

# M4130 cvičení 12

## Programování v R

4. 12. 2019

1. Vytvořte funkci `znamenko`, která pro zadané číslo `x` vypíše (funkce `cat` nebo `print`) text, zda je číslo záporné, kladné, nebo rovno nule, a která na svůj výstup uloží hodnoty  $-1$ ,  $1$ , resp.  $0$ .

```
znamenko <- function(x) {  
  if (x < 0) {  
    print("zaporne")  
    return(-1)  
  }  
  if (x > 0) {  
    print("kladne")  
    return(1)  
  }  
  print("nula")  
  return(0)  
}  
  
z <- znamenko(-2)  
z
```

2. Vytvořte funkci `vypocty`, která pro vstupní číselný vektor `v` spočítá a na svůj výstup vrátí kumulované součty a kumulované součiny čísel ve vektoru `v`.

```
vypocty <- function(v) {  
  a <- cumsum(v)  
  b <- cumprod(v)  
  return(list(soucty = a, souciny = b))  
}  
  
x <- c(1, 2, 3, -10)  
vypocty(x)  
y <- vypocty(x)  
y$soucty  
y$souciny
```

3. Vytvořte funkci `je.cele`, která pro zadané číslo `x` vrátí logickou hodnotu `TRUE/FALSE` podle toho, jestli je zadané číslo celé, nebo není. Proporovnaní vyzkoušejte, jak funguje funkce `is.integer`.

4. Vytvořte funkci `den`, která pro zadné číslo `a` (přirozené od 1 do 7) dne vrátí název dne v týdnu:  $1 \Rightarrow$  pondělí, ...,  $7 \Rightarrow$  pátek. Pro jiná čísla `a` vypíše chybové hlášení nebo vraťte hodnotu `NA`. Funkci vytvořte ve verzi s využitím indexování ve vektoru, a ve verzi s využitím funkce `switch`. Pro test podmínky můžete využít funkci `je.cele` nebo funkci `any`.

5. Pro každé přirozené číslo `n` od 0 do 10 spočítejte  $2^n$ , přímo zapsáním matematického výrazu a poté pomocí funkce `sapply`.

```
n <- seq(0, 10, by = 1)
2^n

moje.funkce <- function(x) {
  z <- 2^x
  return(z)
}
sapply(n, moje.funkce)
```

6. Do vektoru  $x$  uložte posloupnost přirozených čísel od  $-10$  do  $25$ . Pomocí funkce `sapply` pro každé číslo  $x$  z vektoru  $x$  spočítejte:

- $\sqrt{x-5}$  pro  $x \geq 5$ ,
- $|x|$  pro  $x \leq -5$ ,
- $0$  pro ostatní  $x$ .

7. Vygenerujte matici  $A$  řádu  $5$  s náhodnými celočíselnými hodnotami z intervalu  $[-5; 5]$ . Pomocí funkce `apply` a vhodných vlastních funkcí spočítejte:

- minimum v každém řádku (sloupci) matice  $A$ ,
- maximum v každém řádku (sloupci) matice  $A$ ,
- součet absolutních hodnot čísel v každém řádku (sloupci) matice  $A$ ,
- rozdíl maximálního a minimálního čísla v každém řádku (sloupci) matice  $A$ .

8. Vytvořte funkci `Fib.cislo`, která pro zadané  $n$ ,  $n \in \mathbb{N}_0$ , spočítá a na svůj výstup vrátí  $n$ -tý člen Fibonacciho posloupnosti,  $F_n$ . Vyzkoušejte verzi s~rekurzí, s `for`-cyklem a s `while`-cyklem.

```
Fib.cislo <- function(n) {
  if (!je.cele(n) | n < 0) {return(NA)}
  if (n == 0) {return(0)}
  if (n == 1) {return(1)}
  return(Fib.cislo(n - 1) + Fib.cislo(n - 2))
}
```

```
Fib.cislo2 <- function(n) {
  if (!je.cele(n) | n < 0) {return(NA)}
  if (n == 0) {return(0)}
  if (n == 1) {return(1)}
  a <- 0
  b <- 1
  for (i in seq(2, n, by = 1)) {
    c <- a + b
    a <- b
    b <- c
  }
  return(b)
}
```

```
Fib.cislo3 <- function(n) {
  if (!je.cele(n) | n < 0) {return(NA)}
  if (n == 0) {return(0)}
  if (n == 1) {return(1)}
  while (n > 1) {
    c <- a + b
    a <- b
    b <- c
    n <- n - 1
  }
  return(b)
}
```

```

if (n == 0) {return(0)}
if (n == 1) {return(1)}
a <- 0
b <- 1
i <- 1
while (i < n) {
  i <- i + 1
  c <- a + b
  a <- b
  b <- c
}
return(b)
}

```

9. Vytvořte funkci `Fib.posl`, která pro zadané přirozené číslo `n` spočítá a na svůj výstup vrátí Fibonacciho posloupnost až do  $n$ -tého Fibonacciho čísla. Vyzkoušejte verzi s využitím funkce `Fib.cislo`, s `for`-cyklem a s `while`-cyklem.

10. Pracujte s datovým souborem `seniori.csv` z minulých cvičení. Pomocí funkce `tapply` spočítejte průměrnou hodnotu BMI zvlášť pro muže a pro ženy. Analogicky pak spočítejte mediány a rozdíly mezi největší a nejmenší hodnotou BMI.

```

X <- read.csv("seniori.csv", sep = ";", dec = ",")
X$pohlavi <- factor(X$pohlavi, labels = c("zena", "muz"))
str(X)
tapply(X$BMI, X$pohlavi, mean)

```

```

fn3 <- function(x) {
  return(max(x) - min(x))
}
tapply(X$BMI, X$pohlavi, median)
tapply(X$BMI, X$pohlavi, fn3)

```

11. (*Metoda nejmenších čtverců*) Pracujte s datovým souborem `vejce.Rdata`. Načtěte data ze souboru. Pomocí metody nejmenších čtverců spočítejte koeficienty tzv. regresní přímky, popisující závislost velikosti vajec kukačky v závislosti na velikosti vajec v hníždě hostitele. Vykreslete graf, v němž budou pomocí bodů vyznačené hodnoty z datového souboru a přikreslená regresní přímka. Vypište rovnici regresní přímky a spočítejte reziduální součet čtverců  $S_e$ .

Pro výpočty koeficientů regresní přímky si naprogramujte funkci `mnc`, jejímž vstupem budou dva číselné vektory  $\mathbf{x}$  a  $\mathbf{y}$ , a výstupem bude vektor koeficientů  $\boldsymbol{\beta}$ . Výpočty koeficientů  $\boldsymbol{\beta}$  ve funkci `mnc` řešte jako řešení soustavy lineárních rovnic v maticovém tvaru  $\mathbf{A}\boldsymbol{\beta} = \mathbf{b}$  s vhodnou maticí soustavy  $\mathbf{A}$  a pravou stranou  $\mathbf{b}$ .

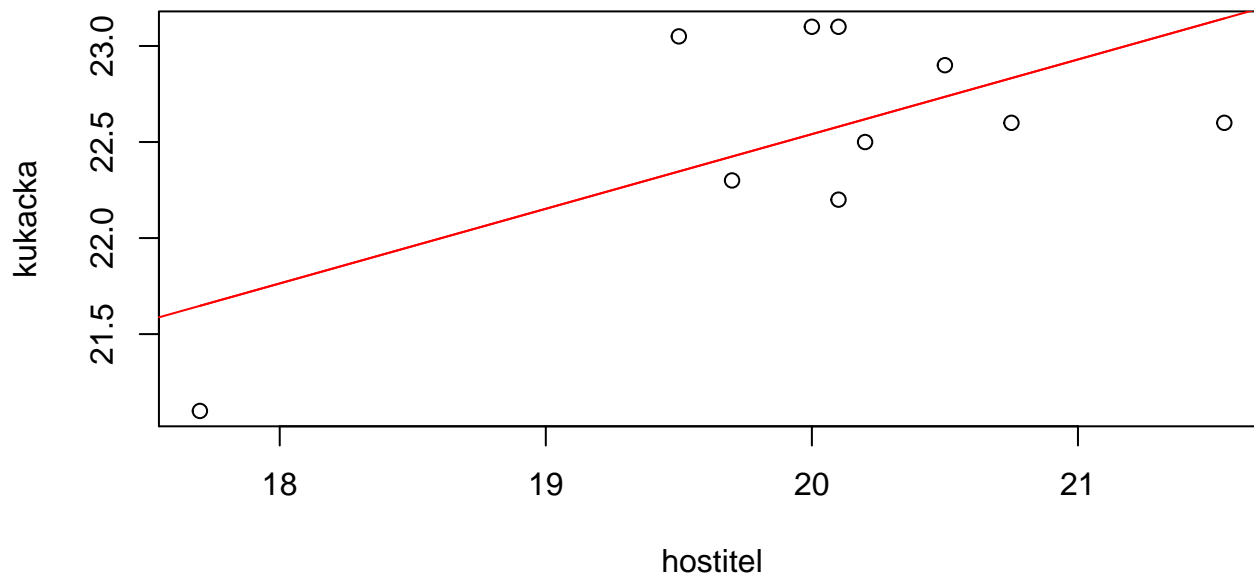
Regresní přímka má rovnici  $y = \beta_0 + \beta_1 x$ , kde koeficienty  $\beta_0, \beta_1$  získáme tzv. metodou nejmenších čtverců, kdy hledáme minimum funkce dvou proměnných

$$S(\beta_0, \beta_1) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \rightarrow \min.$$

Dvojice  $(x_i, y_i)$ ,  $i = 1, \dots, n$ , jsou hodnoty vstupní a výstupní veličiny pro celkem  $n$  měření. Lze odvodit (cvičení z matematické analýzy), že minimum nastává pro  $\boldsymbol{\beta} = (\beta_0, \beta_1)'$  řešící systém rovnic

$$\beta_0 n + \beta_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i, \quad \beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i.$$

Reziduální součet čtverců  $S_e$  je hodnota funkce  $S(\beta_0, \beta_1)$  pro spočítané hodnoty koeficientů  $\beta_0, \beta_1$ .



Rovnice regresni primky:  $y = 14.768 + 0.389 x$

Rezidualni soucet ctvercu:  $S_e = 1.927$