original integral by reuse. The per partes method is used quite often. It is suitable for integrating functions

$$P(x) e^{ax}, \quad P(x) \cos ax, \quad P(x) \sin ax, \quad P(x) \ln Q(x), \quad P(x) \arctan Q(x),$$

where $P(x), Q(x)$ are real polynomials, $a \in R, a \neq 0$.

Substitution method. F is primitive to the f on the interval I ,
 $x = \varphi(t)$ has a derivative on the interval J , $\varphi(J) \subset I$.

$$\int f(\varphi(t)) \cdot \varphi'(t) dt = \int f(x) dx = F(x) + c = F(\varphi(t)) + c, t \in J, c \in R.$$

Sets I, J are intervals, $x = \varphi(t): J \rightarrow I$ has a derivative $\varphi'(t) \neq 0$ on J ,
 $F(t)$ is primitive to $f(\varphi(t)) \cdot \varphi'(t)$ on J .

$$\int f(x) dx = \int f(\varphi(t)) \cdot \varphi'(t) dt = F(t) + c = F(\varphi^{-1}(x)) + c, x \in I, c \in R.$$

In the first case we do not have to use inverse substitution, but in the second case we have to use the inverse substitution $t = \varphi^{-1}(x)$.

- $\int \sin^3 t \cos t dt = \left[\text{Subst. } x = \sin t \left| \begin{array}{l} t \in R \\ dx = \cos t dt \\ x \in (-1; 1) \end{array} \right. \right] = \int x^3 dx = \frac{x^{3+1}}{3+1} + c = \frac{\sin^4 t}{4} + c, t \in R, c \in R.$
- $\int \frac{f'(x) dx}{f(x)} = \left[\frac{t = f(x)}{dt = f'(x) dx} \right] = \int \frac{dt}{t} = \ln |t| + c = \ln |f(x)| + c, x \in D(f), c \in R.$
- $\int e^{5x} dx = \left[\begin{array}{l} t = 5x, x = \frac{t}{5} \\ dx = \frac{dt}{5} \\ t \in R \end{array} \right] = \int \frac{e^t dt}{5} = \frac{e^t}{5} + c = \frac{e^{5x}}{5} + c, x \in R, c \in R.$

To perform a t -substitution, we:

- Decide on a substitution and use `diff` to produce the differential dt (called `del(t)` in wxMaxima), then express dx in terms of dt using `solve`.
- Extract the resulting equation using `%[1]` and replace `del(x)` with its expression in terms of `del(t)` in the integrand.
- Use `subst` to transform the entire integrand in terms of t , then perform the integral, remembering that `integrate` expects only the coefficient of `del(t)`.
- Substitute the definition of t in terms of x into the resulting antiderivative.

```
(%i1) INTEGRAND: (%e^(5*x))*diff(x);
(%o1) e5x del(x)
(%i2) solve(diff(t)=diff(5*x), del(x));
(%o2) [del(x) = del(t)/5]
(%i3) %[1];
(%o3) del(x) = del(t)/5
(%i5) subst(rhs(%), del(x), INTEGRAND)$ subst(t, 5*x, %);
(%o5) et del(t)/5
(%i6) integrate(coeff(%, del(t)), t);
(%o6) et/5
(%i7) subst(5*x, u, %);
(%o7) e5x/5
```

```
(%i8) integrate(%e^(5*x), x);
(%o8) e^{5x}/5
```

$$\begin{aligned} \bullet \int \frac{dx}{\sqrt{1-x^2}} &= \left[\text{Subst. } x = \sin t \mid x \in (-1; 1) \mid dx = \cos t dt, (\sin t)' = \cos t > 0 \text{ for } t \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right) \right. \\ &\quad \left. t = \arcsin x \mid t \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right) \mid \sqrt{1-x^2} = \sqrt{1-\sin^2 t} = \sqrt{\cos^2 t} = |\cos t| = \cos t \right] \\ &= \int \frac{\cos t dt}{\cos t} = \int dt = t + c = \arcsin x + c, x \in (-1; 1), c \in \mathbb{R}. \\ \bullet \int \frac{dx}{\sqrt{1-x^2}} &= \left[\text{Subst. } x = \cos t \mid x \in (-1; 1) \mid dx = -\sin t dt, -(\cos t)' = \sin t > 0 \text{ for } t \in (0; \pi) \right. \\ &\quad \left. t = \arccos x \mid t \in (0; \pi) \mid \sqrt{1-x^2} = \sqrt{1-\cos^2 t} = \sqrt{\sin^2 t} = |\sin t| = \sin t \right] \\ &= \int \frac{-\sin t dt}{\sin t} = -\int dt = -t + c = -\arccos x + c, x \in (-1; 1), c \in \mathbb{R}. \end{aligned}$$

Both solutions are correct because it holds $\arcsin x + \arccos x = \frac{\pi}{2}$, i.e. $\arcsin x = -\arccos x + \frac{\pi}{2}$ for all $x \in \langle -1; 1 \rangle$.

When integrating, different methods are often combined, and sometimes they have to be used several times in a row. If we use different integration procedures, we can arrive at other primitive functions.

The Definite Integral

In this section, we will deal with the definite integral of the function, which, in contrast to the indefinite integral is not a function, but a specific value.

Example. $y = f(x)$, $x \in \langle a; b \rangle$ is a positive continuous function, $a, b \in \mathbb{R}$. Determine the area of the set $P = \{[x; y] \in \mathbb{R}^2, x \in \langle a; b \rangle, 0 \leq y \leq f(x)\}$.

• Let us divide $\langle a; b \rangle$ using points $a = x_0 < x_1 < x_2 < \dots < x_n = b$, $n \in \mathbb{N}$ on n subintervals $I_1 = \langle x_0; x_1 \rangle$, $I_2 = \langle x_1; x_2 \rangle$, \dots , $I_n = \langle x_{n-1}; x_n \rangle$ with the same length $\Delta x = x_1 - x_0 = x_2 - x_1 = \dots = x_n - x_{n-1} = \frac{b-a}{n}$.

• $m_i = \min \{f(x), x \in I_i\}$, $M_i = \max \{f(x), x \in I_i\}$ for $i = 1, 2, \dots, n$.

For the area P then $\sum_{i=1}^n m_i \cdot \Delta x = L_P(n) \leq P \leq U_P(n) = \sum_{i=1}^n M_i \cdot \Delta x$.

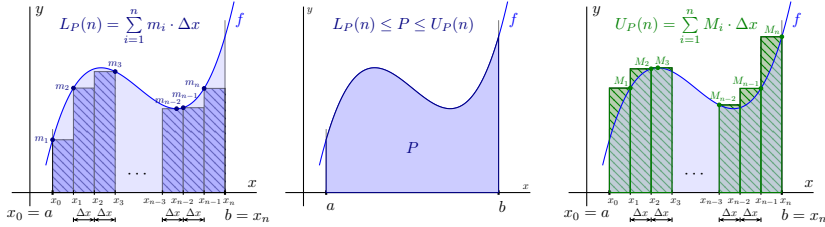
• If we decrease Δx (increase n), the L_P, U_P will improve (do not get worse).

• For $\Delta x = \frac{b-a}{n} \rightarrow 0$, i.e. $n \rightarrow \infty$ will hold $L_P(n) \rightarrow P$, $U_P(n) \rightarrow P$.

Dividing the interval $\langle a; b \rangle$ is called each finite set of points

$$D = \{x_0, x_1, x_2, \dots, x_n\} = \{x_i\}_{i=0}^n, n \in \mathbb{N},$$

for which $a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$. Interval lengths $d_i = \langle x_{i-1}; x_i \rangle$, $i = 1, 2, \dots, n$ we denote $\Delta x_i = x_i - x_{i-1}$. We call the length of the longest of these intervals **dividing norm** D and denote $\mu(D)$, i.e. $\mu(D) = \max \{\Delta x_i, i = 1, 2, \dots, n\}$. For the lengths d_1, d_2, \dots, d_n it holds $\Delta x_1 + \Delta x_2 + \dots + \Delta x_n = x_n - x_0 = b - a$. The set of all dividings of the interval $\langle a; b \rangle$ we denote $\mathfrak{D}_{\langle a; b \rangle} = \{D, D \text{ is the division } \langle a; b \rangle\}$.



We denote $m_i = \inf \{f(x), x \in I_i\}$, $m = \inf \{f(x), x \in \langle a; b \rangle\}$, $M_i = \sup \{f(x), x \in I_i\}$, $M = \sup \{f(x), x \in \langle a; b \rangle\}$ for $i = 1, 2, \dots, n$ for bounded function $y = f(x)$, $x \in \langle a; b \rangle$. **Lower** $S_L(f, D)$ and **upper Riemannian integral sum** $S_U(f, D)$ of function f under dividing D are called numbers $S_L(f, D) = \sum_{i=1}^n m_i \cdot \Delta x_i$, $S_U(f, D) = \sum_{i=1}^n M_i \cdot \Delta x_i$. And numbers

$$\int_a^b f(x) dx = \sup_{D \in \mathfrak{D}_{\langle a; b \rangle}} S_L(f, D), \quad \int_a^b f(x) dx = \inf_{D \in \mathfrak{D}_{\langle a; b \rangle}} S_U(f, D)$$

we call the **lower** and **upper Riemannian (definite) integral** of the function f on the $\langle a; b \rangle$ (from a to b). These numbers always exist and apply

$$m(b - a) \leq \int_a^b f(x) dx \leq \int_a^b f(x) dx \leq M(b - a).$$

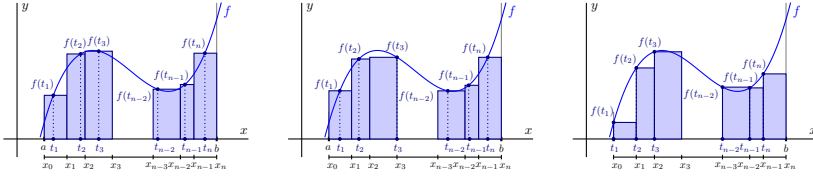
If the equality holds, then this common value we call the **Riemannian (definite) integral** of the function f on the $\langle a; b \rangle$. We call the function f **Riemannian integrable** and we denote $f \in R_{\langle a; b \rangle}$,

$$\int_a^b f(x) dx = \int_a^b f(x) dx = \int_a^b f(x) dx.$$

Riemannian integral sum of the f under dividing D and choice of points $T = \{t_1, t_2, \dots, t_n\} = \{t_i \in I_i\}_{i=1}^n$ we call $S_T(f, D) = \sum_{i=1}^n f(t_i) \cdot \Delta x_i$.

The function f has infinitely many integral sums for a given dividing D . If $y = f(x)$, $x \in \langle a; b \rangle$ is continuous, then f takes its extrema on each interval $\langle x_{i-1}; x_i \rangle$, $i = 1, 2, \dots, n$ and $S_L(f, D)$ and $S_U(f, D)$ are also Riemannian integral sums for some particular choices of points T .

In investigating Riemannian integrability on the $\langle a; b \rangle$ we do not need to deal with all by divisions $D \in \mathfrak{D}_{\langle a; b \rangle}$. It is sufficient to restrict ourselves to **normal sequences** of dividing $\{D_k\}_{k=1}^\infty \subset \mathfrak{D}_{\langle a; b \rangle}$, i.e. $\lim_{k \rightarrow \infty} \mu(D_k) = 0$. Then it holds $\lim_{k \rightarrow \infty} S_T(f, D_k) = \int_a^b f(x) dx$ for every choice of points T .



Example. Calculate the integral $\int_0^1 \frac{x}{2} dx$.

- The function $f(x) = \frac{x}{2}$, $x \in \langle 0; 1 \rangle$ is increasing, continuous, $f \in R_{\langle 0; 1 \rangle}$.
- A normal sequence of dividings $\{D_k\}_{k=1}^\infty \subset \mathcal{D}_{\langle 0; 1 \rangle}$, while $D_k = \left\{ \frac{i}{k} \right\}_{i=0}^k$.
- $\Delta x_i = \frac{1}{k}$, $m_i = f(x_{i-1}) = \frac{i-1}{2k}$, $M_i = f(x_i) = \frac{i}{2k}$ for $i = 1, 2, \dots, k$.

$$S_L(f, D_k) = \sum_{i=1}^k m_i \Delta x_i = \sum_{i=1}^k \frac{i-1}{2k} \cdot \frac{1}{k} = \frac{0+1+\dots+(k-1)}{2k^2} = \frac{(0+k-1)k}{2k^2} = \frac{k-1}{4k}.$$

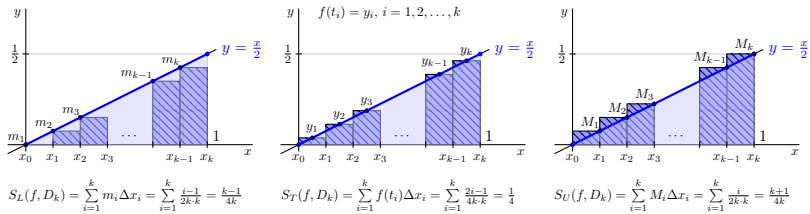
$$S_U(f, D_k) = \sum_{i=1}^k M_i \Delta x_i = \sum_{i=1}^k \frac{i}{2k} \cdot \frac{1}{k} = \frac{1+2+\dots+k}{2k^2} = \frac{(1+k)k}{2k^2} = \frac{k+1}{4k}.$$

$$\Rightarrow \int_0^1 \frac{x}{2} dx = \lim_{k \rightarrow \infty} S_L(f, D_k) = \lim_{k \rightarrow \infty} S_U(f, D_k) = \lim_{k \rightarrow \infty} \frac{k \mp 1}{4k} = \frac{1}{4}.$$

- Let's choose $T = \{t_i\}_{i=1}^k$ as points centers of $\langle x_{i-1}; x_i \rangle$, $i = 1, 2, \dots, k$, i.e. $t_i = \frac{1}{2} \left(\frac{i-1}{k} + \frac{i}{k} \right) = \frac{2i-1}{2k}$, then $f(t_i) = \frac{2i-1}{4k}$ and holds

$$S_T(f, D_k) = \sum_{i=1}^k f(t_i) \Delta x_i = \sum_{i=1}^k \frac{2i-1}{4k} \cdot \frac{1}{k} = \frac{1+3+\dots+(2k-1)}{4k^2} = \frac{(1+2k-1)k}{4k^2} = \frac{1}{4}.$$

$$\Rightarrow \int_0^1 \frac{x}{2} dx = \lim_{k \rightarrow \infty} S_T(f, D_k) = \lim_{k \rightarrow \infty} \frac{1}{4} = \frac{1}{4}.$$



$$S_L(f, D_k) = \sum_{i=1}^k m_i \Delta x_i = \sum_{i=1}^k \frac{i-1}{2k} \cdot \frac{1}{k} = \frac{k-1}{4k}$$

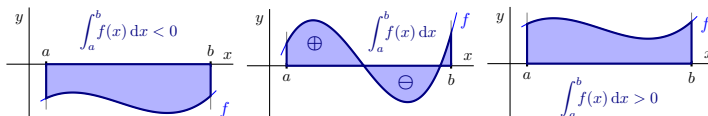
$$S_U(f, D_k) = \sum_{i=1}^k M_i \Delta x_i = \sum_{i=1}^k \frac{2i-1}{4k} \cdot \frac{1}{k} = \frac{1}{4}$$

$$S_T(f, D_k) = \sum_{i=1}^k M_i \Delta x_i = \sum_{i=1}^k \frac{i}{2k} \cdot \frac{1}{k} = \frac{k+1}{4k}$$

Geometrically, it represents the Riemannian definite integral on the $\langle a; b \rangle$ the area of the curvilinear trapezoid determined by the function f and the interval $\langle a; b \rangle$. Below the x axis (i.e., if f is negative), this area is negative.

$$f \in R_{\langle a; b \rangle}, f(x) \geq 0 \text{ for all } x \in \langle a; b \rangle. \Rightarrow \int_a^b f(x) dx \geq 0.$$

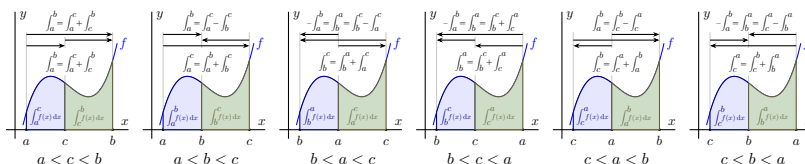
$$f, g \in R_{\langle a; b \rangle}, g(x) \geq f(x) \text{ for all } x \in \langle a; b \rangle. \Rightarrow \int_a^b g(x) dx \geq \int_a^b f(x) dx.$$



Additivity of the integral. $f \in R_I, I \subset \mathbb{R}$ is a bounded interval, Points $a, b, c \in I$ are arbitrary. $\Rightarrow \int_a^b f(x) dx = \int_a^c f(x) dx + \int_c^b f(x) dx$.

We can clearly illustrate the additivity on vectors on the real axis. If we imagine $\int_a^b, \int_b^a = -\int_a^b$ as $\vec{ab}, \vec{ba} = -\vec{ab}$, then e.g. for $a < b < c$ it holds

$$\int_a^b = \int_a^c + \int_c^b = \int_a^c - \int_b^c, \text{ i.e. } \vec{ab} = \vec{ac} + \vec{cb} = \vec{ac} - \vec{bc}.$$



Calculation of the Riemann integral (Newton-Leibniz formula).
 $\int_a^b f(x) dx = F(b) - F(a) = [F(x)]_a^b$,
 where F is any primitive function to the function f on $\langle a; b \rangle$.

- $\int_{-1}^0 \frac{x}{2} dx = \left[\frac{x^2}{2 \cdot 2} \right]_{-1}^0 = \left[\frac{x^2}{4} \right]_{-1}^0 = \frac{0^2}{4} - \frac{(-1)^2}{4} = \frac{1}{4}$.
- $\int_{-1}^1 x^2 dx = \left[\frac{x^3}{3} \right]_{-1}^1 = \frac{1^2}{3} - \frac{(-1)^3}{3} = \frac{2}{3}$.

```
(%i2) f(x):=x^2$ F:integrate(f(x),x);
(F) x^3/3
(%i3) integrate(f(x),x,-1,1);
(%o3) 2/3
(%i4) subst(1,x,F)-subst(-1,x,F);
(%o4) 2/3
(%i5) float(subst(1,x,F)-subst(-1,x,F));
(%o5) 0.6666666666666666
(%i6) float(integrate(f(x),x,-1,1));
(%o6) 0.6666666666666666
```

$$\bullet \int_1^2 \cos x \sin x \, dx = \left[\begin{array}{l} t = \cos x \\ \frac{dt}{dx} = -\sin x \end{array} \right] = - \int_{\cos 1}^{\cos 2} t \, dt = - \left[\frac{t^2}{2} \right]_{\cos 1}^{\cos 2} = -\frac{\cos^2 2}{2} + \frac{\cos^2 1}{2}.$$

```
(%i2) f(x):=cos(x)*sin(x)$ F:integrate(f(x),x);
(F) -cos(x)^2/2
(%i3) integrate(f(x),x,1,2);
(%o3) cos(1)^2/2 - cos(2)^2/2
(%i4) subst(2,x,F)-subst(1,x,F);
(%o4) cos(1)^2/2 - cos(2)^2/2
(%i5) float(integrate(f(x),x,1,2));
(%o5) 0.05937419607911741
(%i6) float(subst(2,x,F)-subst(1,x,F));
(%o6) 0.05937419607911741
```

$$\bullet \int_{-1}^1 \frac{dx}{x} = \left[\ln |x| \right]_{-1}^1 = \ln 1 - \ln 1 = 0 \quad (?!?).$$

The function $\frac{1}{x}$ (it is not defined at point 0) and the function $\ln x$ is not bounded on $\langle -1; 1 \rangle$ and in this sense we cannot calculate the integral.

```
(%i2) f(x):=1/x$ F:integrate(f(x),x);
(F) -log(x)
(%i3) integrate(f(x),x,-1,1);
Principal Value
(%o3) 0
(%i4) subst(1,x,F)-subst(-1,x,F);
(%o4) -log(-1)
```

Definite integrals are generally calculated using indefinite integrals. We can modify the per partes and substitution methods and calculate using them directly. After substitution, we do not need to return to the original variables.

Method per partes.

$$u, v, u', v' \in R_{(a;b)}.$$

$$\int_a^b u(x) v'(x) \, dx = \left[u(x) v(x) \right]_a^b - \int_a^b u'(x) v(x) \, dx.$$

$$\begin{aligned} \bullet \int_0^{2\pi} x^2 \sin x \, dx &= \left[\begin{array}{l} u = x^2 \quad | \quad u' = 2x \\ v' = \sin x \quad | \quad v = -\cos x \end{array} \right] = \left[-x^2 \cos x \right]_0^{2\pi} + \int_0^{2\pi} 2x \cos x \, dx \\ &= \left[\begin{array}{l} u = 2x \quad | \quad u' = 2 \\ v' = \cos x \quad | \quad v = \sin x \end{array} \right] = \left[-4\pi^2 \cdot 1 + 0^2 \cdot 1 \right] + \left[2x \sin x \right]_0^{2\pi} - \int_0^{2\pi} 2 \sin x \, dx \\ &= -4\pi^2 + \left[4\pi \cdot 0 - 2 \cdot 0 \cdot 0 \right] - \left[-2 \cos x \right]_0^{2\pi} = -4\pi^2 - \left[-2 \cdot 1 + 2 \cdot 1 \right] = -4\pi^2. \end{aligned}$$

Substitution method.

$$\varphi(J) \subset I, \varphi(\alpha) = a, \varphi(\beta) = b.$$

$y = f(x)$ is continuous on interval I with boundaries a, b .

$x = \varphi(t), \varphi'$ are continuous on interval J with boundaries α, β .

$$\int_a^b f(x) dx = \int_\alpha^\beta f[\varphi(t)]\varphi'(t) dt.$$

We can use the method in both directions.

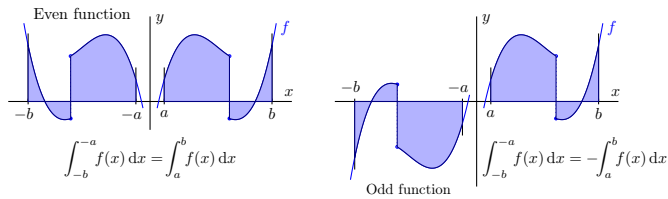
$$\begin{aligned} \bullet \int_{-1}^1 \sqrt{1-x^2} dx &= \left[\begin{array}{l} \text{Subst. } x = \sin t \mid x \in \langle -1; 1 \rangle \mid \pm 1 = \sin(\pm \frac{\pi}{2}) \\ dx = \cos t dt \mid t \in \langle -\frac{\pi}{2}; \frac{\pi}{2} \rangle \mid \cos t \geq 0 \\ \sqrt{1-x^2} = \sqrt{1-\sin^2 t} = \sqrt{\cos^2 t} = |\cos t| = \cos t \end{array} \right] = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^2 t dt \\ &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{1+\cos 2t}{2} dt = \frac{1}{2} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} [1 + \cos 2t] dt = \frac{1}{2} \left[t + \frac{\sin 2t}{2} \right]_{-\frac{\pi}{2}}^{\frac{\pi}{2}} = \frac{\pi}{2}. \end{aligned}$$

$f \in R_{(a;b)}$ is even or odd, where $a < b$. $\Rightarrow f(-x) \in R_{(-b;-a)}$ and holds

$$\int_a^b f(x) dx = \left[\begin{array}{l} \text{Subst. } t = -x \\ dt = -dx \end{array} \right] = - \int_{-a}^{-b} f(-t) dt = \int_{-b}^{-a} f(-t) dt = \int_{-b}^{-a} f(-x) dx.$$

$\bullet f$ is even, then $\int_a^b f(x) dx = \int_{-b}^{-a} f(x) dx$.

$\bullet f$ is odd, then $\int_a^b f(x) dx = \int_{-b}^{-a} [-f(x)] dx = - \int_{-b}^{-a} f(x) dx$.



$a > 0, f \in R_{(-a;a)}$.

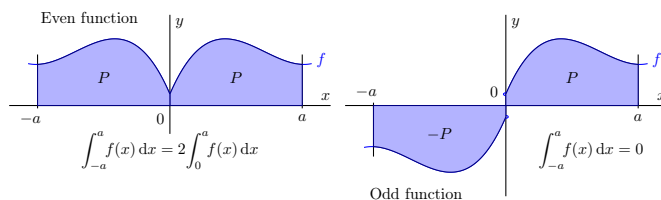
$\bullet f$ is even. $\Rightarrow \int_{-a}^a f(x) dx = \int_{-a}^0 f(x) dx + \int_0^a f(x) dx = \int_0^{-(a)} f(x) dx + \int_0^a f(x) dx = 2 \int_0^a f(x) dx$.

$\bullet f$ is odd. $\Rightarrow \int_{-a}^a f(x) dx = \int_{-a}^0 f(x) dx + \int_0^a f(x) dx = - \int_0^{-(a)} f(x) dx + \int_0^a f(x) dx = 0$.

Example.

$\bullet \int_{-\pi}^{\pi} \sin |x| dx = \left[\begin{array}{l} \sin |x| \text{ is continuous} \\ \text{and even on } \langle -\pi; \pi \rangle \end{array} \right] = 2 \int_0^{\pi} \sin |x| dx$

$$= 2 \int_0^{\pi} \sin x dx = 2 \left[-\cos x \right]_0^{\pi} = -2 \cdot (-1) + 2 \cdot 1 = 4.$$



- $\int_{-1}^1 \frac{x^3 \sqrt[3]{x^2 + \sin^2 x}}{x^4 + 1} dx = \left[\begin{array}{l} \text{integrand is continuous} \\ \text{and odd function on } \langle -1; 1 \rangle \end{array} \right] = 0.$

$f \in R_{\langle a; b \rangle}$ is periodic with period $p > 0$.

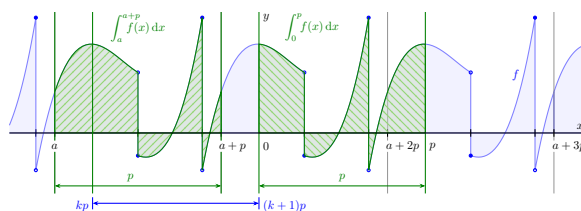
$$\int_a^b f(x) dx = \left[\begin{array}{l} \text{Subst. } x = t - kp \\ t = x + kp, dx = dt \end{array} \right] = \int_{a+kp}^{b+kp} f(t - kp) dt$$

$$= \int_{a+kp}^{b+kp} f(t) dt = \int_{a+kp}^{b+kp} f(x) dx.$$

The value of the Riemann integral of the periodic function f with period p is the same on each interval $\langle a; a + p \rangle$, $a \in R$.

$y = f(x)$ is a periodic function with period $p > 0$, $a \in R$.

$$\int_0^p f(x) dx = \int_a^{a+p} f(x) dx \text{ (if they exist).}$$



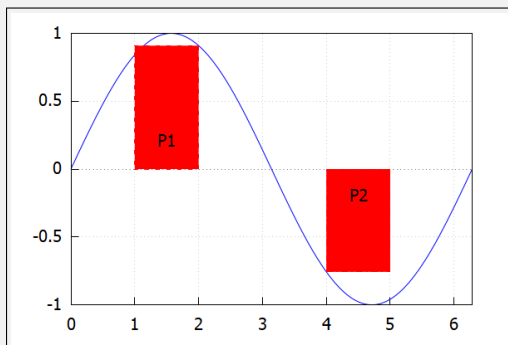
- $\int_a^{a+2\pi} \sin^2(nx) dx = \int_0^{2\pi} \sin^2(nx) dx = \int_0^{2\pi} \frac{1 - \cos(2nx)}{2} dx = \left[\frac{x}{2} - \frac{\sin(2nx)}{2 \cdot 2n} \right]_0^{2\pi}$
 $= \left[\frac{2\pi}{2} - \frac{\sin(2n \cdot 2\pi)}{4n} - \frac{0}{2} + \frac{\sin 0}{4n} \right] = [\pi - 0 - 0 + 0] = \pi$ for $a \in R$, $n \in N$.
- $\int_a^{a+2\pi} \sin(nx) \cos(nx) dx = \int_0^{2\pi} \sin(nx) \cos(nx) dx$
 $= \int_{-\pi}^{\pi} \sin(nx) \cos(nx) dx = 0$ for $a \in R$, $m, n \in N$, $m \neq n$.

```
(%i1) f(x,n):=sin(n*x)^2;
(%o1) f(x,n):sin(n*x)^2
(%i2) integrate(f(x,n),x,0,2*%pi);
(%o2) -sin(4*pi*n)-4*pi*n
      4*n
(%i3) integrate(f(x,n),x,a+0,a+2*%pi);
(%o3) sin(2*a*n)-2*a*n - sin((2*a+4*pi)*n)+(-2*a-4*pi)*n
      4*n              4*n
(%i4) ratsimp(%o3);
(%o4) -sin((2*a+4*pi)*n)-sin(2*a*n)-4*pi*n
      4*n
(%i5) integrate(f(x,4),x,0,2*%pi);
(%o5) pi
(%i6) integrate(f(x,4),x,a+0,a+2*%pi);
(%o6) sin(8*a)-8*a - sin(8*a)-8*a-16*pi
      16              16
(%i7) ratsimp(%);
(%o8) pi
```

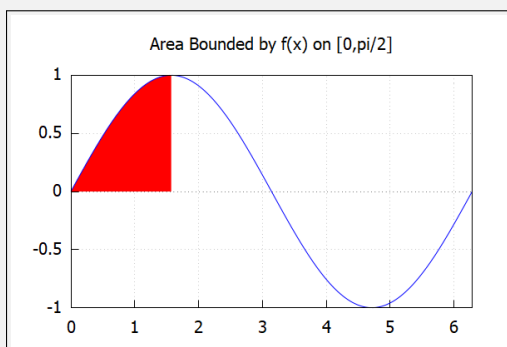
Recall that $\int_a^b f(x) dx$

gives us the area bounded between $f(x)$ and the x -axis.

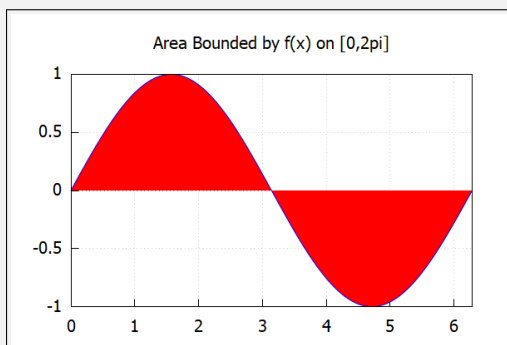
```
(%i1) f(x):=sin(x)$
wxdraw2d(grid=true,xaxis=true,yaxis=true,
color=blue,explicit(f(x),x,0,2*%pi),border=false,
rectangle([1,0],[2,f(2)]),color=black,
label(["P1",1.5,0.2]),
rectangle([4,f(4)],[5,0]),color=black,
label(["P2",4.5,-0.2]));
```



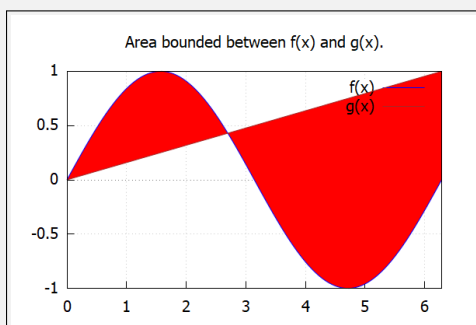
```
(%i1) f(x):=sin(x)$
      wxdraw2d(grid=true,xrange=[0,2*%pi],yrange=[-1,1],
      xaxis=true,yaxis=true,
      title="Area Bounded by f(x) on [0,pi/2]",
      fill_color=red,filled_func=true,filled_func=f(x),
      explicit(0,x,0,%pi/2),filled_func=false,
      color=blue,explicit(f(x),x,0,2*%pi));
```



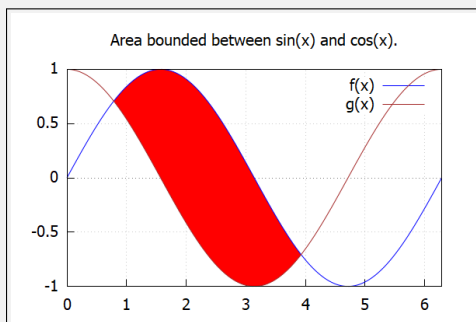
```
(%i1) f(x):=sin(x)$
      wxdraw2d(grid=true,xrange=[0,2*%pi],yrange=[-1,1],
      xaxis=true,yaxis=true,
      title="Area Bounded by f(x) on [0,2pi]",
      fill_color=red,filled_func=true,filled_func=f(x),
      explicit(0,x,0,2*%pi),filled_func=false,
      color=blue,explicit(f(x),x,0,2*%pi));
```



```
(%i1) f(x):=sin(x)$ g(x):=x/(2*%pi)$
      wxdraw2d(grid=true,xaxis=true,yaxis=true,
      xrange=[0,2*%pi],yrange=[-1,1],
      title="Area bounded between f(x) and g(x).",
      fill_color=red,filled_func=true,filled_func=f(x),
      explicit(g(x),x,0,2*%pi),filled_func=false,color=blue,
      key="f(x)",explicit(f(x),x,0,2*%pi),color=brown,
      key="g(x)",explicit(g(x),x,0,2*%pi));
```



```
(%i1) f(x):=sin(x)$ g(x):=cos(x)$
      wxdraw2d(grid=true,xaxis=true,yaxis=true,
      xrange=[0,2*%pi],yrange=[-1,1],
      title="Area bounded between sin(x) and cos(x).",
      fill_color=red,filled_func=true,filled_func=f(x),
      explicit(g(x),x,%pi/4,5*%pi/4),filled_func=false,
      color=blue,key="f(x)",explicit(f(x),x,0,2*%pi),
      color=brown,key="g(x)",explicit(g(x),x,0,2*%pi));
```



Literatúra

- [1] Blaško R., *Matematická analýza I*, Žilina, EDIS 2009, ISBN/ISSN: 978-80-554-0119-5.
- [2] Blaško R., *Neurčitý a určitý integrál reálnej funkcie*, skriptum, <https://frcatel.fri.utc.sk/~beerb/ma1/sa2.pdf>.
- [3] Buša J., *Maxima Open source systém počítačovej algebry*, 2006, online, <http://sccg.sk/~batorova/UPG/priruckaSK.pdf>.
- [4] Bittinger M. L., Ellenbogen D. J., Surgent S. A., *Calculus and its Applications*, Addison-Wesley, ISBN-10: 0-321-69433-3.
- [5] Crowell B., *Calculus, Light and Matter*, www.lightandmatter.com, March 2010.
- [6] Hannan Z., *wxMaxima for Calculus I and II*, Solano Community College, <https://wxmaximafor.wordpress.com/>.
- [7] Marsden J., Weinstein A., *Calculus I — III*, Springer.

Rudolf Blaško, Department of Mathematical methods and Operations Research, Faculty of Management Sciences and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic, e-mail: beerb@frcatel.fri.uniza.sk

GNU Radio and SDR platforms

Mirosław Łazoryszczak

Abstract.

The paper briefly introduces the features of an open-source environment for wireless transmission tasks. Physical data transmission requires target hardware platforms, but the idea of software radio is to move as many operations as possible to software. The leading representative of open-source is GNU Radio. This environment works with professional and budget hardware, particularly for teaching. Available software radio platforms such as RTL-SDR and HackRF One were used to illustrate the software's capabilities.

Abstrakt.

Artykuł krótko przedstawia cechy środowiska open source dla zadań transmisji bezprzewodowej. Fizyczna transmisja danych wymaga docelowych platform sprzętowych, ale idea radia programowego polega na przeniesieniu jak największej liczby operacji do oprogramowania. Wiodącym przedstawicielem open source jest GNU Radio. To środowisko współpracuje z profesjonalnym i niedrogim sprzętem, szczególnie do celów dydaktycznych. Dostępne platformy radiowe oprogramowania, takie jak RTL-SDR i HackRF One, zostały wykorzystane do zilustrowania możliwości oprogramowania.

Key words: GNU Radio, Radio Frequencies (RF), Software Defined Radio (SDR), wireless communication

1 Introduction

Software Defined Radio (SDR) technology originated in the 1970s, although the term SDR itself only appeared in the 1990s [9]. The idea behind SDR is to increase the role of software in the radio circuit, which was initially realized entirely in analog form (e.g. a superheterodyne receiver – Figure 1).

Programmable radio assumes digital conversion at the earliest possible stage of processing and realization of further stages not only in digital form but moreover utilizing software (Figure 2). An essential characteristic of SDR is reprogrammable circuit technology (CPLD, FPGA).

Software Defined Radio, like all hardware-software systems, has disadvantages and significant advantages. The most crucial disadvantage is that the analog-to-digital converter (ADC) limits the highest frequency used. The advantages, however, are much more:

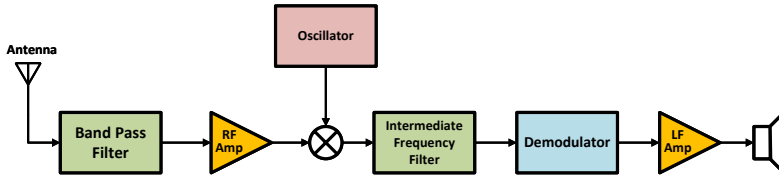


Figure 1: Block diagram of conventional analog receiver

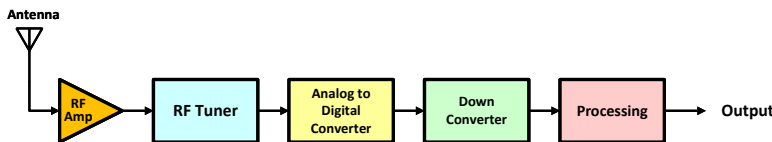


Figure 2: Basic diagram of SDR receiver

- Implementation of all functions (except ADC) using programmable digital circuits, allowing flexible (programmable) changes and updates.
- Rapid prototyping of new solutions due to the typical platform architecture,
- Software component reusing, reducing development costs,
- Remote reprogramming (including over-the-air) to reduce the time and maintenance costs,
- Reconfigurable communications systems (adapting to new standards).

The idea of SDR enables the further development of radio technologies at higher levels of abstraction, embodied in the concept of cognitive radio, which involves recognizing the parameters of one's radio environment and automatically adapting one's characteristics to them. In particular, this involves operating frequencies to achieve the best quality of connection (Quality of Service, QoS) [6, 1].

2 GNU Radio and its capabilities

Software Defined Radio platforms can be supported by a various applications, the vast majority of which are commercial, and dedicated to specific

hardware platforms. However, full use of software radio requires equally flexible software. It should be mentioned that commercial environments also allow the use of SDR in the above-mentioned manner [7]. In the open-source domain, the most popular environment is GNU Radio.

The most important features of GNU Radio environment are:

- Free and open-source software development toolkit,
- Signal processing blocks to implement software radios,
- Support for low-cost external RF hardware to create software-defined radios,
- Support simulation-like environment
- Application in research, industry, academia, government, and hobbyist environments to support both wireless communications research and real-world radio systems.

GNU Radio is available for various operating systems [4]. It is worth noting that numerous tutorials of varying complexity are available to introduce the user to the design environment [5].

The main project file is a Flowchart. It contains a detailed block diagram of the system in graphical form. The flowchart is composed of widgets and connections between them. These blocks include entire subsystems of the radio circuits such as filters, modulators, and demodulators but also basic blocks that perform calculations or convert data formats. GNU Radio can operate in simulation mode (e.g., generating data subject to visualization or downloading previously generated data from external files). It can also operate physical devices in real mode, i.e., as a radio receiver or transmitter. When writing this paper, the latest version of GNU Radio is 3.10 and based on QT widgets. Note that some of the widely available material on GNU Radio and SDR uses flowchart, which is based on WX GUI blocks and has not been officially supported since version 3.8.

In addition to ready-made blocks used in flowcharts, GNU Radio also allows building its graphical user interface (GUI). Such an interface is extremely useful for real-time testing systems.

3 Selected SDR hardware platforms

GNU Radio supports numerous programmable radio hardware platforms. Among them, you can find professional, consumer, and budget solutions.

The former group can include the USRP2 from Ettus Research, offering a base unit and a lot of extensions ([2]). A top-rated solution according to [8] while remaining affordable, is the Hack RF One device. The important thing is that it can act as an RF receiver or transmitter. At the opposite pole are several devices realized in the form of small USB sticks.



Figure 3: RTL-SDR hardware with antenna

The example device used in this work (also listed in the [8]) is the RTL-SDR (Figure 3). It can only act as a receiver, but such a solution is also safe from an educational point of view, as it does not pose the risk of unintentional transmission of radio signals. Due to its popularity, it is supported by GNU Radio, but also by many dedicated applications [3].

4 Example application - FM receiver

An example of using GNU Radio and the RTL-SDR platform can be an FM receiver. In the task, it is possible to achieve several goals simultaneously: designing a receiver using ready-made widgets, performing simulations, receiving physical signals with their visualization, and listening to actual radio stations.

Figure 4 shows a complete block diagram of an FM-band radio receiver realized in the GNU Radio environment. The schematic includes a widget referencing the data source (RTL-SDR Source block), QT GUI Frequency Sink, and QT GUI Waterfall Sink widgets for visualizing the instantaneous frequency spectrum and spectrum history over time, respectively. The colors of the I/O ports indicate the data types on which operations are performed. In addition, the scheme uses blocks corresponding to GUI controls (Variable, QT GUI Range).

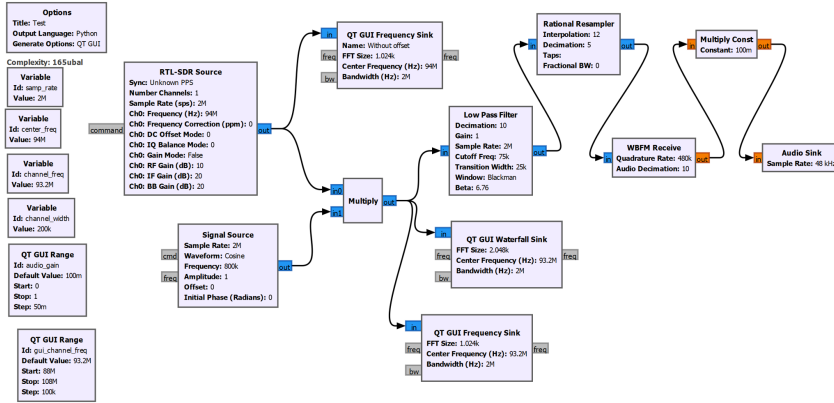


Figure 4: FM receiver GUI

Figure 5 illustrates the system's operation on an actual radio signal using frequency demodulation while receiving a selected radio station. The range (*channel_freq*) control allows frequency tuning during the operation of the running block scheme, while the *audio_gain* control is used to adjust the gain of the demodulated audio signal.

In the GNU Radio environment, it is also possible to save received signals to a file for later analysis. Each of the blocks in the schematic has many properties that can be set by the user when building a virtual circuit, or they can be linked to GUI controls to obtain settings during the operation of a given system.

5 Summary

The open-source GNU Radio environment is undoubtedly an attractive alternative to commercial solutions. Implementing simulations that do not require additional hardware is desirable for teaching the basics of wireless communications. At the same time, the use of available hardware platforms greatly expands the educational value of the whole set. Building the system with the help of a graphical environment (a kind of graphical programming language) can be supplemented in the case of advanced users with programming in Python, which further expands the spectrum of applications of GNU Radio. An essential feature of the GNU Radio environment is also, or primarily, that it is possible to study both existing systems and algorithms, as well as to design or test new solutions concerning, for example, cognitive radio.

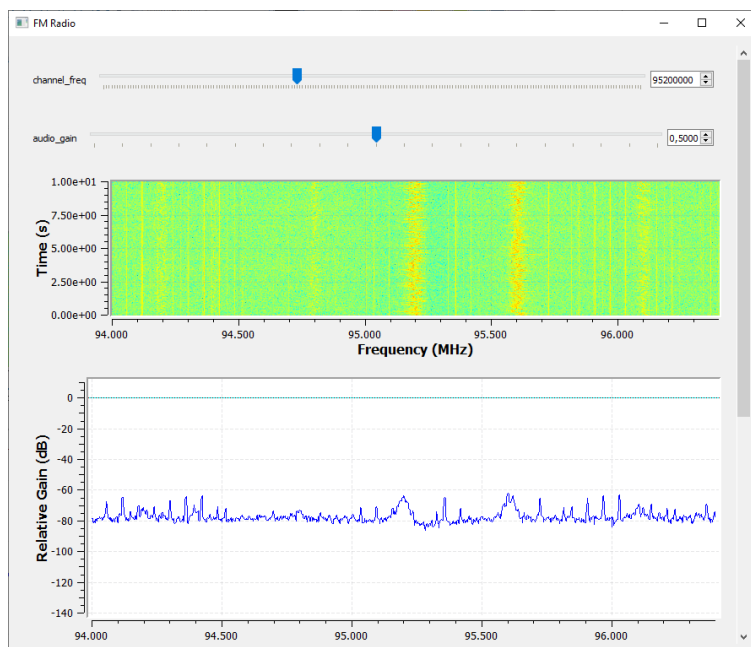


Figure 5: FM receiver GUI

References

- [1] Travis F. Collins, Robin Di Pu Getz, and Alexander M. Wyglinski. *Software-Defined Radio for Engineers*. Artech House, 2018.
- [2] Matthias Föhnle. *Software-Defined Radio with GNU Radio and USRP/2 Hardware Frontend: Setup and FM/GSM Applications*. Ulm: Hochschule Ulm, Bibliothek, 2010.
- [3] Carl Laufer. *The Hobbyist's Guide to the Rtl-Sdr: Really Cheap Software Defined Radio*. CreateSpace Independent Publishing Platform, 2018.
- [4] GNU Radio project. Software, 2022. Accessed on 30.05.2022.
- [5] GNU Radio project. Tutorials, 2022. Accessed on 30.05.2022.
- [6] P. Setoodeh and S. Haykin. *Fundamentals of Cognitive Radio*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2017.

- [7] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, and Louise H. Crockett. *Software Defined Radio using MATLAB & Simulink and the RTL-SDR*. Strathclyde Academic Media, Glasgow, September 2015.
- [8] Bliley Technologies. 10 popular software defined radios (sdrs) of 2022, 2022. Accessed on 30.05.2022.
- [9] Wikipedia. Software-defined radio, 2022. Accessed on 30.05.2022.

Mirosław Łazoryszczak, West Pomeranian University of Technology in Szczecin, Faculty of Computer Science and Information Technology, ul. Żołnierska 49, 71-210 Szczecin, Poland, e-mail: mlazoryszczak@zut.edu.pl

Interaktívna webová aplikácia pre výber a štatistické spracovanie otvorených datasetov

Marcel Puškarik

Abstrakt.

Cieľom textu je predstavenie interaktívnej webovej aplikácie, ktorá umožňuje jednoduché, pre používateľa komfortné, prehľadávanie a ďalšie skúmanie otvorených datasetov a ich dokumentácií, základné štatistické spracovanie a grafickú vizualizáciu.

Abstract.

The aim of this text is to present an interactive web application that enables simple, user-friendly browsing and further exploring of open datasets and their documentation, basic statistical processing and graphic visualization.

Kľúčové slová: Web Application, Python, Panel, Open Data, Open Source

Úvod

V súčasnosti existuje niekoľko webových aplikácií, ktoré sa zaoberajú problematikou zoskupovania už dostupných a nahrávania nových datasetov na podrobnejšie skúmanie. Tieto aplikácie neposkytujú podrobnejšiu analýzu dát. Dávajú k dispozícii iba základné informácie, stručný popis alebo jednoduchý prehľad údajov datasetu. Nezaoberajú sa možnosťou hlbšieho skúmania alebo podrobnejšej analýzy, prípadne štatistického spracovania údajov v rámci webovej aplikácie bez nutnosti sťahovania dát. Práve nedostatok takýchto stránok bol dôvodom vzniku tejto interaktívnej webovej aplikácie.

1 Použité open source knižnice

Na vytvorenie webovej aplikácie bol použitý programovací jazyk Python. Webová aplikácia bola vyvíjaná v prostredí Jupyter Notebook. Grafické rozhranie bolo vytvorené cez knižnicu Panel, ktorá umožňuje tvorbu interaktívnych webových aplikácií a je jednou z najpopulárnejších knižníc na tvorbu webových aplikácií v programovacom jazyku Python. Všetky datasety boli získané pomocou knižnice PyDataset, ktorej zámerom je práve zber a uľahčovanie

prístupu k otvoreným dátam vďaka jednoduchému rozhraniu. Údaje boli vizualizované pomocou knižnice Plotly, ktorá zabezpečuje nekomplikované rozhranie na tvorbu a úpravu interaktívnych, ale aj statických vizuálnych prvkov v podobe grafov, tabuliek alebo iných grafických znázornení údajov. Štatistické spracovanie použitých dát bolo vykonané pomocou knižníc Pandas a Scipy.

1.1 Panel

Panel je otvorená knižnica, ktorá umožňuje vytvárať interaktívne webové aplikácie a dashboards spájaním používateľmi definovanými widgetami [8]. Túto knižnicu je možné spustiť ako výstup Jupyter Notebooku alebo ako samostatný server. Panel je postavený na knižnici Bokeh, ktorá poskytuje webový server založený na knižnici Tornado [7]. Výhodou Panelu je, že je v ňom možné vytvárať a spájať veľké množstvo už definovaných widgetov. Medzi nimi sú aj také, ktoré sú špecifické pre vybrané knižnice, ako napr. samostatný widget pre tvorbu grafov pomocou knižnice Plotly, interaktívna tabuľka knižnice Tabulator alebo widget pre zobrazenie máp vytvorených v knižnici Folium. Vďaka widgetom je možné vytvoriť pokročilú webovú aplikáciu pomerne ľahko a jednoducho. Zabezpečenie interaktivity medzi jednotlivými widgetmi je tiež priamočiare vďaka ich intuitívnemu API.

1.2 Pandas

Pandas [14] poskytuje rýchle a flexibilné štruktúry, ktoré sú navrhnuté tak, aby práca s dátami bola jednoduchá a intuitívna. Knižnica je postavená na knižnici NumPy. Poskytuje dve primárne štruktúry na prácu s dátami, a to **Series** pre jednorozmerné dáta a **DataFrame** pre dvojrozmerné dáta. Cieľom tejto knižnice je stať sa základným vysoko úrovňovým stavebným blokom na vykonávanie praktickej analýzy dát z reálneho sveta. Názov knižnice je odvodený z ‘panel data’, čo označuje “viacrozmerné množiny dát vyskytujúce sa v štatistike a ekonometrii-[5] (preklad autora).

1.3 Scipy

Scipy [13] je knižnica programovacieho jazyka Python, ktorá obsahuje rôzne matematické algoritmy a praktické funkcie. Je postavená na knižnici NumPy. Jej submodul Stats [12] je knižnica s funkciami zameranými na štatistiku. Obsahuje veľké množstvo spojitých, diskretných a viacrozmerných rozdelení pravdepodobností, štatistické testy, ale aj frekvenčnú a súhrnnú štatistiku.

1.4 PyDataset

PyDataset je knižnica v jazyku Python, vďaka ktorej má užívateľ okamžitý prístup k veľkému množstvu otvorených datasetov [1]. Táto knižnica využíva kolekciu datasetov, ktoré boli “pôvodne distribuované spolu so štatistickým softvérovým prostredím R a niektorými z jeho doplnkových balíkov-[2] (preklad autora). Všetky tieto datasety spolu s ich dokumentáciou a ďalšími údajmi sú zverejnené na stránke¹ Vincenta Arel-Bundocka. Celá logika získavania datasetov a dokumentácií je zahrnutá v jednom príkaze. V závislosti od vstupných parametrov je užívateľ schopný získať údaje z datasetu, jeho dokumentáciu alebo tabuľkový zoznam názvov všetkých dostupných datasetov a ich stručných popisov.

Vývoj tejto knižnice sa zastavil pred 6 rokmi, takže neobsahuje všetky datasety, ktoré medzičasom boli zverejnené na vyššie uvedenej stránke. Obsahuje 757 datasetov, pričom na stránke je sprístupnených 1745 datasetov, takže bolo nutné túto knižnicu aktualizovať. Bolo potrebné vytvoriť vlastnú vetvu tejto knižnice a aktualizovať datasety. Boli vykonané drobné úpravy v kóde, ktoré pomohli vyriešiť problémy počas implementácie. Táto aktualizovaná vetva knižnice je dostupná na stránke Github². Bol vykonaný aj pull request na hlavnú vetvu.

1.5 Plotly

Knižnica Plotly [9] je otvorená knižnica, ktorá poskytuje možnosť vytvárať vyše 40 jedinečných typov grafov. Vďaka tejto knižnici je možné vytvárať interaktívne vizualizácie dát a následne ich zobrazíť v Jupyter Notebooku, uložiť do samostatných HTML súborov alebo spustiť ako súčasť webovej aplikácie alebo dokonca uložiť v PDF formáte ako vektorové obrázky. Knižnica obsahuje dve úrovne rozhrania pre používateľa. Vyššia úroveň je dostupná v submodule `express` a umožňuje jednoduchšiu a rýchlejšiu tvorbu grafických znázornení dát vďaka minimalizácii počtu príkazov potrebných na ich vytvorenie. Nevýhodou tejto úrovne je obmedzená možnosť modifikácie vytvorených vizualizácií. Práve preto knižnica Plotly poskytuje aj nižšiu úroveň na tvorbu vizualizácií pre údaje. Tie sú súčasťou modulu `graph_objects`. Tu je užívateľovi umožnené modifikovať si grafy a iné znázornenia dát podľa potreby. Taktiež je mu umožnené pridávať ďalšie vizuálne prvky, ktoré sú nápomocné na zabezpečenie prehľadnosti alebo lepšieho pochopenia grafov, prípadne významu rôznych prvkov daného grafu.

¹<https://vincentarelbundock.github.io/Rdatasets>

²<https://github.com/snap22/PyDataset>

1.6 Statsmodels

Statsmodels je knižnica, ktorá slúži na štatistickú a ekonometrickú analýzu [10]. Poskytuje funkcie a triedy pre odhadovanie rôznych štatistických modelov. Obsahuje triedy a funkcie pre vykonávanie rozličných štatistických testov [11].

2 Rozhranie aplikácie

Celé rozhranie webovej aplikácie je vytvorené pomocou šablóny. Použitá knižnica Panel ponúka viacero predvolených šablón, z ktorých si užívateľ môže vybrať. Všetky predvolené šablóny majú identický vzhľad a skladajú sa z rovnakých komponentov. Štruktúra šablóny sa skladá z nasledujúcich komponentov:

- **Hlavička** (header) - hlavička webovej aplikácie, v ktorej si užívateľ môže zvoliť farbu textu, farbu pozadia a prípadne logo.
- **Bočný panel** (sidebar) - zasúvateľný bočný panel aplikácie, pričom je možné upraviť jeho šírku.
- **Hlavná časť** (main) - predstavuje hlavnú časť webovej aplikácie, kde je možné pridávať ďalšie **widgety**.
- **Modálne okno** (modal) - pomocné vyskakovacie okno, ktoré sa zobrazí pri načítaní stránky

Aj keď štruktúra predvolených šablón vyzerá rovnako, ich rozdiel spočíva v použitých **frameworkoch**, na ktorých sú postavené. Užívateľ si môže vytvoriť v prípade potreby vlastnú šablónu pomocou šablónovacieho jazyka Jinja2.

2.1 Výber datasetu

Bočný panel aplikácie obsahuje zoznam názvov všetkých dostupných datasetov. Po kliknutí na názov sa aktualizuje hlavná časť aplikácie a zobrazia sa informácie ohľadom zvoleného datasetu. V bočnom paneli sa ešte nachádza vyhľadávacie textové políčko na jednoduchšie vyhľadávanie datasetu. Nad týmto políčkom je ešte tlačidlo, po kliknutí ktorého sa zobrazí pomocné vyskakovacie okno. V ňom je prehľadný zoznam všetkých datasetov a ich rozmerov v tabuľkovej podobe.

2.2 Skúmanie datasetu

2.2.1 Sekcie pre všetky datasety

Hlavná časť webovej aplikácie sa skladá zo štyroch spoločných sekcií pre každý dataset. Po zvolení datasetu sa prvotne zobrazí dokumentácia. V nej sa nachádzajú podrobnejšie informácie ohľadom zvoleného datasetu v prehľadnom formáte. Väčšina datasetov obsahuje informácie ohľadom významu jednotlivých stĺpcov. V niektorých datasetoch je aj uvedený zdroj a referencie.

Ďalšou sekciou je prehľad dát zvoleného datasetu v tabuľkovej podobe. V tejto časti sú zobrazené všetky dáta, ktoré dataset obsahuje. Jednotlivé stĺpce je možné zoradiť. Táto časť obsahuje tlačidlo, vďaka ktorému je možné stiahnuť si celý dataset v CSV formáte.

Nasledujúca sekcia obsahuje základné štatistické spracovanie vybraného datasetu. V tejto časti sa zobrazí tabuľka so základnými informáciami ohľadom zvoleného datasetu, ktoré majú pre používateľa informatívny charakter. Vedľa tabuľky sa nachádza bodový graf, v ktorom je možné zvoliť si stĺpce, ktoré sa majú zobraziť na jednotlivých osiach grafu a ďalšie pomocné funkcionality, ako napr. veľkosť, farba a symbol bodov pre lepšiu prehľadnosť. Namiesto bodového grafu je používateľovi umožnené zobraziť si aj korelačnú teplotnú mapu, na ktorej sú zobrazené koeficienty korelácie medzi všetkými dvojicami stĺpcov zvoleného datasetu.

Poslednou sekciou spoločného rozhrania pre všetky datasety je štatistické spracovanie vybraného stĺpca. Tu je umožnené užívateľovi zvoliť si stĺpce, ktoré si želá podrobnejšie preskúmať. V závislosti od typu stĺpca sa mu zobrazí tabuľka obsahujúca sumárnu štatistiku zvoleného stĺpca, stĺpcový diagram, histogram a krabicový graf.

2.2.2 Sekcia ukážkového spracovania vybraných datasetov

Webová aplikácia aktuálne obsahuje 3 datasety, pre ktoré bola vytvorená do-datočná sekcia s ukážkovým spracovaním. Konkrétne sú to datasety Fatality, volcano a speed_gender_height.

Dataset Fatality [3] obsahuje údaje ohľadom zákonov o jazde pod vplyvom alkoholu a o dopravných úmrtiach pre jednotlivé štáty USA v rokoch 1982 až 1988. Pre tento dataset boli vytvorené rôzne stĺpcové diagramy, pomocou ktorých je možné porovnať jednotlivé štáty.

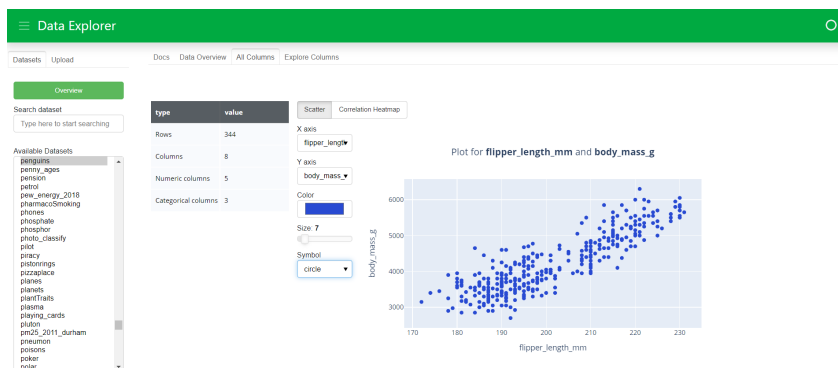
Dataset volcano obsahuje topografické informácie sopky Maunga Whau, ktorá sa nachádza na Novom Zélande [4]. Údaje z tohto datasetu sú ideálne na zobrazenie vo forme 3D povrchu alebo vrstevnice.

Vhodným datasetom pre štatistické spracovanie bol aj dataset speed_gender_height. Dáta sú z prieskumu, v ktorom sa pýtali študentov UCLA na

ich telesnú výšku, pohlavie a najvyššiu rýchlosť, akou kedy v živote jazdili [6]. Primeranosť tohto datasetu spočívala v tom, že na ňom bolo možné vykonať štatistické testy ohľadom normálneho rozdelenia údajov. Užívateľovi je umožnené zvoliť si veľkosť vzorky náhodného výberu. Po získaní náhodného výberu si užívateľ môže zvoliť, či chce skúmať pohlavie alebo rýchlosť a následne si zobrazíť vyhodnotenia štatistických testov ohľadom normalnosti rozdelenia. Môže si zobrazíť vizualizácie v podobe histogramu spojeného s funkciou hustoty rozdelenia pravdepodobnosti normálneho rozdelenia a kvantil-kvantilového grafu.

2.3 Nahrávanie datasetu

V bočnom paneli aplikácie sa nachádza tlačidlo na pridanie vlastného datasetu. Momentálne aplikácia podporuje len datasety v CSV formáte. Tieto datasety sa nahrávajú do aplikácie len dočasne. Po nahratí sa aktualizuje hlavná časť aplikácie pozostávajúca zo sekcií spoločných pre všetky datasety.



Obr. 1: Ukážka aplikácie

3 Záver

Výsledná webová aplikácia je voľne dostupná pre verejnosť³ bez nutnosti registrácie. Služí na jednoduché prehľadávanie a prezeranie viac ako 1700 datasetov. Ponúka používateľom detailný prehľad údajov, ich štatistické sprá-

³Na stránke <https://data.feelmath.eu>

covanie a tiež grafické znázornenie v podobe grafov alebo iných grafických prvkov.

Literatúra

- [1] Alto, A.: PyDataset: Instant access to many datasets in Python, <https://github.com/iamaziz/PyDataset>
- [2] Arel-Bundock, V.: A collection of datasets originally distributed in various R packages, <https://vincentarelbundock.github.io/Rdatasets/>
- [3] Hanck, C. and Arnold, M. and Gerber, A. and Schmelzer, M.: Introduction to Econometrics with R, 2021, <https://www.econometrics-with-r.org/index.html>
- [4] Hengl, T.: Volcano Maungawhau - International Society for Geomorphometry, <https://geomorphometry.org/volcano-maungawhau/>
- [5] McKinney, W.: pandas: a Foundational Python Library for Data Analysis and Statistics, 2011, https://www.dlr.de/sc/Portaldata/15/Resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf
- [6] OpenIntro contributors: Data Sets, https://www.openintro.org/data/index.php?data=speed_gender_height
- [7] Panel contributors: Server Deployment - Panel documentation, https://panel.holoviz.org/user_guide/Server_Deployment.html
- [8] Panel contributors: Panel Documentation, <https://panel.holoviz.org/index.html>
- [9] Plotly: Plotly Python Graphing Library, <https://plotly.com/python/>
- [10] Seabold, S. and Perktold, J.: statsmodels: Econometric and statistical modeling with python, 9th Python in Science Conference, 2010, ISBN 978-1-4583-4619-3
- [11] Seabold, S. and Perktold, J. and Taylor, J.: Introduction - statsmodels, <https://www.statsmodels.org/stable/index.html>
- [12] The SciPy community: Statistical functions (scipy.stats), <https://docs.scipy.org/doc/scipy/reference/stats.html>

- [13] The SciPy community: Introduction - Scipy, <https://docs.scipy.org/doc/scipy/tutorial/general.html>
- [14] The pandas development team: Pandas documentation, <https://pandas.pydata.org/docs/>

Marcel Puškarik, Faculty of Management Sciences and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic, e-mail: marcel.puskarik@gmail.com

Android – Architecture Pattern

Radosław Maciaszczyk

Abstract.

Nowadays, app development requires knowledge of design patterns in addition to knowledge of the platform in question. Android developers generally use the MVVM pattern.

Abstrakt.

W dzisiejszych czasach tworzenie aplikacji oprócz znajomości platformy wymaga znajomości wzorców projektowych. Programiści Androida zazwyczaj używają wzorca MVVM.

Key words: Open Source, Android, MVVM, Design pattern.

Introduction

Learning how to develop applications for newly learned operating systems starts with learning the basic components. Creating user interfaces and adding interactions. As we become more familiar with the environment, we add more elements, including saving data. As simple applications turn into more complex ones, it becomes necessary to structure the application architecture. Programming patterns come to the rescue. In software engineering, a software design pattern is a generic, reusable solution to a commonly occurring problem in a given context in software design. It is not a finished design that can be transformed directly into source or machine code. Rather, it is a description or template of how to solve a problem that can be used in many different situations. Design patterns are formalised best practices that a developer can use to solve common problems when designing an application or system. [1]

Android developers did not originally propose any of the known design patterns, it is only since the Google IO 2017 conference that the Model-View-ViewModel pattern has been proposed, abbreviated as MVVM [2]. The concept of Model is nothing new. It was first defined by Trygve Reenskaug in 1979 as a part of MVC architecture [3].

1 MVVM

MVVM is part of "Android Architecture Components" [6] and introduces three layers:

- **Model** - This layer is responsible for the abstraction of data sources. **Model** and **ViewModel** work together to retrieve and store data. Such a function can be performed by a database together with a framework **Room**.
- **Widok** - The function of this layer is to inform **ViewModel** about user actions. This layer observes **ViewModel** and does not contain any application logic.
- **ViewModel**- Deals with providing model data for the view layer and taking action on a triggered event from the view.

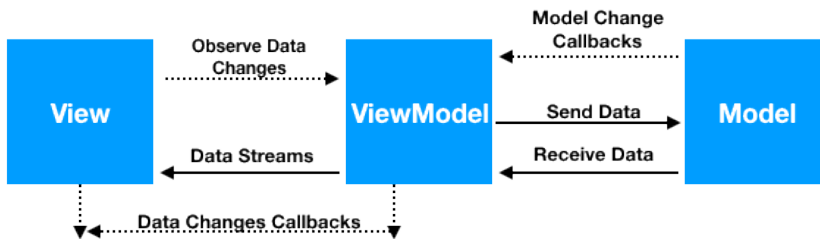


Figure 1: Android MVVM pattern [6]

Its logic is minimised by using a data-binding strategy in the view layer. The code becomes more structured and open to modification, and testing is easier. The idea behind the MVVM pattern is that the view layer observes (Observer pattern) the changing data in the model layer and reacts to the changes through a data binding mechanism.

The Android Jetpack introduces facilities for creating an application that follows the MVVM pattern, in particular, it provides classes:

- ViewModel,
- LiveData, MutableLiveData, MediatorLiveData, or Flow
- Lifecycle.

The below contains good practices for working with components in applications with the MVVM pattern.

UI controllers (activities and fragments) should be as simple as possible. They should not attempt to acquire their data. The UI controller is responsible for updating views when data changes or notifying ViewModel classes of user actions.

In classes that are **ViewModel**, it is recommended to observe objects using **LiveData**. This will allow the user interface to be informed of changes.

It would help if you wrote data-driven user interfaces where the controller **UI** is responsible for updating views when data changes. The **ViewModel** should be notified of user actions.

The data logic should be placed in class **ViewModel**. The **ViewModel** should serve as a link between the UI controller and the rest of the application. However, e.g. IO operations (e.g. download file) should not be implemented in the VM class. Instead, creating separate components that retrieve the data and return the result to the UI controller is recommended. In addition, they must be executed in a non-UI thread. It is recommended to use **data binding** so that the interface between views and the UI controller is transparent and the views become more declarative with a minimum of update code. To manage long-running tasks, use **Kotlin coroutines** [4], which can be executed asynchronously.

2 ViewModel

The main task of the class being **ViewModel** is to maintain information (caching) about the state values of objects on the user interface. It allows the state to persist regardless of the current activity and, when the user returns to the interface associated with a given **ViewModel**, to provide the current values. The second function **ViewModel** is to provide access to the application's business logic. In addition, **ViewModel** is responsible for handling events and forwarding them to other layers. An example of an implementation was presented in the previous chapter in Listing no. 2. The example class extends class *ViewModel*. According to the adopted methodology, this class contains all data operations. Using *LiveData*, it receives all changes to the associated objects. Listing no. 1 shows the code belonging to the user interface layer, which uses the stored data through **ViewModel**. It is also important that many different UIs can access the same VM.

```
class MainActivity : AppCompatActivity() {  
    lateinit var personViewModel: PersonViewModel  
    lateinit var context: Context  
    ...  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        context = this@MainActivity  
  
        personViewModel = ViewModelProvider(this).get(PersonViewModel::class.java)  
  
        btnAddPerson.setOnClickListener {  
            ...  
                personViewModel.insertData(context, person)  
            ...  
        }  
  
        btnGetPersons.setOnClickListener {  
            ...  
        }  
    }  
}
```



```
loginViewModel.getAllPersons().observe(this, new Observer<User>(){
    @Override
    public void onChanged(@Nullable Person data){
        // update ui.
    }
});
}
```

Listing 1: ViewModel Example

3 LiveData

Class **LiveData** - an observable data holder class allows you to create applications that are sensitive to the life cycles of components such as fragments or activities. It notifies the observable of changing data while observing lifecycle changes, updating only active observables. There are subclasses in LiveData that are useful for their properties when updating the UI [?]:

- LiveData - is immutable by default. Using LiveData, we can only observe the data and cannot set the data.
- MutableLiveData - subclass of LiveData. In MutableLiveData, we can observe and set the values using *postValue()* and *setValue()* methods (the former being thread-safe) so that we can dispatch values to any live or active observers.
- MediatorLiveData - can observe other LiveData objects, such as sources and react to their *onChange()* events. MediatorLiveData will give us control over when we want to perform an action in particular or to propagate an event.

Advantages of using **LiveData**:

- Removes the leaks caused by the interfaces/callbacks that send results to the UI thread.
- It de-couples tight integration between data, mediator, and the UI layers.
- Always up to date with the latest data
- Sharing resources, e.g. Extending LiveData

3.1 Using LiveData

We can illustrate the use of **LiveData** in the following steps, with reference in brackets to previously shown examples:

1. Create a LiveData instance in your ViewModel class to hold the data. (Listing no. 2, Line no. 8)
2. Set the data in LiveData. (Listing no. 3, Line no. 26)
3. Return the LiveData. (Listing no. 2, Line no. 8–10)
4. Observe the data with the help of the Observer() function.(Listing no. 1, Line no. 24–30)

```
class PersonViewModel : ViewModel(){
    fun insertData(context: Context, person: Person){
        PersonRepository.insert(context, person)
    }
    fun getAllPersons(context: Context) : LiveData<List<Person>>?{
        return PersonRepository.getAllPersons(context)
    }
}
```

Listing 2: Using the repository class with MVVM [5]

```
class PersonRepository{
    companion object{
        var personalDatabase: PersonalDatabase? = null
        var personDao: PersonDao? = null
    }
    fun initializeDB(context: Context) : PersonalDatabase{
        return PersonalDatabase.getDatasetClient(context)
    }
    fun insertData(context: Context, person: Person){
        personalDatabase = initializeDB(context)
        CoroutineScope(IO).launch{
            personDatabase!!.personDao().insert(person)
        }
    }
    fun getAllPersons(context: Context) : LiveData<List<Person>>?{
        personalDatabase = initializeDB(context)
        return personDatabase!!.personDao().getAllPersons()
    }
}
...
}
```

Listing 3: Repository example [5]

4 Conclusions

Before the MVVM pattern was proposed, Activity and Fragments components were responsible for practically all the application logic. In the code

of these components, there was code related to the creation of the interface, code responsible for the reason and code managing the data. This meant that the classes were very large, and changing the screen's orientation preloaded the interface and destroyed the stored data, resulting in additional handling. The introduction of the MVVM pattern made it possible to divide the code into try layers. This simplified the individual classes and provided a logical division. The code becomes more readable, however, with more interconnections. It is worth mentioning that most of the classes included in the Android Jetpack libraries support the MVVM pattern, making application development easier and faster.

References

- [1] Software design pattern, https://en.wikipedia.org/wiki/Software_design_pattern.
- [2] Lukas Bergstrom, Product Manager, Android Developer Frameworks Team, *Announcing Architecture Components 1.0 Stable*, https://en.wikipedia.org/wiki/Software_design_pattern.
<https://android-developers.googleblog.com/2017/11/announcing-architecture-components-10.html>
- [3] Trygve Reenskaug, *MVC: XEROX PARC 1978-79* <https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>
- [4] *Kotlin coroutines on Android*, <https://developer.android.com/kotlin/coroutines>
- [5] Android's Room in Kotlin ft. MVVM Architecture and Coroutines, <https://github.com/umangburman/MVVM-Room-Kotlin-Example/>
- [6] Guide to app architecture, <https://developer.android.com/topic/architecture>

Radosław Maciaszczyk, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin, ul. Żołnierska 49, 71-210 Szczecin, Poland, e-mail: rmaciaszczyk@zut.edu.pl

Rocket.chat

Ivan Baroňák, Michal Vančo

Abstract.

Rocket.Chat is a free and open source communication platform which can be used as an alternative to Microsoft Teams. It supports instant messaging and audio/video calls including screen sharing and session recording.

Abstrakt.

Rocket.Chat je bezplatná a open source komunikačná platforma, ktorú možno použiť ako alternatívu k Microsoft Teams. Podporuje okamžité správy a audio/video hovory vrátane zdieľania obrazovky a nahrávania relácií.

Key words: Rocket.Chat, Open source, communication platform.

Rocket.Chat features

- Video Conferencing
- Instant messaging
- Multimedia and file sharing
- Digital workspaces via teams, channels, discussions and threads
- Support for 2fa, OAuth, API tokens
- ClamAV antivirus integration
- LDAP support
- KaTeX support – <https://katex.org/docs/supported.html>

B i ↻ </> ↶ \[KaTeX]

Install options

- Rocket.Chat server can be installed on any Linux server
- Official support for Docker and Kubernetes
- Cloud support (AWS, Oracle cloud, Digital Ocean, and others)
- Client applications are available on mobile (Android, iPhone) and desktop (Windows, MacOS, Linux via Snap)

Ivan Baroňák, DiS, <https://ibasterisk.sk/>, e-mail: ivan.baronak@gmail.com

Michal Vančo, DiS, e-mail: vancomail@gmail.com

Prečo učím CLIPS, aj keď sa to zdá mnohým staromódne

Eva Rakovská

Abstrakt.

V rámci študijného programu Hospodárska informatika je potrebné sa zaoberať aj významom využívaním umelej inteligencie v praxi. Príspevok ukazuje aké miesto majú aj deklaratívne programovacie jazyky v pochopení umelej inteligencie, ktorá nie je iba strojové učenie ako sa často v dnešnej dobe prezentuje.

Abstract.

As part of the Business Informatics study program, it is also necessary to deal with the importance of using artificial intelligence in practice. The paper shows the place that declarative programming languages also have in understanding artificial intelligence, which is not just machine learning as it is often presented nowadays.

Kľúčové slová: CLIPS, declarative programming, artificial intelligence, knowledge tasks, expert systems.

Namiesto úvodu

V rámci študijného programu Hospodárska informatika je potrebné sa zaoberať aj významom využívaním umelej inteligencie v praxi. Na to, aby študenti pochopili rozdiely medzi informačnými systémami, ktoré riešia úlohy ľahko algoritmizované, často na relačných databázach a systémoch UI, je dôležité postaviť študentom aj iný spôsob programovania ako ponúka imperatívna paradigma. Deklaratívne programovanie je v súčasnosti vnímané ako "zastaralé", lebo mnohé vyššie programovacie jazyky sú multiparadigmové, avšak veľmi robustné (napr. Python) a teda niekedy ťažko pochopiteľné pre študentov.

Princípy deklaratívneho programovania sú pre študentov odchovaných na klasických algoritmoch a princípoch objektovo-orientovaného programovania je dosť, a preto je najlepšie vysvetľovať ich na pravidlových systémoch. Programovací jazyk CLIPS s jeho prostredím je vhodným nástrojom na to, aby študenti pochopili nielen programovanie pravidelných systémov, ale aj súvis týchto systémov s algoritmiami prehládávania stavového priestoru. CLIPS je "public domain software", ktorý prešiel postupným vývojom od roku 1985


(NASA) a v súčasnosti okrem pravidiel podporuje aj objekty orientovanej verzie COOL, či modulárne programovanie. Taktiež študenti môžu vidieť v CLIPSe klasické reprezentácie znalostí v umelej inteligencii. Najnovšia verzia 6.40 vznikla v roku 2021 a teda CLIPS stále sleduje nové trendy v programovaní.

Príspevok ukazuje ako je možné rozvíjať kreativitu v programovaní v prostredí CLIPS u študentov Hospodárskej informatiky na projektoch, ktoré riešia tzv. znalostné úlohy v manažmente (plánovanie, rozvrhovanie, predvídanie, klasifikácia, priradovanie, hodnotenie, navrhovanie, monitorovanie a pod.) a používaním staršej verzie CLIPS verzia 6.31 z roku 2019 študenti môžu nahliadnuť "do vnútra" práce znalostných a expertných systémov. Riešenia mnohých úloh napr. úlohy rozvrhovania, úloha obchodného cestujúceho, či klasické úlohy ako hry (farbenie slepej mapy, hanojské veže a pod.) sú veľmi efektívne práve v programovacom jazyku CLIPS.


Príspevok ukazuje aké miesto majú aj deklaratívne programovacie jazyky v pochopení umelej inteligencie, ktorá nie je iba strojové učenie ako sa často v dnešnej dobe prezentuje.

V rámci študijného programu Hospodárska informatika je potrebné sa zaoberať aj významom využívaním umelej inteligencie v praxi. Príspevok ukazuje aké miesto majú aj deklaratívne programovacie jazyky v pochopení umelej inteligencie, ktorá nie je iba strojové učenie ako sa často v dnešnej dobe prezentuje.

Prezentácia príspevku

<p>CLIPS 6.4 User's Guide</p>  <p>by Joseph C. Giarratano, Ph.D.</p>	<p>Prečo učím CLIPS, aj keď sa to zdá mnohým staromódne</p> <p>Eva Rakovská Fakulta hospodárskej informatiky, Ekonomická Univerzita Katedra aplikovanej informatiky eva.rakovska@euba.sk</p>
---	---

CLIPS 6.4
User's Guide



by Joseph C. Giarratani, Ph.D.

Študijný program 1. st. Hospodárska informatika


- Zameranie na prepojenie vedomostí z ekonomiky a informatiky
- V súčasnosti IT je súčasťou podnikových procesov, biznis logiky, administratívy a manažmentu
- HI je orientovaná na pochopenie fungovania algoritmov a IS v podnikovej praxi, tvorbu web aplikácií, základných vedomostí zo sietí, OS, databáz a princípov UI a teórie rozhodovania, aby študenti vnímali rozhodovacie procesy ako kľúčové pri podnikových procesoch
- Zameranie na 4 hlavné oblasti:
 - **Algoritmizácia a schopnosť programovať** a rozumieť tomu, čo je v zdrojovom kóde
 - **Orientácia na prácu s dátami** (od MS Access cez relačné databázy po grafové daabázy)
 - **IT základy** - operačné systémy, siete, bezpečnosť IS
 - **Rozhodovacie procesy** ako základ pre pochopenie systémov na podporu rozhodovania, rôzne úrovne riadenia podniku (predmet Hospodárska informatika, Podpora rozhodovania, Umeľá inteligencia a expertné systémy)

Eva Rakovská IOSCS 2022 2

Umeľá inteligencia a expertné systémy

- **Prečo UIES?**
- Cieľom predmetu je poskytnúť študentom celkový pohľad na UI a bližšie ich oboznámiť s významom reprezentácií poznatkov a deklaratívneho programovania. Naučiť ich analyzovať **znalostné úlohy (analytické a syntetické)** a vytvoriť demoverzie programov prostredníctvom pravidlovej paradigmy.
- UI je v dnešnej dobe základom pre získavanie znalostí z dát, Big Data sú trend, potrebné je narábať s heuristikami
- Pravidlové systémy základ asociačných pravidiel, fuzzy logiky a tzv. Business Rule Engine (BRE) alebo workflow


CLIPS 6.4
User's Guide



by Joseph C. Giarratani, Ph.D.

Eva Rakovská IOSCS 2022 3

CLIPS 6.4
User's Guide



by Joseph C. Giarratani, Ph.D.

Umeľá inteligencia a expertné systémy

Výsledky vzdelávania z predmetu:

- Orientovať sa v oblasti Umelej inteligencie a jej použitia v hospodárskej praxi
- Porozumieť **princípom deklaratívneho programovania a rozdielnosti tvorby takéhoto programu oproti procedurálnemu programovaniu**
- Rozlíšiť **znalostné úlohy v praxi**, analyzovať ich a navrhnúť typ systému, ktorý by ich mohol riešiť
- Orientovať sa v používaní znalostných a expertných systémov v praxi s ohľadom na základné funkcie
- Porozumieť základným technikám získavania znalostí od doménového experta a životnému cyklu tvorby expertného systému vo firemnej praxi;
- Vedieť prakticky odpovedať na základné otázky znalostného inžinierstva pri nasadzovaní systémov v praxi
- Naprogramovať jednoduché programy v jazyku podporujúcom pravidlovú paradigmu a rozumieť prepojeniu s rozhraniami na aplikácie tvorené v iných programovacích jazykoch

Eva Rakovská IOSCS 2022 4

CLIPS C Language Integrated Production System

Gary Riley má viac ako tridsaťpäť rokov skúseností s programovaním založeným na pravidlách a expertnými systémami.

- A Tool for Building Expert Systems <https://clipsrules.net/>
- Vývoj verzií, stále dostupné verzie 6.24, 6.30, 6.31, 6.40
- Aktuálny počet sťahnutí: 682/ týždeň



„Integrovaný produkčný systém jazyka C (CLIPS) vyvinutý v Johnsonovom vesmírnom stredisku NASA v rokoch 1985 až 1996 je programovací jazyk založený na pravidlách užitočný na vytváranie expertných systémov a iných programov, kde sa **heuristické riešenie ľahšie implementuje** a udržiava ako algoritmičné riešenie. CLIPS je napísaný v jazyku C kvôli prenositeľnosti a možno ho nainštalovať a používať na širokej škále platformí. Od roku 1996 je CLIPS dostupný ako verejný softvér.“ (Riley, 2022)

Eva Rakovská IOSCS 2022

8

Výhody CLIPSu

Existujú tri spôsoby reprezentácie poznatkov v CLIPS:

- **Pravidlá**, ktoré sú primárne určené pre heuristické poznanie založené na skúsenostiach.
- **Užívateľom definované funkcie a generické funkcie**, ktoré sú primárne určené pre procedurálne poznatky.
- **Objektovo orientované programovanie**, tiež primárne určené pre procedurálne znalosti. Podporovaných je päť všeobecne akceptovaných vlastností objektovo orientovaného programovania: triedy, obslužné programy správ, abstrakcia, zapuzdrenie, dedičnosť a polymorfizmus. Pravidlá sa môžu zhodovať so vzormi objektov a faktov.

CLIPS 6.4
User's Guide

Eva Rakovská IOSCS 2022

9

Výhody CLIPSu verzia 6.31

CLIPS 6.4 User's Guide

Dialogové okno

Vidím aktivované pravidlá

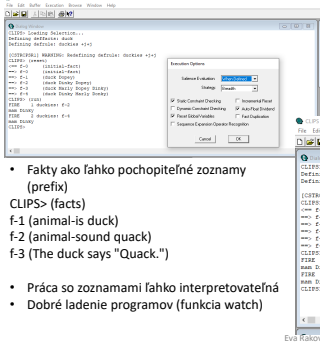
Vidím fakty

Tvorba bázy poznatkov
Deklaratívny program

Eva Rakovská IOSCS 2022

10

Výhody CLIPSu verzia 6.31



```

CLIPS> (facts)
f-1 (animal-is duck)
f-2 (animal-sound quack)
f-3 (The duck says "Quack.")

CLIPS> (watch)
CLIPS> (watch)

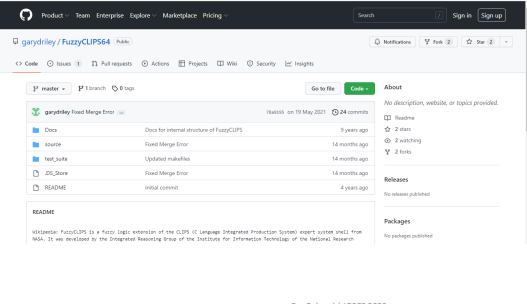
```

Produktívny systém na rozhraní medzi programovacím jazykom a prázdny ES, využívajúci dopredné reťazenie (data driven inferencing). Možnosť nastaviť si stratégiu prehľadávania stavového priestoru (do hĺbky, do šírky, heuristické), práca s duplikovanými faktami, nastavenie spôsobu vyhodnocovania pravidiel tzv. salience).

- Faktky ako ľahko pochopiteľné zoznamy (prefix)
- CLIPS> (facts)
- f-1 (animal-is duck)
- f-2 (animal-sound quack)
- f-3 (The duck says "Quack.")
- Práca so zoznamami ľahko interpretovateľná
- Dobré ladenie programov (funkcia watch)

Eva Rakováková IOSCS 2022 11

Fuzzy CLIPS



Product Team Enterprise Explore Marketplace Pricing

garydille / FuzzyCLIPS64

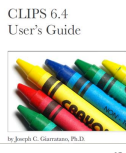
Go to file **Code** About

main · master

- main · Updated README
- main · Updated README
- main · Updated README
- main · Updated README
- main · Updated README
- main · Updated README

README

Installation: FuzzyCLIPS is a Fuzzy Logic extension of the CLIPS (C Language Integrated Production System) expert system shell from Sage. It was developed in the Intelligent Reasoning Group of the Institute for Information Technology of the National Institute.




CLIPS 6.4
User's Guide

by Joseph C. Giarratano, Ph.D.

Eva Rakováková IOSCS 2022 12

Čo na záver? Výhody pri učení



- Výborná dokumentácia a podpora na stránke <https://clipsrules.net/>
- Dokumentácia od User's guide cez Basic Programming Guide, Advanced Programming Guide až po interfaces Guide
- Support Stack overflow, CLIPSESG –CLIPS expert system group, CLIPS sourceforge forum
- Prenositeľnosť na rôzne OS, celý CLIPS môžete mať na USB kľúči
- Vývoj rozhraní od CLIPS Windows, CLIPSu pre PHP, cez CLIPS JNI, CLIPS .NET, Fuzzy CLIPS a pod.

- Možnosť učiť od jednoduchých pravidiel po kombinovanie pravidiel s rámcami, procedúrami a OOP
- Posilnenie princípov OOP, kde vidia „vnútro“ ako fungujú zapuzdrené triedy
- Veľká kreativita pri tvorbe programov (ak nerobia copy-paste)
- Projekty, ktoré dávam podporujú kreativitu. Zadania typu: urobte systém na plánovanie, predikciu, diagnostický systém, systém na priradovanie, na tvorbu rozvrhu, monitorovanie, hodnotiaci systém (podpora rozhodovania) a pod.


Eva Rakováková IOSCS 2022 13

CLIPS ako ďalej?

Univerzity:
Španielsko (Polytechnika
Barcelona), Jordánsko, Poľsko,
Taiwan, USA, GB a krajiny ako
India, Pakistan...

Ďakujem za pozornosť!

CLIPS 6.4
User's Guide



by Joseph C. Giarratano, Ph.D.

Eva Rakovská IOSCS 2022 14

References

- [1] Riley, G., *Adventures in Rule-Based Programming*, (2022)
- [2] CLIPS, *A Tool for Building Expert Systems*, <https://clipsrules.net/>
- [3] Russell, S., Norvig, P.: , *Artificial Intelligence: A Modern Approach* 4th Edition. Pearson.(2021) ISBN 1292401133
- [4] Schreiber, G. at col.: *Knowledge Engineering and Management: The CommonKADS Methodology*, The MIT Press, 1999, 9780262283236

Eva Rakovská, Department of Applied Informatics, Faculty of Economic Informatics, University of Economics in Bratislava, Dolnozemska 1, 852 35 Bratislava, Slovak Republic, e-mail: eva.rakovska@euba.sk

Pravdepodobnosť a štatistika v R

Základy grafiky

Aleš Kozubík

Abstrakt.

Príspevok je venovaný príprave základného kurzu pravdepodobnosti a štatistiky s podporou open source programovacieho prostredia R. Zaoberá sa základmi grafického reprezentovania štatistických výsledkov.

Abstract.

The paper is concerned with preparing an elementary course in probability and statistics with the support of the open-source programming environment R. It deals with the basics of graphical representation of statistical results.

Kľúčové slová: Štatistika, Open Source, R, dáta, graf.

Bodové grafy

Bodový (tiež korelačný) graf zobrazuje hodnoty dvoch rôznych číselných premenných. Poloha každého bodu zodpovedá jednotlivým dátovým hodnotám na vodorovnej a zvislej osi. Medzi najčastejšie použitie bodových grafov patrí:

1. Demonštrácia vzťahu medzi dvoma premennými.
2. Identifikácia korelačných vzťahov.
3. Identifikácia dátových modelov.

Vytvoríme ho jednoducho pomocou funkcie `plot()`. Pri najjednoduchšom použití má funkcia dva argumenty `x` a `y`. Tieto premenné sú vektory, ktoré obsahujú hodnoty, ktoré chceme vykresliť. Dĺžka vektorov musí byť rovnaká.

Príklad.

Predpokladajme, že zmrzlináreň sleduje, koľko zmrzliny predá v závislosti od poludňajšej teploty v daný deň. Tu sú ich údaje za posledných 10 dní:

Teplota	28	30.2	32	31	29.5	26	31.5	30	29	34
Tržby (€)	540	560	530	570	525	490	530	530	500	580

Najprv definujeme dva číselné vektory:

`x`, ktorý obsahuje teploty.

`y`, ktorý bude predstavovať denný objem predaja.

Potom nakreslíme bodový graf.

```

1 > x<-c(28,30.2,32,31,29.5,26,31.5,30,29,34)
2 > y<-c(540,560,530,570,525,490,530,530,500,580)
3 > plot(x,y)

```

Môžeme použiť príkaz `dev.copy()`, aby sme uložili obsah grafického okna do súboru bez nutnosti opätovného zadávania príkazov. Ak chceme vytvoriť súbor `newplot.png` z nášho grafu, zadáme:

```

1 > dev.copy(png,'newplot.png')
2 > dev.off()

```

Taktiež môžeme presmerovať výstup z obrazovky do súboru, napríklad:

<code>pdf()</code>	Vector pdf formát, najlepšia voľba pri použití s <code>pdflatex</code>
<code>svg()</code>	Vektorový svg formát, ľahko sa mení veľkosť.
<code>postscript()</code>	Vektorový formát postscript ps, ľahko meniteľná veľkosť.
<code>jpeg()</code>	Komprimovaný bitmapový formát.
<code>png()</code> , <code>bmp()</code>	Bitmapový formát, vysoké rozlíšenie.
<code>filename</code>	Názov uloženého súboru (prípadne s úplnou cestou).
<code>width</code> , <code>height</code>	Šírka, výška výsledného grafu (implicitne 7 in).
<code>res</code>	Rozlíšenie pre bitmapové formáty (implicitne 72 dpi).
<code>units</code>	Merné jednotky.
<code>bg</code> , <code>fg</code>	Farba pozadia, popredia.
<code>family</code>	Použité písmo (predvolené Helvetica).

Značka bodov je daná hodnotou argumentu `pch` funkcie `plot()`.

□ pch=0	○ pch=1	△ pch=2	+ pch=3	× pch=4
◇ pch=5	▽ pch=6	⊠ pch=7	* pch=8	◊ pch=9
⊕ pch=10	☆ pch=11	⊞ pch=12	⊗ pch=13	⊠ pch=14
■ pch=15	● pch=16	▲ pch=17	◆ pch=18	● pch=19
• pch=20	○ pch=21	□ pch=22	◇ pch=23	△ pch=24

```

1 > plot(x,y,pch=17)
2 > plot(x,y,pch=1)

```

Typ spojnice sa nastavuje pomocou argumentu `type` funkcie `plot()`.

- p Bodový graf, predvolená hodnota.
- l Spojitá čiara.
- b Spojitá čiara s bodmi.
- c Časti spojitých čiar s vynechanými bodmi.
- o Časti súvislých čiar s prekreslenými bodmi.
- h Graf podobný histogramu.
- s Schodovitý graf.

```

1 > plot(x,y,type="l")
2 > dev.off()
3 > plot(x,y,type="s")
4 > dev.off()
5 > plot(x,y,pch=17,type="b")
6 > dev.off()

```

Štýl čiary sa nastavuje pomocou argumentu `lty` funkcie `plot()`.

- 1 Plná čiara (predvolené). 2 Čiarkovaná čiara.
- 3 Bodkovaná čiara. 4 Bodko-čiarkovaná čiara.
- 5 Dlhé čiarky. 6 Dlhá a krátka dvojité prerušovaná čiara.

Šírku čiary nastaví argument `lwd` funkcie `plot()`.

```

1 > plot(x,y,type="l",lty=5)
2 > dev.off()
3 > plot(x,y,type="l",lty=1,lwd=2)
4 > dev.off()

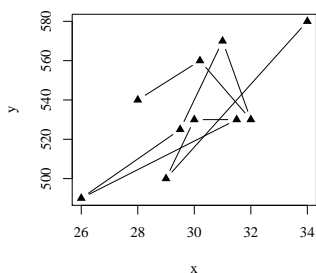
```

```

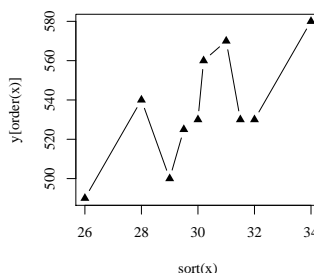
1 > plot(x,y,pch=17,type="l")

```

Dáva:



Ale my chceme:



Použijeme funkcie `sort()` a `order()`.

```

1 > plot(sort(x),y[order(x)],pch=17,type="b")
2 > dev.off()

```

Nastavenie farieb môžeme vykonať pomocou mena farby (napr. `col=red`), číslom farby (napr. `col=636`), podľa hexadecimálneho kódu (v režime RGB, napr. `col="#FFCC00"`). Zoznam dostupných farieb získame ako odpoveď funkcie `colors()`.

```
1 > plot(sort(x),y[order(x)],pch=17,type="b",col="red")
2 > dev.off()
3 > plot(sort(x),y[order(x)],pch=17,type="b",col=636)
4 > dev.off()
5 > plot(sort(x),y[order(x)],pch=17,type="b",col="#FFCC00")
6 > dev.off()
```

Ďalšie možnosti vyfarbenia sú:

<code>col.axis</code>	Farba popisu osí.	<code>col.lab</code>	Farba označení osí.
<code>col.main</code>	Farba hlavného nadpisu.	<code>col.sub</code>	Farba podnadpisu.
<code>bg</code>	Farba výplne znakov.	<code>fg</code>	Farba podkladu.

```
1 >plot(sort(x),y[order(x)],lty =1, type ="b",
2 col="aquamarine",lwd =2,
3 col.axis ="violet",col.main ="green",
4 main ="Main_title",fg="red",
5 col.lab="coral3",pch=17)
6 > dev.off()
7 >par(bg="beige")
8 >plot(sort(x),y[order(x)],lty =1, type ="b",col=30,lwd =2,
9 col.axis ="darkmagenta", col.main ="blue3",col.sub="blue2",
10 main ="Main_title", sub="Subtitle", fg="red",
11 col.lab="coral4",pch=17)
12 > dev.off()
```

Hodnotu argumentu `col` môžeme nastaviť ako vektor. Farby z vektora sa pravidelne menia. Môžeme tiež použiť funkciu `rainbow()` s preddefinovanou postupnosťou farieb.

```
1 > plot(sort(x),y[order(x)],pch=17,type="b",
2 col=c("red", "blue"))
3 > dev.off()
4 >plot(sort(x),y[order(x)],pch=17,type="b",col=rainbow(5))
5 > dev.off()
```

Základné kresliace funkcie v R obsahujú argument s názvom `main`, ktorý umožňuje pridať do grafu nadpis. Pomocou argumentu `sub` je možné pridať aj podnadpis, ktorý bude umiestnený pod grafom. Alternatívnym spôsobom, ako pridať nadpis a podnadpis do grafu, je použitie funkcie `title()`.

```
1 > plot(sort(x),y[order(x)],pch=17,type="b",col=rainbow(4))
2 > title(main="Icecream_sales",col.main="red")
3 > title(sub="Temperature",col.sub="blue",adj=1,line=2)
4 > dev.off()
```

Do grafu môžeme pridať akékoľvek texty pomocou funkcií `text()` a `mtext()`. Funkcia `text()` umiestni daný text na ľubovoľné miesto v kresliacej oblasti, funkcia `mtext()` umiestni text na okraje. Funkcia `text()` má dva ďalšie argumenty:

- `location` definuje súradnice `x` a `y`, kde bude text umiestnený. Súradnice musia byť uvedené ako prvé dva argumenty funkcie.
- `pos` definuje pozíciu podľa aktuálneho miesta, 1=dole, 2=vľavo, 3=hore a 4=vpravo. Definovanie pozície ako `locator(1)` umožňuje umiestnenie textu pomocou myši.

```
1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30)
2 > title(main="Icecream sales", col.main="red")
3 > title(sub="Temperature", col.sub="blue", adj=1, line=2)
4 > text(c(28, 32), c(560, 500), c("Text1", "Text2"), pos=1, col="red")
5 > dev.off()
```

Funkcia `mtext()` má ďalšie dva argumenty:

- `side` definuje stranu plochy grafu, kam umiestnime textový štítok, 1=dole, 2=vľavo, 3=hore a 4=vpravo.
- `line` definuje číslo riadku, kde bude umiestnený štítok. Riadky sú číslované od 0.

```
1 > plot(sort(x), y[order(x)], pch=17, type="b",
2 + col=30, xlab="", ylab="")
3 > mtext("Temperature", side=1, line=2, adj=1)
4 > mtext("Sales", side=2, line=2)
```

Ak chceme odstrániť rámček grafu, nastavíme vo vnútri kresliacej funkcie možnosť `axes=FALSE`. Nové osi pridáme pomocou funkcie `axis()`. Argument funkcie `axis()` definuje stranu grafu, na ktorú bude pridaná os. Ako zvyčajne, čísla definujú strany 1=spodná, 2=ľavá, 3=horná a 4=pravá.

```
1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE)
2 > axis(1)
3 > axis(2)
```

Ďalším prispôbením je zmena farieb osí. Môžeme to urobiť nastavením voliteľných argumentov funkcie `axis()`:

- `col` definuje farbu čiary osí,
- `col.ticks` definuje farbu deliacich bodov,
- `col.axis` definuje farbu značiek.


```

1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE)
2 > axis(1, col="blue", col.ticks="red", col.axis=555)
3 > axis(2, col="deepskyblue2", col.ticks=444, col.axis="red")

```

Môžeme tiež:

- nastaviť počet značiek so zadanými hodnotami začiatku a konca,
- upraviť dĺžku a orientáciu značiek,
- otáčať popisy značiek,
- prispôbiť popisy značiek,
- odstrániť značky,
- pridať menšie značky pomocou `Hmisc`.

Argumenty `xaxp` a `yaxp` umožňujú prispôbiť pozície deliacich značiek na osi `x` a `y`. Ich hodnoty nastavujeme ako vektory `c(start, end, regions)`, `start` a `end` definujú počiatočnú a koncovú hodnotu na každej osi a hodnota `regions` definuje počet intervalov na rozdelenie osi.

```

1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE)
2 > axis(1, col="blue", col.ticks="red",
3   col.axis=555, xaxp=c(26, 34, 8))
4 > axis(2, col="blue", col.ticks="red",
5   col.axis=555, yaxp=c(490, 580, 9))

```

Argument `tck` umožňuje upraviť dĺžku a orientáciu deliacich značiek. Jeho kladná hodnota nastavuje značky do vnútri kresliacej oblasti, kým záporné hodnoty definujú značky von z kresliacej oblasti. Čím väčšia je absolútna hodnota, tým dlhšie sú značky. predvolená hodnota je `tck=-0,05`. Otáčanie je povolené pomocou argumentu `las`, ktorý môže nadobúdať jednu z hodnôt:

- `las=0` štítky sú rovnobežné s osou (predvolené),
- `las=1` všetky štítky sú vodorovné,
- `las=2` štítky sú kolmé na os,
- `las=3` všetky štítky sú vertikálne.

```

1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE)
2 > axis(1, col="blue", xaxp=c(26, 34, 8), tck=0.02, las=3)
3 > axis(2, col="blue", yaxp=c(490, 580, 9), tck=0.02, las=2)

```

Deliace značky môžeme úplne odstrániť nastavením argumentov `xaxt="n"` pre os `x` alebo `yaxt="n"` pre os `y`. Popisy deliacich značiek je možné zmeniť pomocou argumentu `labels` funkcie `axis()`. Aby sme správne umiestnili popisy, musíme nastaviť ich polohu pomocou argumentu `at`.

```

1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE)
2 > axis(1, col="blue", at=seq(round(min(x)), round(max(x)), by=1),
3 + labels=0:8)
4 > axis(2, col="blue", yaxp=c(490, 580, 9), tck=0.02, las=2)

```

Rozsah hodnôt pre osi môžeme definovať pomocou voliteľných argumentov `xlim` a `ylim` funkcie `plot()`. Hranice sa zadávajú ako vektory v tvare `c(start, end)`. Osi môžeme tiež transformovať do logaritmickkej mierky nastavením argumentu `log` tak, aby sa rovnal osi, ktorú plánujeme prispôbiť.

```

log="x " nastaví logaritmickú stupnicu na os x,
log="y " nastaví logaritmickú mierku na os y,
log="xy " transformuje obe osi do logaritmickkej stupnice.

```

```

1 > plot(sort(x), y[order(x)], pch=17, type="b", col=30, axes=FALSE,
2 + ylim=c(400, 600))
3 > axis(1, col="blue", at=seq(round(min(x)), round(max(x)), by=1),
4 + labels=0:8)
5 > axis(2, col="blue", yaxp=c(490, 580, 9), tck=0.02, las=2)

```

Príklad.

Do toho istého grafu chceme zakresliť dve charakteristiky zdravotného stavu pacientov, teplotu a krvný tlak. Máme údaje o 100 pacientoch uložené v premenných `y` a `z`, pričom premenná `x` obsahuje postupnosť identifikátorov pacientov, čísla od 1 do 100.

Najprv pomocou `par(mar = c(3, 4, 2, 4))` upravíme okraje kresliacej oblasti. Potom vykreslíme bodový graf nameraných teplôt. Dôležitým krokom je nastavenie nového grafu pomocou `par(new=TRUE)`. Teraz sme pripravení vykresliť druhý súbor údajov modrou farbou, bez rámečkov a bez osí.

Duálna os `y` sa vykreslí pomocou funkcie `axis(4)` na pravej strane grafu.

```

1 x<-1:100 # generovanie udajov
2 y <- runif(100, min = 35, max = 40)
3 z <- y+10*runif(100, min=7, max=12)
4 par(mar = c(3, 4, 2, 4))
5 plot(x, y, pch = 19, ylab = "Temperature")
6 par(new=TRUE)
7 plot(x, z, col = 4, pch = 19,
8 axes = FALSE, # Bez osi
9 bty = "n", # Bez ramceka
10 xlab = "", ylab = "")
11 axis(4)
12 mtext("Blood pressure", side = 4, line = 3, col = 4)

```

Jednou z mnohých praktických funkcií v R je `curve()`. Je to malá šikovníka funkcia, ktorá umožňuje vykresľovanie kriviek, napr. grafy funkcií. Funkcia `curve()` prijíma ako prvý argument výraz v syntaxi R, napr.:

```
curve(x^2)
curve(x^2,xlim=c(-2,2),col="red",lwd=2)
```

Funkciu `curve()` používame s argumentom `add=TRUE`, napr.:

```
curve(x^2)
curve(sqrt(x),col="red",lwd=2,add=TRUE)
```

Použitie funkcie `curve()` nie je obmedzené iba na jej samostatné použitie. Možno vykresliť nejaké údaje a potom použiť funkciu `curve()` na nakreslenie ľubovoľnej čiary nad nimi.

```
1 set.seed(1)
2 x <- rnorm(100)
3 y <- x^2 + rnorm(100)
4 plot(y ~ x)
5 curve(x^2,add=TRUE)
```

Funkcia `legend()` umožňuje pridať legendu ku grafom v R.

- `x`, `y` pozícia v kresliacej oblasti definovaná súradnicami v grafe,
- `legend` vektor reťazcov pre popis v legende,
- `col` vektor farieb použitých v grafe,
- `pch` vektor tvarov značiek použitých v grafe,
- `lty` vektor typov čiar použitých v grafe,
- `ncol` počet stĺpcov použitých v legende (implicitne jeden stĺpec).

Príklad.

Vytvoríme používateľom definovanú funkciu `gonplot()`, ktorá nakreslí grafy $\sin x$ a $\cos x$ v rozsahu $(-10; 10)$ v dvoch farbách a rôznych typoch čiar. Potom ku grafu pridáme legendu. Užívateľom definovaná funkcia:

```
1 gonplot <- function() {
2   curve(sin(x),xlim=c(-10,10),col="red",lwd=2,type="l",
3     ylab="sin_x",xlab="",ylim=c(-1,2))
4   curve(cos(x),xlim=c(-10,10),col="blue",lwd=2,type="l",lty=2,
5     ylab="sin_x",xlab="",add=TRUE)
6 }
```

Zobrazenie grafu a pridanie legendy.

```
1 gonplot()
2 legend(x = "topright",           # Poloha
3       legend = c("sin_x", "cos_x"), # Text legendy
4       lty = c(1, 2),             # Typ ciar
5       col = c("red", "blue"),    # Farby ciar
6       lwd = 2)
```

Argument polohy `x` môže byť nastavený na jednu z hodnôt: `top`, `topleft`, `topright`, `bottom`, `bottomleft`, `bottomright`, `left`, `right` alebo `center`. Tento scenár nevyžaduje nastavenie argumentu `y`, pretože pozícia legendy je presne určená slovné.

Stĺpcové grafy

Stĺpcový graf zobrazuje kategoriálne údaje pomocou obdĺžnikových stĺpcov s výškou alebo dĺžkou úmernou hodnotám, ktoré predstavujú. Na ich tvorbu sa používa `barplot(H,xlab,ylab,title, names.arg,col)` s parametrami:

- `H` je vektor alebo matica obsahujúca číselné hodnoty použité v stĺpcovom grafe,
- `xlab` je označenie osi `x`,
- `ylab` je označenie osi `y`,
- `title` je nadpis stĺpcového grafu,
- `names.arg` je vektor popisov, ktoré sa zobrazujú pod každým stĺpcom,
- `col` sa používa na priradenie farieb stĺpcov v grafe.

Predpokladajme, že vektor `x` obsahuje denný predaj niektorých výrobkov. Objemy predaja možno graficky znázorniť vo forme stĺpcového grafu.

```
1 x<-c(2000,2400,1400,2600)
2 barplot(x)
```

Argument `horiz=T` nastavíme na hodnotu `true`.

```
1 barplot(x,horiz=T)
```

Na priradenie názvov stĺpcov použijeme parameter `names.arg` stĺpcového grafu. Ďalej definujeme hodnoty parametrov:

- `xlab` a `ylab` pre názvy osí,
- `col` a `border` na vyfarbenie stĺpcov,
- `main` na definovanie nadpisu grafu.

Je to podobné ako v prípade funkcie `plot()`. Nech náš vektor `x` predstavuje denný predaj nejakého ovocia. Ich názvy nastavíme ako vektor `goods` a použijeme ho na priradenie názvov stĺpcov.

```
1 goods<-c("orange","banana","apple","plum")
2 barplot(x,names.arg=goods,xlab="Fruit",ylab="Sales",
3 col="cyan",main="Monthly sale",border="black")
```

Graf môžeme upraviť pomocou rôznych farieb stĺpcov. Požadované farby nastavíme ako vektor `colours` a použijeme ho ako hodnotu argumentu `col`. Argument `border` definuje farbu okraja stĺpcov.

```
1 colours<-c("orange","yellow","red","blue")
2 barplot(x,names.arg=goods,xlab="Fruit",ylab="Sales",
3 col=colours,main="Monthly sale",border="black")
```

Pomocou matice ako vstupných hodnôt môžeme farebne vyznačovať a značkovat podiely v stĺpcoch. Rozsah predajných objemov vo vektore `x` za viac mesiacov. Následne informácie prezentujeme graficky. Najprv nastavíme:

```
1 months<-c("Jan","Feb","Mar","Apr")
2 x<-matrix(c(2000,2400,1400,2600,1800,2200,1600,2400,2100,
3 2300,1500,2400,2400, 1800,1200,2200),nrow=4,ncol=4)
```

Teraz sme pripravení nakresliť graf s podielmi. Pridáme aj legendu.

```
1 barplot(x, main = "Sale volumes", names.arg = months,
2 xlab = "Month", ylab = "Sales",
3 col = colours, ylim=c(0,11000))
4 legend("topright", goods, fill = colours,ncol=2)
```

Rovnaké informácie môžeme alternatívne prezentovať zoskupením stĺpcov v grafe. Nastavíme argument `beside=T`.

```
1 barplot(x, beside=T,main = "Sale volumes",
2 names.arg = months, xlab = "Month", ylab = "Sales",
3 col = colours, ylim=c(0,3000))
4 legend("topright", goods, fill = colours,ncol=2)
```

Namiesto farieb môžeme vyplniť stĺpce textúrami. Najjednoduchšie sú rovnobežné čiary. Hustotu čiar môžeme zmeniť nastavením `density`, ktorého hodnota je vektor s dĺžkou rovnou počtu čiar. Podobne môžeme nastavením argumentu `angle` ako vektora s dĺžkou rovnajúcou sa počtu pruhov určiť uhol vyplňajúcich čiar.

```
1 x<-c(2000,2400,1400,2600)
2 barplot(x,density=c(5,10,20,30), angle=c(0,30,60,90),
3 col="blue",names.arg=goods,main="Sale volumes",
4 xlab="Fruit",ylab="Sales")
```

```
1 angle1<-c(0,30,60,90)
2 angle2<-c(90,120,150,0)
3 barplot(x,density=c(10,15,20,25), angle=angle1,beside = TRUE,
4 main="Sale volumes", col = colours,names.arg=months,
5 xlab = "Month",ylab = "Sales",ylim=c(0,3000))
6 barplot(x,density=c(10,15,20,25), angle=angle2,beside = TRUE,
7 col = colours, add=TRUE)
```

```

8 legend("topright", goods, ncol=2, fill=colours, angle=angle1,
9 density=c(10,15,20,25))
10 legend("topright", goods, ncol=2, fill=colours, angle=angle2,
11 density=c(10,15,20,25))

```

Histogramy

Histogram je znázornenie približného rozdelenia číselných údajov. Zobrazujú frekvencie hodnôt premennej rozdelené do intervalov. Histogram je podobný stĺpcovému grafu, ale s tým rozdielom, že agreguje hodnoty do súvislých intervalov. Histogramy poskytujú približný obraz o hustote základného rozdelenia údajov. Histogram možno vytvoriť v R pomocou funkcie `hist()`.

```
barplot(H,xlab,ylab,title, names.arg,col)
```

Parametre použité vo funkcii sú nasledovné:

- `data` je vektor obsahujúci číselné hodnoty použité v histograme,
- `main` označuje názov grafu,
- `col` sa používa na nastavenie farby stĺpcov,
- `border` sa používa na nastavenie farby okraja každého stĺpca,
- `xlab` sa používa na popis osi x,
- `xlim` sa používa na zadanie rozsahu hodnôt na osi x,
- `ylim` sa používa na určenie rozsahu hodnôt na osi y,
- `breaks` sa používa na uvedenie šírky jednotlivých stĺpcov.

Príklad.

Ilustrujme vykreslenie histogramov na prípade hodu kockou. Predpokladajme, že budeme hádzať dvoma kockami 10 000-krát a zaujíma nás súčet padnutých bodov. Najprv simulujeme hod kockou:

```

1 dice1<-sample(1:6,replace=T,10000)
2 dice2<-sample(1:6,replace=T,10000)
3 c<-dice1+dice2

```

Teraz môžeme vykresliť histogram súčtov pomocou funkcie `hist()`:

```

1 hist(c,breaks=1.5:12.5, main="Rolling 2 dice",
2 xlab="two dice", ylab="Frequency")

```

Koláčové grafy

Koláčový graf je graf pre jednu kategoriálnu premennú a je alternatívou stĺpcového grafu. Koláčový graf (alebo kruhový graf) je diagram v tvare kruhu, ktorý je rozdelený na výseky zodpovedajúcich pomerom zobrazovaných hodnôt. V koláčovom grafe je dĺžka oblúka každého výseku (a teda aj jeho stredový uhol a plocha) úmerná veličine, ktorú predstavuje. Základná syntax pre vytvorenie koláčového grafu v prostredí R je:

```
pie(data, labels, radius, main, col, clockwise)
```

Význam argumentov:

- `data` je vektor obsahujúci číselné hodnoty použité v koláčovom grafe,
- `labels` sa používa na popis výsekov,
- `radius` označuje polomer kruhu koláčového grafu (hodnota medzi -1 a $+1$),
- `main` označuje názov grafu,
- `col` označuje paletu farieb,
- `clockwise` je logická hodnota, ktorá udáva, či sa výseky kreslia v smere alebo proti smeru hodinových ručičiek.

Príklad.

Predpokladajme, že chceme znázorniť podiely mesačných výdavkov domácnosti pomocou koláčového grafu. Berieme do úvahy tieto kategórie výdavkov: bývanie, potraviny, oblečenie, zábava a ostatné. Hodnoty, ktoré použijeme ako parametre koláčového grafu:

```
1 data<-c(200,300,100,80,150)
2 labels<-c("housing","food","clothing","entertainment","other")
3 pie(data,labels,main="Monthly expenses")
```

Na zmenu farieb použijeme `rainbow()`, ktorá definuje paletu farieb.

- `n` počet farieb (≥ 1), ktoré majú byť v palete,
- `s`, v „sýtosť“ a „hodnota“, ktoré sa majú použiť na doplnenie popisu ku farbám
- `start` (opravený) odtieň v $\langle 0; 1 \rangle$, pri ktorom začína zvolená dúha,
- `end` (opravený) odtieň v $\langle 0; 1 \rangle$, pri ktorom dúha končí,
- `gamma` korekcia gama pre každú farbu, (r, g, b) v priestore RGB (so všetkými hodnotami v $\langle 0; 1 \rangle$), výsledná farba zodpovedá $(r^\gamma, g^\gamma, b^\gamma)$,
- `alpha` priehľadnosť, číslo v $\langle 0; 1 \rangle$, (0) znamená priehľadný a 1 opak.

```

1 description<-paste(labels , "\n", data , sep=" ")
2 pie(data , description , main="Monthly expenses" ,
3 col=rainbow(length(data)))

```

Zmenili sme aj popisky. K ich názvom sme pridali aj číselné hodnoty.

Ako ďalšie vylepšenia môžeme požadovať popisy s percentami a zobrazenie grafu s 3D efektom. Najprv musíme prepočítať percentá a pridať výsledky do popisov. Aby sme získali percentá v celých číslach, použijeme funkciu `trunc()`. Potom môžeme vytvoriť koláčový graf, tentoraz pomocou palety `heat.colors()`.

```

1 description<-paste(labels , "\n", trunc(100*data/sum(data)) ,
2 "%", sep=" ")
3 pie(data , description , main="Monthly expenses" ,
4 col=heat.colors(length(data)))

```

Na získanie 3D efektu v grafe musíme použiť balíček `plotrix`. Používame grafy `pie3D()` s 3D efektom.

```

1 library("plotrix")
2 pie3D(data , labels=description , main="Monthly expenses" ,
3 col=rainbow(length(data)))

```

Vzhľad 3D grafu môžeme ďalej prispôsobiť pomocou parametrov:

- `height`, ktorý udáva výšku 3D koláča (predvolená hodnota je 0,1).
- `theta`, ktorý mení uhol pohľadu (predvolený uhol je $\frac{\pi}{6}$).
- `explode`, ktorý definuje oddelenie častí koláča.

```

1 pie3D(data , labels=description , main="Monthly expenses" ,
2 col=terrain.colors(length(data)) , height=0.2 , theta=1.5 ,
3 explode=0.1)

```

Užitočnou alternatívou ku koláčovým grafom je `fun.plot()` definovaný v balíku `plotrix`. Umožňuje vizuálne porovnať koláčové sektory grafu. Vejárový graf môžeme prispôsobiť nastavením ďalších argumentov:

- `max.span` uhol maximálneho sektora v radiánoch. Predvolené nastavenie je škálovanie `data` tak, aby sa súčet rovnal 2π .
- `ticks` počet políčok, ktoré by sa objavili, keby boli sektory na koláčovom grafe. Predvolené nastavenie je žiadne políčka.

Ilustrácia vejárového grafu

```

1 fan.plot(data , labels=description , main="Monthly expenses" ,
2 col=rainbow(length(data)) , max.span=pi , ticks=max(data))

```


Nevýhodou vejárového grafu je veľké biele miesto nad grafom. Tento priestor môžeme odstrániť nastavením nového grafického zariadenia s výškou a šírkou definovanou používateľom. Nové grafické okno otvoríme pomocou funkcie `new.dev()`. Veľkosť okna definujeme pomocou argumentov `height` a `width`.

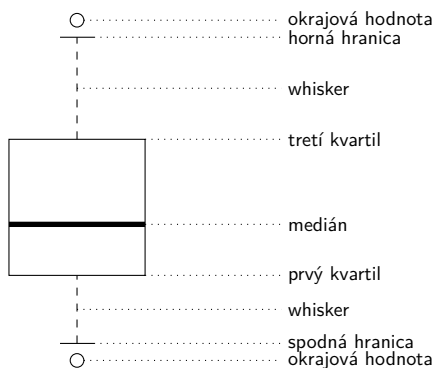
```
1 dev.new(width=10,height=5,unit="cm")
2 fan.plot(data,labels=description,main="Monthly expenses",
3 col=rainbow(length(data)),max.span=pi,ticks=max(data))
```

Krabicový graf

Krabicové grafy vizualizujú rozdelenie číselných údajov a šikmost prostredníctvom zobrazenia kvartilov (alebo percentilov) a priemerov údajov. Boxplot sa skladá z dvoch častí, boxu a sady „whiskerov“. Box je zobrazuje rozsah od dolného kvartilu po horný kvartil s vodorovnou čiarou nakreslenou vo vnútri boxu, ktorá označuje medián. Whiskery môžu alternatívne znázorňovať niekoľko okrajových hodnôt spomedzi pozorovaných údajov:

- minimum a maximum všetkých údajov,
- jedna štandardná odchýlka nad a pod priemerom údajov,
- 9. percentil a 91. percentil,
- 2. percentil a 98. percentil.

Všetky údaje, ktoré nie sú zahrnuté medzi whiskery, by sa mali vykresliť ako odlahlé hodnoty vyznačené bodkou, malým krúžkom alebo hviezdíčkou, občas sa to však nerobí.



Boxploty sa v R vytvárajú pomocou funkcie `boxplot()`.

```
boxplot(x, data, notch, varwidth, names, main)
```

- `x` je vektor alebo vzorec,
- `data` je data frame.
- `notch` je logická hodnota. Nastavte ako `TRUE`, ak chcete nakresliť vrub.
- `varwidth` je logická hodnota. Nastavte ako `TRUE`, aby sa vykreslila šírka rámečka úmerná veľkosti vzorky,
- `names` sú označenia skupín, ktoré sa vypíšu pod každým boxplotom,
- `main` sa používa na zadanie nadpisu grafu.

Príklad.

Predpokladajme, že v dátovom súbore `players.csv` máme štatistické údaje z basketbalového zápasu. Tento dátový súbor obsahuje identifikáciu hráčov, ich pozíciu, počet streleckých pokusov a úspešné strelecké pokusy. Pomocou boxplotu porovnajme dosiahnuté body podľa pozície.

```
1 players<-read.csv("players.csv")
2 boxplot(made~position,data=players,
3 xlab="Position",ylab="Points_gained",
4 main="Scoring_by_position")
```

Podobne ako pri ostatných typoch grafov môžeme upraviť vzhľad grafu. Ilustrujeme vyfarbenie grafu a nastavením hodnoty `varwidth=TRUE` upravíme šírku boxov tak, aby bola úmerná veľkosti vzorky.

```
1 boxplot(made~position,data=players,
2 xlab="Position",ylab="Points_gained",
3 main="Scoring_by_position",col="cyan",varwidth=TRUE)
```

Nastavením logickej premennej `horizontal` na hodnotu `TRUE` môžeme otáčať grafy v boxplote. Okrem toho sa farby môžu v jednotlivých boxoch líšiť.

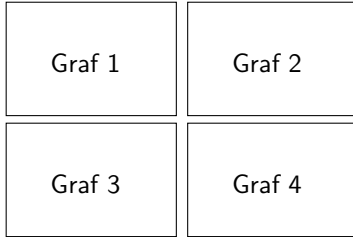
```
1 boxplot(made~position,data=players,
2 xlab="Position",ylab="Points_gained",
3 main="Scoring_by_position",
4 col="col=c("blue","cyan","green"),
5 varwidth=TRUE,horizontal=TRUE)
```

Kombinácie grafov

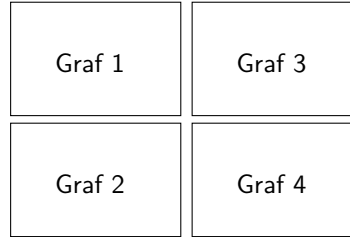
V prostredí R môžeme graf kombinovať s grafickými parametrami `mfrow` a `mfcol`. Stačí zadať vektor, ktorý určuje počet riadkov a počet stĺpcov, ktoré plánujeme vytvoriť.

- `mfrow` budú grafy usporiadané podľa riadkov,
- `mfc` budú grafy usporiadané podľa stĺpcov.

`mfrow=c(2,2):`



`mcol=c(2,2):`



```

1 set.seed(5)
2 x <- rexp(80)
3 # Dva riadky, dva stlpce
4 par(mfrow = c(2, 2))
5 # Grafy
6 hist(x, main = "Histogram") # Vľavo hore
7 boxplot(x, main = "Boxplot") # Vpravo hore
8 plot(x, main = "Scatterplot") # Vľavo dole
9 pie(table(round(x)), main = "Piegraph") # Vpravo dole
10 # Spät na puvodne graficke zariadenie
11 par(mfrow = c(1, 1))

```

Často potrebujeme vytvoriť obrázok so zložitejšou štruktúrou. Vtedy používame `layout()`, ktorá má štyri dôležité argumenty:

- `mat` matica, kde každá hodnota predstavuje umiestnenie obrázkov.
- `widths` vektor pre šírky stĺpcov, v cm môžeme použiť `lcm()`.
- `heights` vektor pre výšku stĺpcov, v cm môžeme použiť `lcm()`.
- `respect` logická hodnota alebo matica vyplnená 0 a 1 rovnakých rozmerov ako `mat` na určenie, či sa majú rešpektovať vzťahy medzi šírkami a výškami alebo nie.

```

1 l <- layout(matrix(c(1, 2, 2, # Prvy, druhy,
2                    3, 3, 4), # tretí a štvrtý graf
3                nrow = 2,
4                ncol = 3,
5                byrow = TRUE))
6 layout.show(l)

```

Pred pridaním grafov môžeme zobrazíť náhľad ich rozloženia pomocou `layout.show()`. Túto metódu ilustrujeme na bodovom grafe realizovanom s okrajmi vo forme histogramu a krabicového grafu.

```
1  l <- layout(matrix(c(2, 0, 1, 3),
2                    nrow = 2, ncol = 2,
3                    byrow = TRUE),
4                    widths = c(9, 3),
5                    heights = c(3, 9), respect = TRUE)
6  plot(x, main = "Scatterplot")
7  hist(x, main = "Histogram")
8  boxplot(x, main = "Boxplot")
```

Literatúra

- [1] BRAUN, W.,J. & MURDOCH, D.,J.:*A First Course in Statistical Programming with R*, 2nd edition, New York, Cambridge University Press, 2016.
- [2] CHANG, W.:*R Graphics Cookbook.*, Sebastopol, United States, O'Reilly Media, Inc, 2013.
- [3] CHIU, Y.:*R for Data Science Cookbook*, Birmingham, Packt Publishing Ltd., 2016.
- [4] CRAWLEY, M.,J.:*The R Book.*, 3rd edition, Chichester, England, John Wiley&Sons, Ltd., 2022.
- [5] CRAWLEY, M.J.:*Statistics: An Introduction Using R*. Boston, Addison-Wesley Publishing company, 2015.
- [6] DALGAARD, P.:*Introductory Statistics with R*, 2nd edition, New York, Springer, 2008.
- [7] DEGROOT, M.,H. & SCHERVISH, M.,J.:*Probability and Statistics*, 4th edition, Boston, Addison-Wesley Publishing company, 2012.
- [8] EVERIT, B.,S., & HOTHORN, T.: *A Handbook of Statistical Analysis Using R*, Boca Raton, Chapman&Hall, CRC Press, Taylor & Francis Group, 2010.
- [9] QUICK, J.,M.:*Statistical Analysis with R*, Birmingham, Packt Publishing Ltd., 2010.
- [10] SCHUMACKER, R. & TOMEK, S.:*Understanding Statistics Using R*, New York, Springer, 2013.
- [11] SPECTOR, P.:*Data Manipulation with R*, New York, Springer, 2008.

- [12] SUHOV, Y. & KELBERT, M.: *Probability and Statistics by Example. Volume I. basic Probability and Statistics*, New York, Cambridge University Press, 2005.
- [13] WICKHAM, H., & GROLEMUND, G.: *R for Data Science*, Sebastopol, United States, O'Reilly Media, Inc., 2017.

Aleš Kozubík, Department of Mathematical methods and Operations Research, Faculty of Management Sciences and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic, e-mail: alesko@frcatel.fri.uniza.sk

Project: Innovative Open Source Courses for Computer Science Curriculum



This material was created as one of the activity of the project “Innovative Open Source Courses for Computer Science Curriculum”, funded by the Erasmus+ grant nr. 2019-1-PL01-KA203-065564.

2nd International Conference on Open Source tools in Computer Science
university education
Conference Proceedings
Rudolf Blaško (Ed.)
University of Žilina
Publisher University of Žilina
Print EDIS-vydavateľstvo UNIZA
Graphic editing and typesetting: Rudolf Blaško
First edition 2022
Number of pages: 88
Number of prints: 60
ISBN 978-80-554-1941-1

ISBN 978-80-554-1941-1



9 788055 419411