

PROGRAMOVANIE V JAZYKU R
stručné poznámky a zbierka príkladov

Lenka Filová, KAMŠ, FMFI UK

March 20, 2022

Contents

1 Úvod	2
1.1 Inštalácia	2
1.2 Menu a help	2
1.3 Práca s objektami	2
1.4 Zásady písania vlastných funkcií	3
1.5 Inštalácia prídavných balíkov	3
1.6 Prostredie RStudio	3
2 Základné matematické operácie	5
3 Práca s dátovými súborami	8
4 Riadenie toku	10
5 Funkcie	11
6 Práca s maticami a vektormi	13
7 Optimalizácia	15
8 Rozdelenia pravdepodobnosti	16
9 Vizualizácia dát a grafika	18
10 R Markdown	19
10.1 Základné príkazy	19
10.2 Pridávanie R a LaTeX kódu	20

Chapter 1

Úvod

1.1 Inštalácia

Web stránka a základné prostredie:

www.r-project.org

IDE prostredie:

<https://www.rstudio.com/products/RStudio/>

1.2 Menu a help

- `?solve` alebo `help(function)` otvorí dokumentáciu k danej funkcii.
- `apropos("solve")` nájde všetky funkcie obsahujúce dané slovo.
- `example(solve)` spustí príklady k funkcii `solve`.
- `help.search("solve")` použijeme, ak nepoznáme presný názov funkcie: prehľadáva `help` alebo `??function`.
- `RSiteSearch('regression', restrict='functions', matchesPerPage=100)`
- `help.start()` spustí lokálne html prostredie pre `help`.
- `sessionInfo()` vypíše informácie o aktuálnej verzii R a nainštalovaných balíčkoch.

1.3 Práca s objektami

- `ls()` vypíše všetky objekty, ktoré sa pri ukončení R uložia do súboru `'.RData'`.
- `rm(obj1, obj2, ...)` slúži na odstraňovanie objektov.
`rm(list = ls())` vymaže všetky objekty (bez ďalšieho varovania!).
- `class(object)` vypíše typ objektu.
- `summary(object)` je generická funkcia pre súhrnné informácie o rôznych typoch objektov.
- `gc()` (garbage collection) niekedy užitočné na uvoľnenie pamäte po vymazaní veľkých objektov.
- `.Last.value` vypíše hodnotu posledného vyčíslovaného výrazu.

- `dir()` vypíše obsah aktuálneho adresára.
- `getwd()` vypíše aktuálny adresár.
- `setwd("/home/user")` zmena aktuálneho adresára.
- `q()` ukončenie R.

1.4 Zásady písania vlastných funkcií

1. Názvy súborov by mali byť zmysluplné a mať príponu `.R`.
2. Názvy objektov by nemali byť rovnaké ako názvy už existujúcich funkcií v R.
3. Pred a za operátormi (`=`, `+`, `-`, `<-`, etc.) treba dať medzeru.
4. Ľavá kučeravá zátvorka sa nepíše na nový riadok, ale za ňou nasleduje nový riadok. Naopak, pravú kučeravú zátvorku píšeme na nový riadok a nič už za ňou nenasleduje (výnimkou je `else`). Príkazy vnútri kučeravých zátvoriek treba odsadiť.
5. Jeden riadok by nemal mať viac ako 80 znakov.
6. Na odsadenie príkazov používajte dve medzery.
7. Na priradenie používajte `<-`, nie `=`.
8. Nezabúdajte do svojich programov napísať dostatočné množstvo komentárov.

1.5 Inštalácia prídavných balíkov

R je rozšíriteľné pomocou balíkov (packages), z ktorých mnohé naprogramovali a dali verejne k dispozícii samotní používatelia R. Ich kompletný zoznam je na adrese: <https://cran.r-project.org/web/packages/>.

Každý takýto balík má svoju stránku, na ktorej sú základné informácie o ňom, manuály a zdrojové kódy. Napríklad balík `TeachingDemos` nájdeme na adrese:

<http://cran.r-project.org/web/packages/TeachingDemos/index.html>

Samotné balíky môžeme inštalovať priamo z prostredia R. Postup je nasledovný: V záložke "PACKAGES" -> `install package(s)` -> 1.krok zvolit CRAN, 2. krok zvolit balík alebo:

```
install.packages("TeachingDemos")
library() # vypise vsetky dostupne nainstalovane balicky
library(TeachingDemos) # nacita konkretny balicek
library(help=TeachingDemos) # vypise vsetky funkcie a objekty balicka
search() # vypise ktore balicky su momentalne nacistane
faces(rbind(1:3,5:3,3:5,5:7))
faces(matrix(sample(1:1000,128,),16,8))
```

1.6 Prostredie RStudio

RStudio nám dáva možnosť mať spolu pokoje všetky súbory spojené s projektom: vstupné dáta, skripty, výsledky, obrázky. Nový projekt môžeme vytvoriť cez *Click File > New Project*. Po vytvorení projektu sa R nachádza v jeho adresári (skontrolujte to cez `getwd()`) a sem bude ukladať všetky výstupy a snažiť sa odtiaľ otvoriť súbory, ktoré

nebudú mať napísanú kompletnú cestu. Keď reštartujete RStudio, s adresári projektu sa bude nachádzať súbor `.Rproj`. Jeho otvorením dostaneme RStudio do stavu, v ktorom bolo pred zatvorením: dostaneme sa k objektom, histórii príkazov a sme v rovnakom adresári.

Chapter 2

Základné matematické operácie

```
# +, -, *, /, ^
# sqrt(), abs(), choose(n, k)
# sin(), cos(), tan(), asin(), acos(), atan()
# exp()
# log(number, base=number)
# log10(number) - dekadicky
e
pi
1:10
diag(3)
# round(number, digits = pocet.miest)
round(pi, 2)
round(0.5 + (-2:4))
```

```
    Základná syntax
    object <- function(arguments)
```

```
var <- 12
var
# alebo
( var <- 12 )
var2 <- "a"
var3 <- "slovo"
"a" -> var3
var*var
var*var2
```

Na názvy premenných sa môžu používať písmená, číslice, bodka, podčiarkovník. Nesmie sa začínať číslom. R je case sensitive.

```
v <- c( 2 , 3.1 , 5.4 )
v
v+5
sqrt(v)
```

```
A <- matrix( c(1,2,3,4,4,5,5,6,1), ncol=3, nrow=3, byrow=TRUE )
```

```
dim(A) # rozmery matice
length(B) # pocet prvkov matice
```

```
t(A) # transpozícia matice
A %% A # násobenie matic
A * A # násobenie po prvkoch
solve(A) # inverzia matice
A %% solve(A)
```

Príklad 2.1. Vytvorte vektory

- $(0.1^{10} \cdot 0.2^{25}, 0.1^{20} \cdot 0.2^{24}, \dots, 0.1^{25} \cdot 0.2^1)$,
- $(2, \frac{2^2}{2}, \frac{2^3}{3}, \dots, \frac{2^{25}}{25})$,
- vektor s hodnotami $e^x \cos(x)$ pre $x = 4, 4.2, \dots, 8$.

Príklad 2.2. Vypočítajte

- $\sum_{i=1}^{300} (i^2 + 2/i)$
- $\sum_{i=1}^{100} (i^3 + 4 \sin(x))$
- $\sum_{i=1}^{50} (\frac{2^i}{i^2} + \frac{3^i}{i^3})$
- $(\sum_{i=1}^{100} i)^2 - \sum_{i=1}^{100} i^2$
- $1 + \frac{2}{3} + \frac{2}{3} \frac{4}{5} + \frac{2}{3} \frac{4}{5} \frac{6}{7} + \dots + \frac{2}{3} \frac{4}{5} \dots \frac{40}{41}$.

Príklad 2.3. Vytvorte maticu M s 12 rovnakými riadkami $(1, -3, 4, 2)$. Vypočítajte $M'M$. Vypočítajte súčet najväčšej a najmenšej vlastnej hodnoty $M'M$.

Príklad 2.4. Uvažujme vektor $1:n$, kde n je prirodzené číslo. Napíšte príkaz, ktorý určí, koľko prvkov vektora je deliteľných tromi.

Príklad 2.5. Pomocou funkcií `rep` a `:` vytvorte vektory

- $(1, 1, 1, 2, 2, 2, \dots, 10, 10, 10)$
- $(1, 2, \dots, 10, 1, 2, \dots, 10, 1, 2, \dots, 10)$
- $(4, 6, 3, 4, 6, 3, \dots, 4, 6, 3)$, kde sa trojica $4, 6, 3$ zopakuje 10-krát.
- $(4, 4, \dots, 4, 6, 6, \dots, 6, 3, 3, \dots, 3)$, kde sa 4 zopakuje 10-krát, 6 sa zopakuje 15-krát a 3 sa zopakuje 8-krát

Príklad 2.6. Pomocou funkcie `sample` vytvorte dva vektory $x = (x_1, \dots, x_n)$ a $y = (y_1, \dots, y_n)$ dĺžky $n = 100$, ktorých prvkami budú náhodné prirodzené čísla z $(0, 1, 2, \dots, 999)$ (napríklad ako `sample(0:999, 100, replace=FALSE)`).

- Vytvorte vektor $(y_2 - x_1, y_3 - x_2, \dots, y_n - x_{n-1})$.
- Vytvorte vektor

$$\left(\frac{\sin y_1}{\cos x_2}, \dots, \frac{\sin y_{n-1}}{\cos x_n} \right).$$

- Vytvorte vektor $(2x_1 + x_2x_3, 2x_2 + x_3x_4, \dots, 2x_{n-2} + x_{n-1}x_n)$.
- Vypíšte všetky prvky y , ktoré sú väčšie ako 500.
- Vypíšte indexy všetkých prvkov y , ktoré sú väčšie ako 500.
- Vypíšte tie prvky x , ktoré zodpovedajú prvkom y väčším ako 500.

- Nájdite najväčší prvok y a vypíšte všetky prvky y , ktoré sa od neho líšia najviac o 100.
- Vypíšte indexy všetkých párnych prvkov x .
- Zoradte prvky vektora x tak, aby jeho indexy zodpovedali zoradeniu vektora y od najmenšieho prvku po najväčší.

Chapter 3

Práca s dátovými súbormi

Príklad 3.1. Nech `x <- c(NA, 3, 33, 6, 7, 9, NA, 23, NA, 11, NA, 22, NA, NA, 12, 4)`. Napíšte príkaz, ktorý

- Spočíta, koľko chýbajúcich hodnôt je vo vektore.
- Vrátí indexy zodpovedajúce chýbajúcim hodnotám v `x`.
- Všetky chýbajúce hodnoty nahradí číslom 0.
- Vymaže všetky chýbajúce hodnoty.
- Vrátí logický vektor, ktorý bude mať prvky s hodnotou `TRUE` na mieste chýbajúcich hodnôt a inde `FALSE`.

Príklad 3.2. Pomocou príkazu `scan` načítajte súbor `ako`

- vektor dĺžky 20
- maticu s 20 riadkami
- maticu s 10 riadkami a 2 stĺpcami

Príklad 3.3. Vytvorte jednoduchý dátový rámec typu 10×4 . Usporiadajte jeho položky podľa veľkosti prvej premennej.

Príklad 3.4. Načítajte do premennej `seishu` dáta

<http://www.iam.fmph.uniba.sk/ospm/Filova/vsa/Seishu.txt>, ktoré popisujú výsledky testov japonského vína.

1. Vytvorte dátový rámec `seishu2`, ktorý bude obsahovať len premenné `y1` a `x1`.
2. Zo `seishu2` vymažte všetky riadky, v ktorých premenná `y1` nadobúda zápornú hodnotu
3. Určte koeficienty priamky, ktorá vznikne fitovaním lineárneho modelu závislosti `y1` od `x1` v `seishu2` metódou najmenších štvorcov.

Príklad 3.5. Načítajte dáta `VADeaths` (dostupné priamo v R).

- Presvedčte sa, či je `VADeaths` typu `data.frame` a ak nie, skonvertujte ho.
- Vytvorte novú premennú `sucet` a zapíšte do nej súčty jednotlivých riadkov súboru.
- Zmeňte poradie premenných tak, aby `sucet` bol prvý stĺpec.

- Usporiadajte riadky zostupne podľa hodnoty premennej `sučet`.

Príklad 3.6. Načítajte dáta `swiss` (dostupné priamo v R).

- Vypíšte tie riadky, pre ktoré je hodnota premennej `Fertility` vyššia ako 70.
- Usporiadajte súbor podľa hodnôt premennej `Agriculture`.
- Vypíšte tie riadky, pre ktoré je hodnota premennej `Infant.Mortality` vyššia ako 20 a zároveň hodnota premennej `Catholic` nižšia ako 10.
- Určte koeficienty priamky, ktorá vznikne fitovaním lineárneho modelu závislosti `Infant.Mortality` od `Education` metódou najmenších štvorcov.
- Vytvorte premennú `Relig`, ktorá bude nadobúdať hodnotu 0, ak `Catholic` je menej ako 10, hodnotu 1, ak `Catholic` je medzi 10 a 30 a hodnotu 2 pre `Catholic` nad 30.

Chapter 4

Riadenie toku

Príklad 4.1. Vypočítajte $1^2 + 2^2 + 3^2 + \dots + 400^2$.

Príklad 4.2. Vypočítajte $1 * 2 + 2 * 3 + 3 * 4 + \dots + 249 * 250$.

Príklad 4.3. Bez použitia funkcie `factorial` vypočítajte $10!$.

Príklad 4.4. Napíšte skript, ktorý vypočíta $1 + 1/1! + 1/2! + 1/3! + \dots + 1/10!$ a po každom pripočítaní vypíše priebežný súčet.

Príklad 4.5. Nájdite súčet párnych členov Fibonacciho postupnosti, ktoré sú menšie ako 4 milióny.

Príklad 4.6. Pomocou cyklu vytvorte nasledovnú maticu typu 5×5 :

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

Príklad 4.7. Nájdite súčet všetkých násobkov 3 a 5, ktoré sú menšie ako 1000.

Príklad 4.8. Napíšte program, ktorý vypočíta $1 + 2 + \dots + 300$ a po každom dvadsiatom sčítaní vypíše medzisúčet.

Príklad 4.9. Vypočítajte

•

$$\sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^3}{2+3j}$$

•

$$\sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^3}{2+3ij}$$

•

$$\sum_{i=1}^{20} \sum_{j=1}^i \frac{i^3}{2+3ij}$$

Príklad 4.10. Vytvorte maticu M typu 10×10 , pre ktorú $M_{i,i} = 3$, $M_{i+1,i} = M_{i,i+1} = 1$, $i = 1, \dots, 10$ a všetky ostatné prvky sú nuly.

Chapter 5

Funkcie

Príklad 5.1. Napíšte funkciu, ktorá na vstupe načíta tri čísla a vypíše, či môžu byť stranami trojuholníka.

Príklad 5.2. Napíšte funkciu, ktorá načíta štyri čísla a,b,c,d, ktoré predstavujú dva uzavreté intervaly [a,b] a [c,d]. Program vypíše tú z nasledujúcich viet, ktorá pre ne platí:

Intervaly su disjunktne.

Jeden z intervalov je podmnožinou druhého.

Intervaly sa pretínajú.

Príklad 5.3. Napíšte program, ktorého vstupom budú koeficienty kvadratickej rovnice $ax^2 + bx + c = 0$ a výstupom reálne korene (dva alebo jeden) alebo string "rovnica nema realne korene".

Príklad 5.4. Napíšte program, ktorého vstupom bude prirodzené číslo n a ktorý na vypočíta súčet $2 + 4 + \dots + n$, ak n je párne alebo $2 + 4 + \dots + (n - 1)$, ak n je nepárne.

Príklad 5.5. Napíšte funkciu, ktorá vypočíta súčin abc pre také $a < b < c \in \mathbb{N}$, pre ktoré $a^2 + b^2 = c^2$ a $a + b + c = 1000$.

Príklad 5.6. Pre prirodzené čísla definujme nasledovnú postupnosť:

$n \rightarrow n/2$, ak n je párne

$n \rightarrow 3n + 1$, ak n je nepárne

Teda napríklad, ak by prvým členom postupnosti bolo číslo 13, postupnosť by vyzerala nasledovne:

13 40 20 10 5 16 8 4 2 1

Napíšte program, ktorý zistí, pre ktoré prirodzené číslo menšie ako N, má takáto postupnosť najväčšiu dĺžku.

Príklad 5.7. Napíšte funkciu pre výpočet sumy

$$f(n) = \sum_{r=1}^n \sum_{s=1}^r \frac{s^2}{10 + 4r^3}.$$

Príklad 5.8. Napíšte funkciu, ktorá určí, či sú dve vstupné množiny A, B (dané reálnymi vektormi) disjunktne. Ak nie sú disjunktne, vypíše ich prienik.

Príklad 5.9. Napíšte funkciu, ktorá pre daný vektor x vráti všetky prvky x, ktoré sú väčšie ako $\text{mean}(x)$.

Príklad 5.10. Napíšte funkciu, ktorá pre daný vektor $x \in \mathbb{R}^n$ a funkciu $f : \mathbb{R} \rightarrow \mathbb{R}$, vypočíta $\sum_{i=1}^n i \cdot f(x_i)$.

Príklad 5.11. Napíšte funkciu, ktorá určí, či sú dve vstupné slová anagramy. Anagram je slovo, ktoré z iného slova vznikne iným usporiadaním písmen.

Príklad 5.12. Napíšte funkciu, ktorá pre vektor x a reálne číslo k nájde taký prvok x , ktorý je najbližšie ku k .

Príklad 5.13. Vytvorte funkciu, ktorá zo vstupného vektora vymaže všetky duplicitné prvky.

Príklad 5.14. Vytvorte funkciu, ktorá vráti TRUE, ak sa reálne číslo na vstupe nachádza vo vstupnom vektore x .

Chapter 6

Práca s maticami a vektormi

Príklad 6.1. Vytvorte maticu M_1 , ktorej i, j -ty prvok je minimom z indexov i, j a maticu M_2 , ktorej i, j -ty prvok je maximom z indexov i, j .

Príklad 6.2. Vyriešte sústavu rovníc

$$3x + 2y - z = 1 \quad (6.1)$$

$$2x - 2y + 4z = -2 \quad (6.2)$$

$$-x + 0.5y - z = 0. \quad (6.3)$$

Príklad 6.3. Napíšte funkciu, ktorá vráti počet NA vo vektore.

Príklad 6.4. Napíšte funkciu, ktorá pre dva vektory x, y a index k vráti počet prvkov y , ktoré sú menšie ako x_k .

Príklad 6.5. Napíšte funkciu, ktorej vstupom budú čísla $k \in \mathbb{R}$ a $n \in \mathbb{N}$ a výstupom matica typu $n \times n$, ktorá ma na diagonále k a mimo diagonály 1.

Príklad 6.6. Napíšte funkciu, ktorej vstupom bude vektor x a prirodzené číslo d a výstupom matica, ktorá bude mať v i -tom stĺpci vektor i -tych mocnín x , kde $i = 1, \dots, d$.
Například pre vstup `x <- c(1,2,3)`, `d=3` dostaneme výstup

```
1 1 1
2 4 8
3 9 27
```

Príklad 6.7. Napíšte funkciu, ktorej vstupom bude vektor $x = (x_1, \dots, x_n)$ a výstupom vektor $((x_1 + x_2 + x_3)^2/1, (x_2 + x_3 + x_4)^2/2, \dots, (x_{n-2} + x_{n-1} + x_n)^2/(n-2))$.

Príklad 6.8. Napíšte funkciu, ktorej vstupom bude n -prvkový vektor a výstupom zvyšok ciferného súčtu prvkov tohto vektora po delení tromi.

Například pre vektor $(0,1,2,7)$ je požadovaný výstup 1 a pre vektor $(2,3,4)$ je výstup 0.

Príklad 6.9. Napíšte funkciu, ktorej vstupom bude n -prvkový vektor pozostávajúci z čísel 0,1,2 a výstupom trojprvkový vektor udávajúci počty týchto čísel.

Teda například pre vstupný vektor $(0,0,1,2,2,1,0,1,1,1)$ dostaneme výsledok $(3,5,2)$.

Príklad 6.10. Napíšte funkciu, ktorej vstupom bude n -prvkový vektor a výstupom logická hodnota (T/F) v závislosti od toho, či je daný vektor palindróm.

Za palindróm považujeme reťazec, ktorý sa "rovnako číta" odpredu aj odzadu, například `abcba` alebo `abccba` sú palindrómy, ale `abcba` nie.

Príklad 6.11. Napíšte funkciu, ktorej vstupom bude vektor pozostávajúci z núl a jednotiek a výstupom číslo, ktoré udáva počet "sérií" v tomto vektore. Pod sériou rozumieme postupnosť rovnakých čísel, pred a za ktorou je iné číslo.

Teda napríklad vektor $(0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1)$ má 6 sérií.

Príklad 6.12. Napíšte funkciu, ktorej vstupom bude vektor $x = (x_1, \dots, x_n)$ a výstupom matica F , ktorej i -ty riadok je $(x_i, x_i^3, \ln(1 + x_i))$. Vypíšte všetky nepárne riadky matice F , ktorá je výstupom tejto funkcie pre vstupný vektor x , ktorý obsahuje diskretizáciu intervalu $[0, 10]$ na 21 rovnakých častí.

Príklad 6.13. Napíšte funkciu, ktorej vstupom je matica s celočíselnými prvkami. Výstupom funkcie bude matica tých istých rozmerov s tým, že každé nepárne číslo bude nahradené svojim dvojnásobkom.

Príklad 6.14. Napíšte funkciu, ktorá pre vstupnú štvorcovú maticu M určí, či je

- symetrická
- pozitívne semidefinitná
- ortogonálna

Príklad 6.15. Vytvorte maticu typu 7×4 , ktorej stĺpce budú ľubovoľné štyri rôzne permutácie prvkov vektora $(2, 6, 8, 12, 21, 34, 49)$.

Príklad 6.16. Napíšte funkciu, ktorá pre maticu M vráti vektor, ktorého prvky tvoria diagonálu matice M^2 .

Príklad 6.17. Napíšte funkciu, ktorá pre vstupnú maticu M

- vypíše rozdiel medzi najmenším a najväčším prvkom
- vypíše riadkové a stĺpcové indexy najväčšieho a najmenšieho prvku
- vzájomne vymení najväčší a najmenší prvok

Príklad 6.18. Napíšte funkciu, ktorá pre vstupnú maticu M vypočíta Kroneckerov súčin matice

$$\begin{pmatrix} 1 & 5 & 0 \\ 2 & 4 & 3 \end{pmatrix}$$

s maticou M , teda

$$\begin{pmatrix} 1 & 5 & 0 \\ 2 & 4 & 3 \end{pmatrix} \otimes M = \begin{pmatrix} M & 5M & 0 \\ 2M & 4M & 3M \end{pmatrix}$$

Príklad 6.19. Načítajte dáta `swiss` (dostupné priamo v R). Pomocou funkcie `apply` vypíšte riadkové a stĺpcové priemery.

Chapter 7

Optimalizácia

Príklad 7.1. Nájdite minimum funkcie $\sin^2(x) + x/10$ na intervale $(0, 2\pi)$.

Príklad 7.2. Nájdite maximum funkcie $2\sqrt{x} - x^2 - x$ na intervale $[0, 2]$.

Príklad 7.3. Nájdite minimum funkcie $f(x_1, x_2) = (x_1^2 + x_2^2 - 11)^2 + (x_1 + x_2^2 - 7)$. Za počítačové body voľte postupne body $(-4, 4)$, $(2, -2)$ a $(2, 2)$.

Príklad 7.4. Nájdite korene rovnice $x^2 + x = 1$.

Príklad 7.5. Spoločnosť musí zakúpiť riešenie na redukciu emisií oxidu siričitého o 2 milióny jednotiek a oxidu uhličitého o 3 milióny jednotiek. Prvé riešenie stojí 5 eur na jednotku znečistenia pre obe zlúčeniny. Druhé riešenie zníži emisie oxidu siričitého rovnako ako prvé, ale emisie oxidu uhličitého zníži dvojnásobne; jednotková cena je 8 eur. Aká kombinácia oboch riešení splní požiadavky na emisie pri minimálnych nákladoch?

Chapter 8

Rozdelenia pravdepodobnosti

Príklad 8.1. Výrobca vyrába súčiastky, ktorých hmotnosť má normálne rozdelenie so strednou hodnotou 17.46g a variáciou 375.67g. Aká je pravdepodobnosť, že náhodne vybraná súčiastka váži viac ako 19g?

Príklad 8.2. Nech súčiastky z predchádzajúcej úlohy majú chybovosť 0.005. Dodávajú sa v baleniach po 25 kusov. Aká je pravdepodobnosť, že náhodne vybrané balenie obsahuje práve jednu chybnú súčiastku?

Príklad 8.3. Testujeme vedľajšie účinky lieku nasledovným spôsobom: prvému pacientovi dáme najnižšiu dávku, akú máme k dispozícii. Ak mal pacient i nejaké vedľajšie účinky, pacientovi $i+1$ dávku znížime (ak môžeme). Ak pacient i nemal žiadne vedľajšie účinky, rozhodujeme sa podľa výsledku hodu mincou: ak padne hlava, pacient $i+1$ dostane rovnakú dávku ako pacient i , ak padne znak, pacientovi $i+1$ dávku zvýšime (ak môžeme).

Napíšte program, ktorého vstupom bude vektor pravdepodobností vedľajších účinkov p pre jednotlivé dávky, pravdepodobnosť padnutia hlavy na minci b a počet pacientov n a ktorého výstupom bude vektor dĺžky n s dávkami lieku pridelenými jednotlivým pacientom. Predpokladáme, že máme k dispozícii dávky $(1, 2, \dots, k)$.

Príklad 8.4. Máme n nezávislých javov, ktoré nastávajú s pravdepodobnosťami (p_1, \dots, p_n) . Napíšte funkciu, ktorej zadáme vektor (p_1, \dots, p_n) a ktorá vypočíta pravdepodobnosť, že nastane práve jeden z týchto javov.

Príklad 8.5. Máme dve urny. V prvej je 10 modrých a 8 žltých guľičiek, v druhej je 6 modrých a 6 žltých. Vytiahneme guľičku z prvej urny a dáme ju do druhej. Potom vytiahneme náhodne guľičku z druhej urny. Simulačne určte pravdepodobnosť, že táto guľička je modrá.

Príklad 8.6. Vygenerujte vektor x , ktorý predstavuje logaritmus náhodného výberu rozsahu 1000 zo štandardizovaného normálneho rozdelenia. Pomocou funkcie `ifelse` nahraďte všetky chýbajúce hodnoty hodnotou 9999.

Príklad 8.7. Nech pravdepodobnosť prenosu choroby z komára na človeka je pri jednom uštipnutí $1/500$. Napíšte funkciu, ktorá vypočíta pravdepodobnosť nákazy po uštipnutí n komármi.

Príklad 8.8. Napíšte funkciu, ktorej vstupom bude prirodzené číslo n a ktorá vykoná nasledovné operácie:

1. vygeneruje n čísel $x = (x_1, \dots, x_n)$ zo štandardizovaného normálneho rozdelenia
2. vypočíta ich aritmetický priemer \bar{x}

3. ak $\bar{x} > 0$, vygeneruje n čísel $y = (y_1, \dots, y_n)$ z exponenciálneho rozdelenia s parametrom \bar{x}
ak $\bar{x} \leq 0$, vygeneruje n čísel $z = (z_1, \dots, z_n)$ z exponenciálneho rozdelenia s parametrom $-\bar{x}$ a položí $y = -z$
4. vypočíta hodnotu k , ktorá udáva počet tých i , pre ktoré je $|y_i| > |x_i|$
5. vráti x , y a k .

Chapter 9

Vizualizácia dát a grafika

Príklad 9.1. • Načítajte dáta `cars` a nakreslite ich scatterplot.

- Pomocou funkcie `points` pridajte do grafu nové pozorovania v bodoch (20,58) a (19,60) červenou farbou.
- Pomocou funkcie `lines` pridajte do grafu regresnú priamku modrou farbou.
- Do grafu pridajte popis, ktorý rozlíši pôvodne čierne a nové červené pozorovania.
- Pomocou `par` a `mfrow` vytvorte vedľa seba dva histogramy, jeden pre ľavý a druhý pre pravý stĺpec dát.

Príklad 9.2. Vygenerujte diskretizáciu intervalu $[-50,50]$ na 1000 častí a nakreslite grafy funkcií $\ln(x)$, $\exp(x)$, $\sin(2x)$, $\sin(x)/\cos(x)$.

Príklad 9.3. Vytvorte maticu typu 20×20 , ktorej prvky budú náhodné čísla z intervalu $[0, 1]$. Vykreslite ju pomocou funkcie `image`.

Chapter 10

R Markdown

R Markdown je nástroj na tvorenie komplexných výstupov v rôznych formátoch, napríklad pdf, wordoské súbory, prezentácie a podobne. Keďže spája R s množstvom externých nástrojov, help nie je v tomto prípade dostupný cez ?. Súbor R Markdown je textový súbor s príponou .Rmd, ktorý má obvykle nasledovný obsah:

- Hlavička, ktorá má na začiatku a na konci `--`
- Kusy Rkovského kódu, ktoré majú na začiatku a na konci `““`
- Text

Príklad jednoduchého R Markdown dokumentu

10.1 Základné príkazy

Zdôraznenie

```
*italic*   **bold**
```

```
_italic_   __bold__
```

Hlavičky

```
# Header 1
```

```
## Header 2
```

```
### Header 3
```

Zoznamy

Neočíslované zoznamy:

```
* Item 1
* Item 2
+ Item 2a
+ Item 2b
```

Očíslované zoznamy:

1. Item 1
2. Item 2
3. Item 3
- + Item 3a
- + Item 3b

Linky

`http://example.com`

`[linked phrase](http://example.com)`

Obrázky zo siete alebo lokálne súbory v domovskom adresári:

``

`![optional caption text](figures/img.png)`

10.2 Pridávanie R a LaTeX kódu

Rkovské príkazy, ktoré chceme vykonať a zobrazíť:

```
““{r}
summary(cars$dist)
summary(cars$speed)
```

Rkovské príkazy vnútri textu:

There were `'r nrow(cars)'` cars studied

Grafy:

```
““{r}
plot(cars, pch = 18)
““
```

Bloky kódu, ktoré chceme zobrazíť, ale nie vykonať:

```
““
This text is displayed verbatim / preformatted
““
```

LaTeXrovnice vnútri textu:

`$equation$`

Rovnice na samostatnom riadku:

`$$ equation $$`

Bibliography

- [1] Gillespie, Lovelace. *Efficient R programming*. O'Reilly Media, Incorporated, 2016.
- [2] Wickham, Golemund. *R for data science: import, tidy, transform, visualize, and model data*. "O'Reilly Media, Inc.", 2016.