

Wstęp do środowiska R

R możemy używać jako dobrego kalkulatora. Np.

```
> 2*3      # mnożenie
> 2/3      # dzielenie
> 3 %% 2    # reszta z dzielenia
> 3 %/% 2   # część całkowita z dzielenia
> 2^3      # potęgowanie
> exp(2)   #  $e^2$ 
> sqrt(3)  # pierwiastkowanie (pierwiastek kwadratowy)
> log(3)   # logarytm naturalny z 3
> log(3,10) # logarytm z 3 o podstawie 10
> abs(-3)  # wartość bezwzględna
```

Oczywiście za pomocą R możemy zrobić dużo więcej niż tylko wykonywać działania arytmetyczne. Aby poznać możliwości R, potrzebujemy najpierw zaznajomić się z jego składnią. Zaczniemy od **typów obiektów** w R.

Typ liczbowy (numeric) służy do zapisywania liczb. Trzeba pamiętać, że pisząc ułamki dziesiętne używamy kropki, nie przecinka; np. $1/2$ to 0.5.

Typ czynnikowy (factor) jest przydatny do zapamiętywania danych jakościowych.

Typ znakowy (character) używamy do zapisywania napisów, np.

```
> "napis"
```

Typ logiczny (logical) mają obiekty przyjmujące jedną z dwóch wartości: TRUE (T) lub FALSE (F). Na takich obiektach możemy wykonywać działania logiczne:

symbol działania	jego znaczenie
&	i
	lub
!	nieprawda, że

Ponadto poniższe dwa działania zwracają TRUE lub FALSE.

symbol działania	jego znaczenie
==	sprawdza czy dwa obiekty są równe
!=	sprawdza czy dwa obiekty nie są równe

Typ funkcyjny. Możemy używać gotowych funkcji lub tworzyć własne. Do tworzenia funkcji służy słowo kluczowe `function()`. W nawiasie podajemy argumenty funkcji a dalej instrukcję, co funkcja ma robić.

Np. utwórzmy funkcję wykonującą dodawanie modulo n:

```
> dodawanie.modulo.n <- function(a,b,n)
+ (a+b)%% n
```

i używając jej obliczmy $3 + 4$ modulo 5:

```
> dodawanie.modulo.n(3,4,5)
```

Instrukcja może być także blokiem instrukcji - musimy wtedy blok ten umieścić w nawiasach klamrowych

Wektor to skończony ciąg obiektów tego samego typu. Tworzymy go używając funkcji `c()`, np

```
> c(1,2,3,4,5)
```

Powyższy wektor można uzyskać także pisząc

```
> 1:5
```

Natomiast komenda

```
> rep(1,5)
```

utworzy wektor złożony z pięciu jedynek. Do tworzenia wektorów przydatna jest także funkcja `seq(from, to, by, length.out)`. W wyniku jej wywołania powstanie wektor o pierwszym elemencie równym argumentowi `from`, ostatnim elemencie równym argumentowi `to` i o długości równej argumentowi `length.out`. Zamiast żądanej długości można podać krok (argument `by`) czyli informację o ile mają być zwiększane kolejne wartości wektora. Np. komenda

```
> seq(-2,1,by=0.5)
```

zwróci wektor złożony z liczb -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0.

Macierz tworzymy używając funkcji

```
> matrix(data, nrow, ncol, byrow = FALSE)
```

gdzie

- argument `data` określa wektor z elementami macierzy,
- argument `nrow` określa liczbę wierszy macierzy, zamiennie można podawać `ncol` czyli liczbę kolumn macierzy,
- jeśli `byrow = FALSE` (ta wersja jest ustawiona domyślnie, czyli jeśli pominiemy argument `byrow`, to R właśnie taką wersję przyjmie), to elementy macierzy są do niej wpisywane kolumnami; jeśli `byrow = TRUE`, to wierszami.

Np. aby utworzyć macierz

$$\begin{bmatrix} -1 & 2 & 3 \\ 0 & 5 & 6 \end{bmatrix}$$

użyjemy komendy

```
> matrix(data=c(-1,2,3,0,5,6), nrow=2, byrow=T)
```

lub

```
> matrix(data=c(-1,0,2,5,3,6), nrow=2)
```

Ramka danych to zestaw wektorów o tej samej długości. Funkcja

```
> data.frame(wektor1,wektor2,...,wektorN)
```

tworzy ramkę danych z podanych w nawiasie wektorów. Natomiast

```
> read.table(file,header=FALSE)
```

wczytuje dane z pliku o strukturze ramki danych. Argument `file` jest typu znakowego i określa ścieżkę wskazującą plik z danymi. Aby uniknąć wpisywania ścieżki, możemy użyć funkcji `file.choose()`, która otworzy okno systemowe i pozwoli na wskazanie pliku poprzez klikanie. Jeśli argument `header = TRUE`, to pierwszy wiersz w pliku jest traktowany jako nazwy kolumn.

Przykład 1. Utworzymy ramkę danych zawierającą informacje o płci, wadze i wzroście 10 osób:

płeć	wzrost (w cm)	waga (w kg)
kobieta	174	57
kobieta	168	58
kobieta	170	81
kobieta	165	62
kobieta	162	53
mężczyzna	177	89
mężczyzna	182	102
mężczyzna	168	62
mężczyzna	172	71
mężczyzna	191	97

Sposób I: Tworzymy najpierw trzy wektory:

```
> plec <- c("K", "K", "K", "K", "K", "M", "M", "M", "M", "M")
lub krócej
> plec <- rep(c("K", "M"), c(5, 5))
> wzrost <- c(174, 168, 170, 165, 162, 177, 182, 168, 172, 191)
> waga <- c(57, 58, 81, 62, 53, 89, 102, 62, 71, 97)
```

Wektor `plec` ma elementy typu znakowego. Przerabiamy je na typ czynnikowy pisząc

```
> plec <- as.factor(plec)
```

Następnie sklejamy wektory w ramkę danych:

```
> dane <- data.frame(plec, waga, wzrost)
```

Na koniec wyświetlamy dane by sprawdzić czy wszystko jest w porządku:

```
> dane
lub
> print(dane)
```

Sposób II: Trzy kolumny z danymi (`plec`, `waga` i `wzrost`) zapisujemy w notatniku i zapamiętujemy.

Następnie plik ten wczytujemy do R, używając funkcji `read.table()`.

```
> dane <- read.table(file.choose(), header=T)
```

Odwoływanie się do elementów wektora lub ramki danych

Posłużmy się danymi z przykładu 1.

- Odczytajmy jaka jest płeć trzeciej osoby:

```
> dane$plec[3]
```

lub

```
> dane[3, 1]
```

- Wypiszmy wagę jedynie tych osób, których wzrost przekracza 170 cm:

```
> dane$waga[dane$wzrost > 170]
```

- Wypiszmy wagę i wzrost jedynie dla mężczyzn:

```
> dane[dane$plec == "M", c(2, 3)]
```

lub

```
> dane[dane$plec != "K", -c(1)]
```

Na wektorach możemy wykonywać działania jak na liczbach

Np. dla osób z przykładu 1:

- obliczymy wzrost w metrach

```
> wzrost.w.metrach <- dane$wzrost/100
```

```
> wzrost.w.metrach
```

- obliczymy BMI (waga w kg dzielona przez kwadrat wzrostu w metrach)

```
> bmi <- dane$waga/(wzrost.w.metrach)^2
```

```
> bmi
```