

# Przetwarzanie i wizualizacja danych

prof. UAM dr hab. Tomasz Górecki

tomasz.gorecki@amu.edu.pl

Zakład Statystyki Matematycznej i Analizy Danych  
Wydział Matematyki i Informatyki  
Uniwersytet im. Adama Mickiewicza w Poznaniu



## Uwagi ogólne

- Zmienne nie muszą być deklarowane.
- Klasycznym operatorem przypisania w R jest `<-` (lub `->`), natomiast wykorzystuje się również znak równości (`=`).
- Komentarz poprzedzamy znakiem hash (`#`), wszystko do końca linii jest już komentarzem.
- Dla czytelności kodu można używać spacji gdziekolwiek się chce, są one pomijane.
- R rozróżnia wielkie i małe litery.
- W nazwach można używać wszystkich znaków A-Za-z0-9 oraz „.” i „\_”. Dodatkowo nazwa musi zaczynać się od „.” lub litery, jeśli jednak zaczyna się od „.”, to drugi znak nie może być cyfrą.

# Instrukcja warunkowa

Najprostsza postać tej instrukcji to:

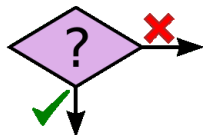
**if** (**warunek**) wyrażenie

Instrukcja ta sprawdza czy *warunek* jest prawdziwy, jeśli tak to wykonuje *wyrażenie*, w przeciwnym razie nic nie robi. Jeśli chcemy aby w przypadku nieprawdziwości warunku również „coś się działo” musimy użyć następującej postaci instrukcji warunkowej:

**if** (**warunek**) wyrażenie1 **else** wyrażenie2

## Instrukcja warunkowa

Operator	Działanie
&	Logiczne AND (i)
	Logiczne OR (lub)
!	Logiczne NOT (nieprawda, że)
==	równy
!=	różny
<	mniejszy
>	większy
<=	mniejszy bądź równy
>=	większy bądź równy
%in%	w zbiorze



W R mamy wszystkie trzy klasyczne postaci pętli:

`for (licznik in start:koniec)` wyrażenie  
`while (warunek)` wyrażenie  
`repeat` wyrażenie

# Pętla

Pętla **for** jest używana wtedy gdy z góry wiemy ile kroków będzie konieczne do zakończenia zadania. Określamy to poprzez zmienną *licznik*, dla której zadajemy zakres zmian. Pętla **while** polega na tym, że wyrażenie jest wykonywane tak długo dopóki warunek jest prawdziwy, w momencie gdy warunek staje się fałszywy pętla się kończy. Ostatnim rodzajem pętli jest pętla typu **repeat**. W tym przypadku wyrażenie jest tak długo wykonywane aż gdzieś wewnątrz nakażemy zakończenie pętli za pomocą polecenia **break**. Jest to jedyna możliwość zakończenia tego typu pętli. Można tej funkcji również używać do wcześniejszego opuszczenia innych pętli. Czasami przydaje się również polecenie **next**, które przerywa bieżącą iterację i przechodzi do następnej.

Własne funkcje tworzymy następująco

NazwaFunkcji = **function(argumenty)** ciało funkcji

Tak skonstruowana funkcja może być wywołana za pomocą swojej nazwy, jeśli funkcja nie posiada argumentów wywołujemy ją za pomocą `NazwaFunkcji()`. Jeśli chcemy aby pewne argumenty miały wartości domyślne, to możemy wykorzystać konstrukcję:

NazwaFunkcji = **function(arg1 = wartość)** ciało funkcji

W przypadku takiej konstrukcji `arg1` ma wartość domyślną i jeśli wywołamy funkcję `NazwaFunkcji()`, to zostanie ona wywołana właśnie z takim domyślnym argumentem.

Funkcja zwraca ostatnią linię kodu. Jeśli planujemy zakończyć wykonywanie funkcji wcześniej należy użyć polecenia **return**.



## Control Structures in R

If else condition

for loops

while loops

repeat statement

break statement

next statement

functions in R

## Podstawowe funkcje matematyczne oraz pomocnicze

$\text{abs}(x)$	Wartość bezwzględna
$\text{exp}(x)$	Funkcja wykładnicza
$\text{log}(x)$	Logarytm naturalny, istnieją też funkcje $\text{log}_2(x)$ oraz $\text{log}_{10}(x)$ , które są odpowiednio logarytmem o podstawie 2 i 10. Ogólna funkcja licząca logarytm o podstawie $b$ , ma postać $\text{log}(x, b)$
$\text{sqrt}(x)$	Pierwiastek kwadratowy
$\text{sum}(x)$	Suma elementów

## Podstawowe funkcje matematyczne oraz pomocnicze

<code>min(x)</code> , <code>max(x)</code>	Minimum, maksimum. Funkcje <code>which.min</code> oraz <code>which.max</code> zwracają numer odpowiedniej obserwacji (pierwszej). Funkcje <code>pmin</code> oraz <code>pmax</code> wywołane dla dwóch wektorów zwracają odpowiednio minimum (maksimum) dla każdej pary.
<code>prod(x)</code>	Iloczyn elementów
<code>round(x)</code>	Zaokrąglanie, <code>round(x, n)</code> - zaokrąglanie do $n$ miejsc. R zaokrągla 0.5 do 0, odwrotnie niż np. Matlab.
<code>sin(x)</code> , <code>cos(x)</code> , <code>tan(x)</code>	Funkcje trygonometryczne

# Podstawowe funkcje matematyczne oraz pomocnicze

<code>length(x)</code>	Liczba elementów wektora, funkcje <code>nrow</code> oraz <code>ncol</code> podają odpowiednio liczbę wierszy i kolumn ramki danych
<code>rev(x)</code>	Odwraca porządek wektora
<code>sort(x)</code>	Sortuje wektor