

Podstawowe komendy

setwd(ścieżka) ustawia katalog roboczy określony **ścieżka** (np. **'D:/R'**).

getwd() określa bieżący katalog roboczy

dir() informacja o plikach w katalogu roboczym (lub podanej ścieżce)

ls() wyświetla wszystkie obiekty w pamięci

rm(x) usuwa obiekt **x** z pamięci

rm(list=ls()) usunięcie wszystkich elementów z pamięci

source('plik.R') uruchomienie komend zapisanych w **plik.R**

q() kończy pracę programu

summary(x) informacja podsumowująca o obiekcie **x**; np. w przypadku obiektów przechowujących dane liczbowe są to statystyki opisowe.

Pomoc



help(funkcja) opis funkcji



?funkcja opis funkcji


help.search('mean') szukanie pomocy nt. wartości średniej (ang. mean)

Skróty klawiszowe



Konsola




 oraz  przewijanie historii komend

 +  wyczyszczenie konsoli z dotychczasowych wydruków

 przerwanie obliczeń

Edytor kodu

 +  kompilacja zaznaczonego kodu lub bieżącej linii kodu

 +  +  kompilacja całego kodu

Przypisanie

a=2 przypisanie obiektowi **a** liczby 2

a<-2 j.w.

Podstawowe typy danych

dane liczbowe **1, 2.34**

dane tekstowe **'A', "ALA"**

dane logiczne **TRUE, FALSE**

typ danych czynnikowych **factor('a', 'b', 'a')**

gdzie **Levels** to powtarzające się elementy, tj. czynniki **Levels: a b**

Warunki logiczne

x==y $x = y$ **x** równe **y**

x!=y $x \neq y$ **x** różne **y**

x<y $x < y$ **x** mniejsze od **y**

x>y $x > y$ **x** większe od **y**

x>=y $x \geq y$ **x** nie mniejsze od **y**

x<=y $x \leq y$ **x** nie większe od **y**

Łączenie warunków logicznych

!W negacja warunku **W**

W1 && W2 koniunkcja warunków **W1** oraz **W2**

W1 || W2 alternatywa warunków **W1** oraz **W2**

W1 & W2 koniunkcja dla elementów warunków **W1** oraz **W2**

W1 | W2 alternatywa dla elementów warunków **W1** oraz **W2**

Funkcje matematyczne

x+y	$x + y$	x-y	$x - y$
x*y	xy	x/y	$\frac{x}{y}$
x^y	x^y	sqrt(x)	\sqrt{x}
abs(x)	$ x $	sign(x)	$\begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$
x%%y	reszta z dzielenia x przez y		
round(x,y)	zaokrąglenie liczby x do y miejsc po przecinku		
floor(x)	zaokrąglenie w <i>dół</i> liczby x		
ceiling(x)	zaokrąglenie w <i>górze</i> liczby x		
sin(x)	$\sin(x)$	cos(x)	$\cos(x)$
tan(x)	$\text{tg}(x)$	arcsin(x)	$\arcsin(x)$
arccos(x)	$\arccos(x)$	arctan(x)	$\arctg(x)$
log(x)	$\ln(x)$	log(x,base=n)	$\log_n(x)$
log1p(x)	$\ln(x+1) \approx x$	log10(x)	$\log_{10}(x)$
exp(x)	$\exp(x)$		

Definiowanie wektorów

c(n,m,l) 3-elementowy wektor składający się z liczb n, m, l .

n:m wektor zawierający liczby całkowite od n do m

rep(1,n) n -wymiarowy wektor z liczbą 1


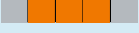



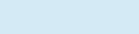
rep(1:3,n) wektor składający się z n powtórzeń ciągu 1:3

rep(1:3,each=n) wektor powtarzający każdy element ciągu 1:3 n -krotnie

seq(n,m,by=1) wektor będący ciągiem arytmetycznym zaczynający się od n a kończący na m o różnicy 1

seq(n,m,length.out=1) 1-wymiarowy wektor będący ciągiem arytmetycznym zaczynający się od n a kończący na m

Elementy wektora

	x[3]	trzeci element wektora x
	x[2:4]	od drugiego do czwartego elementu wektora x
	x[c(1,3,5)]	pierwszy, trzeci i piąty element wektora x
	x[-c(2,4)]	wszystkie elementy wektora x oprócz drugiego i czwartego
	x[c('A','C','E')]	elementy wektora x pod nazwą 'A', 'C' lub 'E'
	x[x<=3]	wszystkie elementy wektora x nie większe od 3

Manipulowanie elementami wektorów

sort(x) sortowanie elementów wektora x

rev(x) wektor z odwróconą kolejnością elementów x

which.min(x) wskazuje indeks najmniejszego elementu wektora x

which.max(x) wskazuje indeks największego elementu wektora x

which(x=<3) wskazuje indeksy elementów wektora x , które są **nie większe od 3**

Funkcje statystyczne i matematyczne

mean(x)	wartość średnia x
sum(x)	suma elementów x
sd(x)	odchylenie standardowe x
var(x)	wariancja x
weighted.mean(x,w)	średnia ważona x o wagach w
median(x)	mediana x
quantile(x,q)	kwantyle rozkładu x o prawdopodobieństwie opisanym wektorem q
min(x)	wartość minimalna x
max(x)	wartość maksymalna x
cor(x,y)	współczynnik korelacji pomiędzy x i y
cumsum(x)	kolejne elementy stanowią skumulowaną sumę x
prod(x)	iloczyn wszystkich elementów wektora x
cumprod(x)	kolejne elementy stanowią iloczyn dotychczasowych elementów wektora x

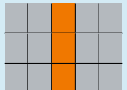
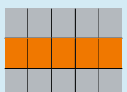
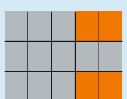
Definiowanie macierzy

matrix(v,ncol=3,nrow=4)	macierz o 3. kolumnach i 4 wierszach; zawierająca elementy wektora v
diag(n)	n -wymiarowa macierz diagonalna

Operacje na macierzach

t(X)	transpozycja macierzy X
X%*%Y	mnożenie macierzy X i Y
solve(X)	odwracanie macierzy X
diag(X)	elementy diagonalne macierzy X
trace(X)	śląd macierzy
chol(X)	dekompozycja Choleskiego macierzy X
eigen(X)	wartości własne oraz wektory własne macierzy X

Elementy macierzy

	X[,3]	trzecia kolumna macierzy X
	X[2,]	drugi wiersz macierzy X
	X[c(1,3),4:5]	pierwszy i trzeci wiersz oraz czwarta i piąta kolumna macierzy X

data.frame

Data.frame pozwala przechowywać zmienne o tym samym wymiarze, ale o różnych typach danych, np. dane liczbowe oraz tekstowy typ danych.

Przykład

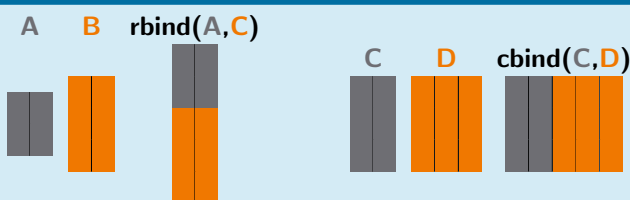
```
DF=data.frame(liczby=1:3,litery=c('A','C','B'))
```

Odwołanie do zmiennych zapisanych w **data.frame** jest możliwe w analogiczny sposób jak w przypadku macierzy, jak i korzystając z $\$$.

Przykład (cd)

DF	DF[2,3]	drugi wiersz trzeciej kolumny
liczby litery	DF\$liczby	zmienna liczby
1 'A'	DF\$litery[2]	drugi element zmiennej litery
2 'B'		
3 'C'		

Łączenie macierzy i data.frame



Listy list()

Listy **list()** pozwalają przechowywać zmienne o **różnym rozmiarze** oraz o różnych typach danych, np. dane liczbowe oraz tekstowy typ danych.

Przykład

```
Lista=list(x=1:5,y='A',z=matrix(1:9,3,3))
```

Elementy listy list()

Lista\$x	element Listy zapisany pod nazwą x
Lista[['x']]	element Listy zapisany pod nazwą x
Lista[[1]]	pierwszy element Listy
Lista[[1]][3]	trzeci element pierwszego elementu Listy

Wymiary wektorów, macierzy, list i data.frame

length(x)	liczba elementu x (wektory i listy)
dim(x)	wymiary obiektu x (macierze i data.frame)
ncol(x)	liczba kolumn obiektu x (macierze i data.frame)
nrow(x)	liczba wierszy obiektu x (macierze i data.frame)

Funkcje function()

f=function(arg){
obliczenia
return(wynik)
}

Procedury deklarowane przez **function()** pozwalają na wykonanie **obliczeń** dla **argumentu (arg)**. Produktem końcowym funkcji jest **wynik**, który jest dostępny w pamięci dzięki **return()**.

Przykład funkcji function()

```
sq=function(x){
  y=x^2
  return(y)
}
```

Funkcja **sq()** pozwala obliczyć kwadrat liczby.

Pętla for()

```
for(i in zakres){
  instrukcja
}
```

Pętla **for()** pozwala na wykonanie **instrukcji** dla pewnego **zakresu** zmiennej i .

Przykład pętli for()

```
for(i in 1:5){
  print(i)
}
```

Pętla pozwala drukować (funkcja **print()**) liczby z **zakresu** zmiennej i , tj. od 1 do 5 (**i in 1:5**).

Pętla while()

```
while(warunek){
  instrukcja
}
```

Pętla **while()** pozwala na wykonanie **instrukcji** dopóki jest spełniony **warunek**.

Przykład pętli while()

```
i=1
while(i<=5){
  print(i)
  i=i+1
}
```

Po wstępnej deklaracji (**i=1**), w każdej kolejnej iteracji drukowana jest liczba i (**print(i)**) a następnie zwiększana jest jej wartość (**i=i+1**) aż do momentu gdy jest mniejsza od 5 (**i<=5**).

Instrukcja warunkowa if()

`if (warunek){
 instrukcja
}`
Instrukcja warunkowa `if()` umożliwia uzależnienie obliczeń (tj. `instrukcji`) w zależności od spełnienia warunku logicznego (`warunek`).

Przykład instrukcji if()

```
if(x<=5){  
  print(x)  
}
```

Jeżeli liczba `x` jest mniejsza od `5` (`x<=5`) to jej wartość będzie drukowana (`print(x)`).

Rozkłady statystyczne

`funkcja``rozkład(argumenty funkcji, argumenty rozkładu)`

Np. `rnorm()` pozwala na losowanie z rozkładu normalnego.

Funkcje wykorzystujące rozkłady statystyczne

`drozkład(x,...)` funkcja gęstości prawdopodobieństwa rozkładu dla wektora kwantyli `x`,
`prozkład(x,...)` dystrybuanta rozkładu dla wektora kwantyli `x`
`qrozkład(p,...)` pozwala na uzyskanie kwantyli rozkładu dla wektora prawdopodobieństw `p`
`rrozkład(n,...)` pozwala na losowanie z rozkładu `n` n liczb

Rozkłady statystyczne

`norm` rozkład normalny; dodatkowe argumenty to `mean` (średnia) oraz `sd` (odchylenie standardowe)
`unif` rozkład jednostajny; dodatkowe argumenty to `min` (minimalna wartość) oraz `max` (maksymalna wartość)
`t` rozkład t-Studenta; dodatkowe argumenty to `df` (liczba stopni swobody) oraz `ncp` (non-centrality parameter)
`chisq` rozkład χ^2 ; dodatkowe argumenty to `df` (liczba stopni swobody) oraz `ncp` (non-centrality parameter)
`F` rozkład F; dodatkowe argumenty to `df1` oraz `df2` (liczba stopni swobody) oraz `ncp` (non-centrality parameter)

Regresja liniowa lm()

`model=lm(y~ x1+x2, dodatkowe argumenty)`

W specyfikacji modelu regresji liniowej należy uwzględnić zmienną objaśnianą (`y`) oraz zmienne objaśniające (`x1` oraz `x2`). Brak wyrazu wolnego oznaczany jest jako `-1`. Wyrażenie `y~ x1+x2` jest formułą (klasa obiektów).

Dodatkowe argumenty lm()

`data` zbiór danych (data.frame), którego zmienne są wykorzystywane w estymacji
`subset` wektor określający podzbiór danych jeżeli oszacowania mają zostać uzyskane na próbie
`weights` wektor wag dla obserwacji
`na.action` określa strategię szacowania parametrów w przypadku braku danych

Pakiety

`library(pakiet)` ładowanie pakietu
`install.packages('pakiet')` instalacja pakietu
`help(package=pakiet)` wyświetla listę dostępnych funkcji w pakiecie.

Wykresy

`plot(x)` wykres zmiennej `x`
`plot(x,y)` wykres rozrzutu czyli tzw. *scatterplot* zmiennej `y` wobec zmiennej `x`
`hist(x)` histogram zmiennej `x`
`boxplot(x)` wykres pudełkowy zmiennej `x`
`barplot(x)` wykres słupkowy zmiennej `x`