

ANALÝZA DAT V R

Datové typy, datové struktury (vektory, faktory)

1 Datové typy

1. numerický (numerical)

```
> a <- -2.514
```

2. znakový (character)

– skládá se z řetězců, zadává se do "" nebo '', pro textové hodnoty

```
> b <- "univerzita"
```

3. logický (logical)

– jen hodnoty TRUE (nebo T), FALSE (nebo F)

```
> a > 0
```

```
[1] FALSE
```

4. komplexní (complex)

– numerická hodnota doplněná o imaginární jednotku i ($i^2 = -1$)

```
> d <- 6-2i
```

– `mode` zjistí typ objektu

```
> mode(b)
```

```
[1] "character"
```

– speciální hodnoty: **NA** (z angl. Not Available, chybějící nebo neznámá hodnota), **Inf** (z angl. Infinity, nekonečno), **NaN** (z angl. Not a Number, výsledek nedefinovaných numerických výpočtů), **NULL** (v objektu není nic uloženo)

2 Datové struktury

– umožňují více hodnot specifikovat jako jeden objekt

1. vektory

2. faktory

3. matice

4. pole
5. datová tabulka
6. seznam

2.1 Vektory

- `c()` vytvoří vektor (`c` od angl. combine), kde do závorky zapíšeme složky vektoru oddělené čárkami

```
> v <- c(-5, 1, 0.6, 17, 1124)
> v
[1] -5.0 1.0 0.6 17.0 1124.0
```

- lze tvořit numerické, logické, komplexní i textové vektory
- `length()` určí délku vektoru, který je uvedený v závorce

```
> length(v)
[1] 5
```

- matematické operátory a funkce aplikované na vektory se provádějí po složkách, např.

```
> v/2          (tj. každá složka je vydělena 2)
[1] -2.5 0.5 0.3 8.5 562.0
> v^5          (tj. každá složka je umocněna na 5.)
[1] -3.125000e+03 1.000000e+00 7.776000e-02 1.419857e+06 1.794038e+15
```

```
> w <- c(1, 0.5, 2, 3, 12)
> w
```

```
[1] 1.0 0.5 2.0 3.0 12.0
```

```
> v/w          (tj. 1. složka vektoru v je vydělena 1. složkou vektoru w, 2. složka vektoru v je vydělena 2. složkou vektoru w, atd.)
```

```
[1] -5.000000 2.000000 0.300000 5.666667 93.666667
```

```
> v*w          (tj. 1. složka vektoru v je vynásobena 1. složkou vektoru w, 1. složka vektoru v je vynásobena 1. složkou vektoru w, atd.)
```

```
[1] -5.0 0.5 1.2 51.0 13488.0
```

- pokud bychom chtěli skalární součin vektorů, použijeme místo obyčejného znaménka pro součin (tj. `*`) následující `%*%`

```
> v%*%w
```

```
[,1]
```

```
[1,] 13535.7
```

- `a:b` vytvoří aritmetickou posloupnost od `a` do `b` s krokem 1

```
> (z <- 1:6)
```

```
[1] 1 2 3 4 5 6
```

- `seq(from=a, to=b, by=k)` vytvoří aritmetickou posloupnost od `a` do `b` s krokem `k`

```
> seq(from=1,to=6,by=1)
```

```
[1] 1 2 3 4 5 6
```

```
> seq(from=2,to=10,by=2)
```

```
[1] 2 4 6 8 10
```

```
> seq(from=1,to=10,by=2.4)
```

```
[1] 1.0 3.4 5.8 8.2
```

```
> seq(from=-5,to=1,by=1)
```

```
[1] -5 -4 -3 -2 -1 0 1
```

- `rep(o, n)` vytvoří vektor, ve kterém se zadaný objekt `o` opakuje `n`-krát

```
> rep(1.5,3)
```

```
[1] 1.5 1.5 1.5
```

- chceme-li vypsat i -tou složku vektoru, použijeme název vektoru a do hranaté závorky uvedeme danou pozici, tj. i (podobně pro skupinu složek)

```
> v <- c(-5, 1, 0.6, 17, 1124)
```

```
> v[3]
```

```
[1] 0.6
```

```
> v[2:4]
```

```
[1] 1 0.6 17
```

```
> v[c(1,4,5)]
```

```
[1] -5 17 1124
```

- chceme-li vymazat i -tou složku vektoru, použijeme název vektoru a do hranaté závorky uvedeme danou pozici opatřenou znaménkem minus, tj. $-i$ (podobně pro skupinu složek)

```
> v <- v[-4]
```

```
[1] -5 1 0.6 1124
```

```
> v <- v[-c(1,2)]
```

```
[1] 1 0.6 1124.0
```

2.2 Faktory

- speciální případ vektorů pro práci s nominálními nebo ordinálními daty
- navíc obsahují informaci `Levels` (konečná množina hodnot, kterých proměnná může nabývat)

```
> factor(c("zena","zena","muz","zena","muz","zena"))
```

```
[1] zena zena muz zena muz zena
```

```
Levels: muz zena
```

- argument `labels` lze použít k definici popisků, pokud jsou hodnoty vyjádřeny číselně

```
> factor(c(1,1,0,1,0,1),labels=c("muz","zena"))
```

```
[1] zena zena muz zena muz zena
```

```
Levels: muz zena
```

- na faktory nelze aplikovat matematické operátory a elementární funkce

SPECIÁLNÍ FUNKCE	
<code>sum(·)</code>	součet prvků
<code>cumsum(·)</code>	kumulativní součty prvků
<code>prod(·)</code>	součin prvků
<code>cumprod(·)</code>	kumulativní součiny prvků
<code>sort(·)</code>	uspořádání prvků do neklesající posloupnosti
<code>rank(·)</code>	pořadí prvků v neklesající posloupnosti
<code>which(·)</code>	vrátí pozici prvku, který vyhovuje dané podmínce
<code>min(·)</code>	minimální prvek
<code>max(·)</code>	maximální prvek

Příklad 1

Definujte vektory $u = (2, -1, 3, 0)$, $v = (4, -1, 5, -2)$ a vypočítejte:

(a) $a = u + v$ $[a = (6, -2, 8, -2)]$

(b) $b = 2u$ $[b = (4, -2, 6, 0)]$

(c) $c = u - v$ $[c = (-2, 0, -2, 2)]$

(d) $d = -4u - \frac{1}{2}v$ $[d = (-10, \frac{9}{2}, -\frac{29}{2}, 1)]$

(e) $e = uv$ $[e = 24]$

(f) $f = \sum_{i=1}^4 u_i$, kde u_i je i -tá složka vektoru u $[f = 4]$

(g) $g = \sum_{i=1}^4 v_i^2$, kde v_i je i -tá složka vektoru v $[g = 46]$

(h) $h = \sum_{i=1}^4 (u_i - v_i)^2$, kde u_i , resp. v_i je i -tá složka vektoru u , resp. v $[h = 12]$

Příklad 2

- (a) Vytvořte nulový vektor délky 15.
- (b) Vytvořte vektor obsahující všechna sudá čísla od 1 do 100.
- (c) Vytvořte vektor obsahující druhé mocniny čísel od 1 do 10.

Příklad 3

Denní tržby obchodu (v tis. Kč) od 1. 8. do 31. 8. byly následující:

22, 24, 21, 22, 25, 26, 25, 24, 23, 25, 25, 26, 27, 25, 26,
25, 26, 27, 27, 28, 29, 29, 29, 28, 30, 29, 30, 31, 30, 28, 29.

- (a) Vytvořte z těchto hodnot vektor a pojmenujte jej *tržba*.
- (b) Zjistěte průměrnou denní tržbu, rozptyl a směrodatnou odchylku tržby. $[26\ 483, 87; 6\ 636\ 837; 2\ 576, 206]$
- (c) Zjistěte, který den byla tržba nejmenší a který den největší. $[3.8.; 28.8.]$
- (d) Zjistěte, jaká byla tržba 18. 8. $[27\ 000\ \text{Kč}]$
- (e) Určete součet tržeb od 9. 8. do 14. 8. $[151\ 000\ \text{Kč}]$
- (f) Určete kumulované součty tržeb.

$[22, 46, 67, 89, 114, 140, 165, 189, 212, 237, 262, 288, 315, 340, 366, 391, 417, 444, 471, 499, 528,$
 $557, 586, 614, 644, 673, 703, 734, 764, 792, 821\ \text{tis. Kč}]$

Příklad 4

Průzkumu se zúčastnilo 23 vysokoškolsky vzdělaných osob, 15 středoškolsky vzdělaných osob a 4 osoby, které nechtěly uvést své nejvyšší dosažené vzdělání. Vytvořte faktor *vzdelani* a uložte do něj zjištěné hodnoty.