

# Práce s maticemi v R

**Matice** (*matrix*) je dvojrozměrným polem, které obsahuje prvky stejného datového typu. Všechny sloupce matice mají shodnou délku a také všechny řádky matice mají shodnou délku.

## Vytvoření a rozšíření matice

Matice je v matematice schéma nějakých matematických objektů – prvků matice. Můžeme si tedy vytvořit matici písmen:

$$C = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

Jdeme na to. Matice se vytvoří příkazem `matrix()`. Jako první parametr vektor obsahující všechny prvky matice. Parametr `nrow` udává počet řádků matice, parametr `ncol` udává počet sloupců matice. V základu se matice vytvoří ze zadaného vektoru po sloupcích. Pokud chceme mít písmena v řádcích, musíme dodat ještě parametr `byrow` s logickou hodnotou `TRUE`.

```
# matice se vytváří příkazem matrix()
# vestavěný vektor "letters" obsahuje malá písmena anglické abecedy
# 1:12 je zkratka pro c(1, 2, 3, ..., 11, 12)
# letters[1:12] tedy vypíše prvních dvanáct písmen abecedy
C <- matrix(letters[1:12], nrow = 3, byrow = TRUE)

# vypíšeme výsledek
C
#>      [,1] [,2] [,3] [,4]
#> [1,] "a"  "b"  "c"  "d"
#> [2,] "e"  "f"  "g"  "h"
#> [3,] "i"  "j"  "k"  "l"

# je to matice?
is.matrix(C)
#> [1] TRUE

# jaké prvky obsahuje?
mode(C)
#> [1] "character"

# příkaz str(...) nám dokáže stručně vypsát proměnnou
str(C)
#> chr [1:3, 1:4] "a" "e" "i" "b" "f" "j" "c" "g" "k" "d" "h" "l"

# je to tedy matice 3 řádky x 4 sloupce a po sloupcích obsahuje písmena v uvedeném pořadí

# jaké jsou rozměry matice C?
dim(C)
#> [1] 3 4

# je to vlastně vektor c(3, 4), který můžeme využít ve výpočtech
dim(C) == c(3, 4)
#> [1] TRUE TRUE

# dvojnásobně potvrzená pravda!
```

Přidávání či ubírání sloupců a řádků se provádí příkazy `rbind()` (pro řádky, *r* jako *row*) a `cbind` (pro sloupce, *c* jako *column*).

Pokud chceme řádky nebo sloupce odebrat, použijeme negativní index (případně vektor s negativními indexy).

```
# přidání řádku c("x", "x", "x", "x") k matici C
rbind(C, c("x", "x", "x", "x"))
#>      [,1] [,2] [,3] [,4]
#> [1,] "a"  "b"  "c"  "d"
#> [2,] "e"  "f"  "g"  "h"
#> [3,] "i"  "j"  "k"  "l"
#> [4,] "x"  "x"  "x"  "x"

# přidání sloupce c("x", "x", "x") k matici C
cbind(C, c("x", "x", "x"))
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,] "a"  "b"  "c"  "d"  "x"
#> [2,] "e"  "f"  "g"  "h"  "x"
#> [3,] "i"  "j"  "k"  "l"  "x"

# DŮLEŽITÉ: pokud výsledek příkazu nepřihodíte nějaké proměnné,
# pak budou nové řádky nebo sloupce jen v aktuálním výstupu do konzole,
# pokud chcete výstup uložit do matice C, je nutné řádek začít třeba:
# C <- rbind(...)

C[-1, ] # odebrání 1. řádku matici C
C[, -2] # odebrání 2. sloupce matici C
C[-1:-2, ] # odebrání prvních dvou řádků matice C
```

# Pojmenování řádků a sloupců matice

Někdy se hodí sloupce a řádky v matici pojmenovat. Pak se na jednotlivé prvky můžeme odkazovat mnohem pohodlněji. Jmenovky sloupcům přiřazuje příkaz `colnames()`, řádkům pak příkaz `rownames()`. Jmenovky můžeme odstranit příkazem `unname()`.

```
# přidáme jmenovky sloupcům
colnames(C) <- c("c1", "c2", "c3", "c4")

# přidáme jmenovky řádkům
rownames(C) <- c("r1", "r2", "r3")

# smažeme všechny jmenovky: tektokrát musíme výsledek přiřadit zpátky do proměnné
C <- unname(C)

# jiným způsobem založíme nové jmenovky příkazem dimnames(),
# kam můžeme napsat řádky i sloupce v jednom seznamu vektorů :-)
dimnames(C) <- list( c("r1", "r2", "r3"), c("c1", "c2", "c3", "c4") )

# na prvek 2. řádku 3. sloupce se můžeme odkázat jmenovkou
C["r2", "c3"]
```

## Adresace a výpis prvků

Jak se odkážeme na nějaký konkrétní prvek matice, případně na konkrétní oblast matice? Je to podobné jako u vektorů: většinu práce zvládnou hranaté závorky za jménem matice.

```
# zopakujeme si naši matici i se jmenovkami
C <- matrix(letters[1:12], nrow = 3, byrow = T, dimnames = list( c("r1", "r2", "r3"), c("c1", "c2", "c3", "c4")))

# zobrazíme prvek 2. řádku, 3. sloupce
C[2, 3]
#> [1] "g"

# zobrazíme prvek 2. řádku, 3. sloupce pomocí jmenovek
C["r2", "c3"]
#> [1] "g"

# zobrazíme celý první řádek
C[1, ]
#> c1 c2 c3 c4
#> "a" "b" "c" "d"

# je to tedy vektor c("a", "b", "c", "d") s popisky

# zobrazíme celý třetí sloupec
C[, 3]

# zobrazíme submatici sestávající z 1. a 3. řádku a 2. a 4. sloupce s popisky
C[c(1, 3), c(2, 4)]

# chceme zobrazit prvek vpravo dole a nevíme, jak je matice velká?
# příkaz dim(...) zjistí rozměry matice a vrátí vektor c(řádky, sloupce)
# na návratovou hodnotu se pak můžeme odkazovat při hledání prvku vpravo dole:
C[dim(C)[1], dim(C)[2]]

# chceme pátý prvek v pořadí, jdeme po sloupcích
C[5]
#> [1] "f"

# chceme osmý a devátý prvek v pořadí (po sloupcích)
C[c(8, 9)]
#> [1] "g" "k"

C[13] # NA
diag(C) # c("a", "f", "k"); hlavní diagonála
diag(C[, dim(C)[2]:1]) # c("d", "g", "j"); vedlejší diagonála
```

## Příklady

- Matice - příklad č.1
- Matice - příklad č.2

## Matematické operace: Maticová algebra

Kdo chce v R pracovat s maticemi, zřejmě už má nějakou představu o maticové algebře. Někaké informace na toto téma jsou např. ZDE (<https://is.cuni.cz/webapps/zzp/detail/91026/26957317/>).

Tady si uvedeme jen, jak některé prvky maticové algebry zapíšeme v R.

- Hadamardův součin matic A a B: `A * B`
- Maticový součin matic A a B: `A %*% B`
- Transpozice matice A ( $A^T$ ): `t(A)`
- Jednotková matice  $I_2$ : `diag(2)`
- Inverzní matice  $A^{-1}$ : `solve(A)`
- Vlastní čísla matice A: `eigen(A)$values`
- Vlastní vektory matice A: `eigen(A)$vectors` *# R vrací ortonormální vlastní vektory*

