

Úvod do R - Matice

Vít Syrovátka

Matrice – Vektor s rozměry

- vektor uspořádaný do řádků a sloupečků, tvoří tabulku
- vektor rozšířený o atribut `$dim` – dimensions, rozměry
 - přiřazením rozměrů vektoru vznikne matice
 - odebrání rozměrů matici dá vznik vektoru
- shodně s vektory
 - může obsahovat jen hodnoty stejného typu



matice není vhodná pro ukládání dat

- hodí se pro různé výpočty, maticová algebra, většinou ale matici moc nevyužijeme

```
> x <- sample(20, 6)
> x
[1] 6 17 19 9 12 5

> dim(x) <- c(2, 3)
> x
      [,1] [,2] [,3]
[1,]    6   19   12
[2,]   17    9    5

> dim(x) <- NULL
> x
[1] 6 17 19 9 12 5
```

Jak vytvořit matici

1) přiřazením rozměrů vektoru pomocí

```
dim(x) <- c(a,b)
```

2) spojením vektorů (nebo matic) do sloupečků či řádků pomocí `cbind()` a

```
rbind()
```

3) poskládáním pomocí `matrix()`

- `nrow`= počet řádků
- `ncol`= počet sloupců
- `byrow`= FALSE skládat po řádcích?
- `dimnames` jména řádků a sloupců
- `matrix()` replikuje `x` na požadovaný rozměr

- defaultně se hodnoty skládají **po sloupcích**

```
x <- sample(20, 6)
> x
[1] 6 8 11 16 4 14
```

```
> dim(x) <- c(2, 3)
```

```
> x
      [,1] [,2] [,3]
[1,]    6   11    4
[2,]    8   16   14
```

```
> cbind(1:3, 2:4)
```

```
      [,1] [,2]
[1,]    1    2
[2,]    2    3
[3,]    3    4
```

```
> rbind(1:3, 2:4)
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
```

Jak vytvořit matici

1) přiřazením rozměrů vektoru pomocí

```
dim(x) <- c(a,b)
```

2) spojením vektorů (nebo matic) do sloupečků či řádků pomocí `cbind()` a `rbind()`

3) poskládáním pomocí `matrix()`

- `nrow`= počet řádků
- `ncol`= počet sloupců
- `byrow`= FALSE skládat po řádcích?
- `dimnames` jména řádků a sloupců
- `matrix()` replikuje `x` na požadovaný rozměr

- defaultně se hodnoty skládají **po sloupcích**

```
> x
[1] 6 8 11 16 4 14
```

```
> matrix(x, nrow = 2)
      [,1] [,2] [,3]
[1,]    6   11    4
[2,]    8   16   14
```

```
> matrix(x, ncol = 3)
      [,1] [,2] [,3]
[1,]    6   11    4
[2,]    8   16   14
```

```
> matrix(x, nrow = 4, ncol = 3)
      [,1] [,2] [,3]
[1,]    6    4   11
[2,]    8   14   16
[3,]   11    6    4
[4,]   16    8   14
```

```
> matrix(x, nrow = 2, byrow = TRUE)
      [,1] [,2] [,3]
[1,]    6    8   11
[2,]   16    4   14
```

Pojmenování rozměrů matice

1) při tvorbě matice funkcí `matrix()` - arg.

```
dimnames= list(rownames,  
               colnames)
```

2) u již existující matice zadáním:

- `rownames()` <- `c()`
- `colnames()` <- `c()`
- `dimnames()` <- `list(c(),
 c())`

```
> dimnames(x) <- list(c("1", "2"),  
                     c("A", "B", "C"))
```

```
> x  
  A  B  C  
1 6 11 4  
2 8 16 14
```

```
> x<- matrix(x, nrow = 2,  
            dimnames = list(c("A", "B"),  
                            c("i", "j", "k")))
```

```
> x  
  i  j  k  
A 6 11 4  
B 8 16 14
```

```
> rownames(x) <- c("radek 1", "radek 2")  
> colnames(x) <- c("prvni", "druhy", "treti")
```

```
> x  
      prvni  druhy  trei  
radek 1     6    11    4  
radek 2     8    16   14
```

```
> dimnames(x) <- list(NULL,  
                     c("A", "B", "C"))
```

```
> x  
      A  B  C  
[1,] 6 11 4  
[2,] 8 16 14
```

Podvýběry v maticích

- 1) `x[n]` – n-tý element
- 2) `x[i, j]` – element na i-tém řádku a j-tém sloupci
- 3) `x[i,]` – celý i-tý řádek
- 4) `x[, j]` – celý j-tý sloupec
 - `n` – numerický nebo logický subscript
 - `i, j` – numerický, logický i jmenný subscript
 - `drop = FALSE` zajistí zachování složitosti objektu – výsledkem bude opět matice, i když bude vybrána jen např. jedna hodnota

```
> x
  i   j   k
A 6  11  4
B 8  16 14

> x[3:5]
[1] 11 16  4

> x[1, 2:3]
  j   k
11  4

> x[, 2:3]
  j   k
A 11  4
B 16 14

> x[1:2, 2]
  A   B
11 16

> x[1:2, 2, drop = F]
  j
A 11
B 16
```

Podvýběry v maticích

- 1) `x[n]` – n-tý element
- 2) `x[i, j]` – element na i-tém řádku a j-tém sloupci
- 3) `x[i,]` – celý i-tý řádek
- 4) `x[, j]` – celý j-tý sloupec
 - `n` – numerický nebo logický subscript
 - `i, j` – numerický, logický i jmenný subscript
 - `drop = FALSE` zajistí zachování složitosti objektu – výsledkem bude opět matice, i když bude vybrána jen např. jedna hodnota

```
> x
  i  j  k
A 6 11  4
B 8 16 14

> x["A" , c("i", "k")]
i k
6 4

> x[, colSums(x) > 20, drop = F]
  j
A 11
B 16

> x[x > 10]
[1] 11 16 14
```

List - “seznam”

- vektor objektů - komponentů
- nejsložitější objekt
- může kombinovat komponenty s různými hodnotami
- listy mohou být vnořené – např. list listů listů
- tvorba stejná jako tvorba vektorů, jen místo `c()` použijeme `list()`
- pojmenování komponentů listu pomocí `names()`
- podvýběry:
 - 1) `[]` – vybere danou část listu
 - 2) `[[]]` – vytáhne komponent v dané části listu
 - 3) `$` – vytáhne daný komponent
- subscript numerický, logický, jmenný

```
> list(1:5, c("a", "b", "c"), TRUE)
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "a" "b" "c"

[[3]]
[1] TRUE

> list(cisla = 1:5,
      pismenka = c("a", "b", "c"),
      anoano = TRUE)
$cisla
[1] 1 2 3 4 5

$pismenka
[1] "a" "b" "c"

$anoano
[1] TRUE
```


List - "seznam"

- vektor objektů - komponentů
- nejsložitější objekt
- může kombinovat komponenty s různými hodnotami
- listy mohou být vnořené – např. list listů listů
- tvorba stejná jako tvorba vektorů, jen místo `c()` použijeme `list()`
- pojmenování komponentů listu pomocí `names()`
- podvýběry:
 - 1) `[]` – vybere danou část listu
 - 2) `[[]]` – vytáhne komponent v dané části listu
 - 3) `$` – zkratka pro `[[]]`
- subscript numerický, logický, jmenný

```
> x
$cisla
[1] 1 2 3 4 5

$pismenka
[1] "a" "b" "c"

$anoano
[1] TRUE

> x["cisla"]
$cisla
[1] 1 2 3 4 5

> x[["cisla"]]
[1] 1 2 3 4 5

> x$cisla
[1] 1 2 3 4 5
```



x



x[1]

x[2]

x[[2]]



x[[1]][[1]]