Práce s vektory v R

Vektor (*vector*) je jednorozměrný výčet prvků stejného datového typu. Například řada čísel, řada textových řetězců nebo řada logických hodnot (TRUE nebo FALSE).

Vytvoření a prodloužení vektoru

```
# vektor vytvoříme příkazem c() s výčtem prvků
# vektor "x" bude obsahovat čísla 3, 1, 2 v uvedeném pořadí
x <- c(3, 1, 2)

# zjistíme délku vektoru, tj. počet prvků
length(x)
#> [1] 3

# definujeme další vektory
y <- 1                # "y" má přiřazenou hodnotu 1
z <- c(2)             # "z" je vektor o jednom číselném prvku 2
w <- c(5, 7)          # "w" je vektor o dvou číselných prvcích 5, 7

# zkusíme vektor "x" prodloužit o hodnotu "y" a vektor "z"
x <- c(x, y, z)

# co je x?
x
#> [1] 3 1 2 1 2
```

Vektor může být i logický nebo textový. Vždy však musí být všechny hodnoty stejného datového typu.

```
y <- c(TRUE, TRUE, FALSE, TRUE)          # logický vektor
y <- c("a", "b", "c")                    # textový vektor
```

Vektor obsahující celočíselnou aritmetickou řadu lze s výhodou vytvořit zkratkou s dvojtečkou. `1:10` třeba znamená výčet celých čísel od 1 do 10.

```
x <- 1:10             # vektor o prvcích 1, 2, …, 9, 10
y <- 5:1              # vektor o prvcích 5, 4, 3, 2, 1

# můžeme i s příkazem c(), je to to samé
c(1:10) == 1:10
#>  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

# je to desetkrát potvrzená pravda
```

Pro číselné sekvence můžeme použít příkaz `seq()`.

```
z <- seq(from = 2, to = 10, by = 2)            # vektor o prvcích 2, 4, 6, 8, 10
w <- seq(2, 10, 2)                             # totéž zkráceně
```

Pojmenování prvků vektoru

Někdy může být užitečné mít prvky vektoru pojmenované.

```
# prvky můžeme pojmenovat rovnou
snidane <- c("chleba" = 28, "maslo" = 40, "syr" = 15)

# nebo dodatečně
ceny <- c(28, 40, 15)
potraviny <- c("chleba", "maslo", "syr")

names(ceny) <- potraviny       # pojmenujeme prvky vektoru "ceny"
setNames(ceny, potraviny)      # totéž, ale rovnou nám to výsledek vypíše v konzoli
#> chleba  maslo    syr
#>     28     40     15

# zrušíme pojmenování prvků, příkaz rovnou vypíše výsledek do konzole
unname(ceny)
#> [1] 28 40 15
```

Adresace, výběr prvků vektoru

K jednotlivým prvkům vektoru budeme chtít přistupovat. Je potřeba mít k dispozici nástroj, který nám vytáhne z vektoru jenom daný prvek. Nebo třeba část vektoru. Zkrátka budeme chtít adresovat nebo vybírat část prvků vektoru podle pozice nebo podle vlastnosti.

```
cenypotravin <- c(28, 40, 15, 18, 26)

# vybereme první prvek;
# jiné programovací jazyky mají první prvek očíslovaný jako 0 (nula),
```

```r
# ale R čísluje rovnou od jedničky
cenypotravin[1]
#> [1] 28

# vybereme první tři prvky
cenypotravin[1:3]
#> [1] 28 40 15

# vybereme šestý prvek???
cenypotravin[6]
#> [1] NA

# vybereme poslední prvek
cenypotravin[length(cenypotravin)]
#> [1] 26

# vybereme první a třetí prvek
cenypotravin[c(1, 3)]
#> [1] 28 15

# vybereme prvky od posledního k prvnímu (tj. opačné pořadí)
rev(cenypotravin)
#> [1] 26 18 15 40 28
cenypotravin[length(cenypotravin):1]
#> [1] 26 18 15 40 28
```

Výběr prvků je možný i pomocí logických vektorů (T=true, F=false). To je užitečné, pokud chceme vybrat jen některé prvky podle jejich vlastností.

```r
seznam <- c(TRUE, TRUE, FALSE, TRUE, FALSE)          # logický vektor

cenypotravin[seznam]                      # (sub)vektor vypíše jen ty prvky, které se kryjí s TRUE
#> [1] 28 40 18

# vektor cenypotravin však zůstává nezměněn
cenypotravin
#> [1] 28 40 15 18 26

# takže pro snídani si vybereme jen potraviny, jejichž cena je menší než 28
seznam <- cenypotravin<28
seznam
#> [1] FALSE FALSE  TRUE  TRUE  TRUE
cenypotravin[seznam]
#> [1] 15 18 26
```

Využití tzv. recyclingu: můžeme vypsat jen prvky na určitém n-tém pořadí.

```r
# vybereme pouze prvky na sudých pozicích
cenypotravin[c(F, T)]
#> [1] 40 18

# nebo na lichých pozicích
cenypotravin[c(T, F)]
#> [1] 28 15 26
```

Výpis každé třetí a páté hodnoty z vektoru x

```r
x <- c(34, 65, 4, 0, 56, 23, 54, 17, 2, 4, 8, 5, 44, 84, -5, 4444, 49, 3, 37, 86, 45, 65, 36, 72, 54, 36, 4, 56, 74, 26, 88, 36, 76, 46,
5, 17, 84, 57, 25, -75, 634, 5578, 6, -6, 46, 44, 743, 577, 466, 7, 645, 33, 64, 67)
x[c(F, F, T, F, T)]
#> [1]   4  56  17   4  44  -5   3  86  36  54  56  26  76   5  57 -75   6  46 577
#> [20]   7  64
```

Matematické operace

Na vektory můžeme použít většinu standardních matematických operací.