

Podstawowe dane

> jest zachętą do wprowadzania poleceń

+ znak kontynuacji, wprowadzamy dalszy ciąg polecenia

znak komentarza, wszystko co znajdzie się za tym znakiem jest pomijane przez program

Napis **[1]** rozpoczynający linie związany jest ze sposobem działania funkcji wyświetlającej liczby. Jest to indeks elementu wyświetlanego bezpośrednio za tym nawiasem.

Zmienne

Zmienne służą do przechowywania wprowadzonych danych lub wyników wykonanych poleceń. Etapy procesu tworzenia zmiennej:

1. Podaj nazwę zmiennej np. *lata*
2. Dodaj znak przypisania wartości do zmiennej <- lub znak równości =
3. Dodaj zawartość zmiennej np. 5 i naciśnij enter
4. Sprawdź zawartość zmiennej przez wpisanie jej nazwy. W konsoli powinien pojawić się wynik

W nazwach zmiennych nie używamy spacji ani znaków specjalnych. Dopuszczalne są: litery, cyfry, „.”, „_”. Zaczynamy zawsze od liter. **Uwaga!!!** Środowisko R rozróżnia małe litery od dużych.

Typy obiektów

Typ liczbowy – liczby całkowite i rzeczywiste, specjalne oznaczenie: NaN – not a number

Typ znakowy – napisy (łańcuchy znaków), rozpoczynane są znakiem ‘ lub ” i tak kończone

Typ logiczny – przyjmuje tylko dwie wartości, T – TRUE lub F – False

Typ czynnikowy (kategoryczny) – przechowywanie wartości występujących na kilku poziomach (w kilku kategoriach, klasach np. płeć, rasa)

Wektor – uporządkowany zbiór obiektów tego samego typu

Lista – wektor zawierający zmienne z różnego typu

Macierz – wszystkie elementy są tego samego typu

Ramka danych – lista wektorów o takiej samej długości, wyświetlana jak macierz, w której elementy w kolumnie są tego samego typu

Funkcje

Są to wbudowane fragmenty kodu, które możemy wykorzystać później przywołując nazwę funkcji.

Wywołanie funkcji: *nazwa(argumenty)*

Funkcja obliczająca średnią (z ang. mean) dla danych ze zmiennej „liczby” wygląda następująco: *mean(liczby)*

Tworzenie własnej funkcji:

Nazwa funkcji <- *function(lista argumentów)*

```
{  
  blok instrukcji  
}
```

Wektory

Funkcja *c()* łączy podane argumenty w wektor, czyli uporządkowany ciąg danych.

Funkcja tworząca wektor zawierający liczby 1,2 i 3 będzie wyglądała następująco: *c(1,2,3)*

Generowanie ciągów:

Operator: powoduje wygenerowanie wektora z liczbami całkowitymi zawierającymi się w danym przedziale. np. *c(1:4)* jest równoważny z *c(1,2,3,4)*

- Funkcja *seq* zmienia krok na inny niż 1 np.: *seq(1,6, by=0.5)*
- Funkcja *rep* pozwala na powtórzenie wektora określoną ilość razy np.: *rep(c(1:3),5)*

indeksowanie wektorów

element 2 wektora **a**: *a[2]*

wszystkie elementy wektora **a** poza drugim: *a[-2]*

element pierwszy i piąty wektora **a**: *a[c(1,5)]*

wektor logiczny, sprawdzamy czy kolejne elementy są większe niż 5: *a>5*

wyświetla kolejne elementy większe niż 5: *a[a>5]*

przykładowe funkcje operujące na wektorach

Składnia: *nazwa funkcji(nazwa wektora)*

edit — manualna edycja wektora lub ramki danych, aby zapisać zmiany: *nowa_ramka<-edit(stara_ramka)*

max, min — maksymalna i minimalna wartość z wektora

mean — średnia arytmetyczna. Jeśli podamy dodatkowy parametr *trim* - to funkcja policzy, średnią po odrzuceniu określonego odsetka wartości skrajnych, np. *mean(x,trim=0.1)* – to średnia z *x* po odrzuceniu 10% wartości skrajnych

quantile — dowolny kwantyl, np. *quantile(x,.5)* - to mediana z *x*, inaczej *median(x)*

sd — odchylenie standardowe

var — wariancja

length — długość wektora (liczba elementów)

sum — suma elementów wektora

prod — iloczyn elementów wektora

sort — daje wektor z wartościami uporządkowanymi rosnąco

summary — statystyka opisowa dla obiektu (np. wektor, ramka danych)

Macierz

Zamiana wektora w macierz o dwóch wierszach i pięciu kolumnach

dim(wektor)<-c(2,5)

Tworzenie macierzy o trzech kolumnach:

macierz=matrix(data=c(9,2,3,4,5,6),ncol=3)

Ramki danych

indeksowanie

- *ramka\$columna1* — odwołujemy się do kolumny 1

- *ramka[2,]* — wiersz drugi

- *ramka[,4]* — kolumna 4

- *ramka[3,6]* — wartość z 3 wiersza i 6 kolumny

funkcja *attach(ramka)* — po wywołaniu tej funkcji na ramce danych wystarczy podać nazwę kolumny, żeby się do niej odwołać, potem możemy anulować przez funkcję *detach(ramka)*. Na raz możemy „przyłączyć” tylko jedną ramkę danych.

dodawanie i usuwanie kolumn

ramka[\"nowa kolumna\"]<-kolumna1+kolumna2 - dodaje nową kolumnę z sumą dwóch pierwszych

zmiana nazwy kolumny

colnames(dane)[colnames(dane) == \"stara nazwa\"] <- \"nowa nazwa\"

np.: *colnames(dane)[1:5] <- c(\"Imie\", \"Nazwisko\", \"Rok_ur\", \"Wiek\", \"Miasto\")*

tworzenie ramki danych

data.frame(wektory które chcemy połączyć oddzielone przecinkami)

np.: *d=data.frame(ID=1:5,gender=c(\"M\", \"F\", \"F\", \"M\", \"F\"),score=c(8,3,6,5,5))*

Tworzenie ramki danych: *cbind* (po kolumnach) i *rbind* (po wierszach)

a=c(18,22,34)

b=c(14,29,10)

cbind(a,b)

przydatne funkcje:

str(ramka) — struktura obiektu

dim(ramka) — zwraca liczbę wierszy i kolumn w ramce danych (można również użyć *nrow* lub *ncol*)

names(ramka) — zwraca nazwy kolumn z ramki danych

head(ramka) — wyświetla kilka pierwszych wierszy z ramki danych

podsumowania zmiennych

Dla macierzy: oblicza średnią dla każdego wiersza (1) lub kolumny (2)

`apply(macierz,1,mean)`

Dla ramki danych: funkcja (np. mean) dla zmienna1 pod względem zmienna2 dla każdej kolumny
`tapply(zmienna1, zmienna2, funkcja)`

Tabela liczebności

`table(ramka$zmienna1)` # dla jednej zmiennej

`table(ramka$zmienna1,ramka$zmienna2)` # dla dwóch zmiennych

Tabela częstości

`table(ramka$zmienna1) / length (ramka$zmienna1)`

podzbiory

`subset()` - Funkcja za pomocą której wybieramy tylko dane które spełniają podany warunek

np. Funkcja wybierająca z ramki danych tylko te wiersze dla których wartość w kolumnie pierwszej jest równa „tak” wygląda następująco: `subset(ramka, ramka$skolumna1=="tak")`

Operatory

<code>+, -, *, /, ^</code>	Standardowe operatory dwuargumentowe, kolejno: dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie
<code>%*%</code>	Mnożenie macierzowe
<code>%%</code>	Reszta modulo z dzielenia
<code><, ==, >, <=, >=</code>	Standardowe operatory porównywania wartości liczbowych, kolejno: mniejsze, równe, większe, mniejsze lub równe, większe lub równe
<code>!, !=</code>	Operator negacji logicznej i nierówności
<code>&,&&, , </code>	Logiczny iloczyn („i”, $p \cap q$) oraz suma („lub”, $p \cup q$) dla pojedynczych wartości (pojedyncze) oraz równocześnie wszystkich wartości w liście/wektorze (podwójne)
<code>isTRUE(x)</code>	Sprawdza czy x jest prawdą

Pakiety

`install.packages(„nazwa pakietu”)`

Instalacja pakietów (bibliotek)

`library(nazwa pakietu)`

ładowanie pakietu – dopiero po załadowaniu możemy korzystać z funkcji i danych w nim dostępnych

Wykresy

Wykres punktowy – `plot(zmienna)` lub `plot(zmienna1, zmienna2)`

Wykres pudełkowy – `boxplot(zmienna)` lub `boxplot(zmienna1~zmienna2)`

Wykres kolumnowy – najpierw `barplot(table(zmienna))`

Histogram – `hist(zmienna)`

Przykładowe argumenty funkcji:

`main` – tytuł np. `main="wykres 1"`

`xlab, ylab` – tytuły osi np. `xlab="nazwa osi x"`

`ylim` – dolna i górna granica wartości dla osi x np. zakres od 0 do 50 co 5: `ylim=c(0,50,5)`

`breaks` – liczba przedziałów przy tworzeniu histogramu

przykłady:

`barplot(table(plec,typ_ur),ylim=c(0,50),col=c('pink','skyblue'),
ylab='liczba osobników',xlab='typ urodzenia')`

legenda:

`x` – położenie legendy na osi x

`y` – położenie legendy na osi y

`legend` – nazwy używane w opisie

`fill` – kolory używane w opisie

przykłady:

`legend(x="topleft",legend=c(„a”, „b”), fill=rainbow(2))`

lub

`legend(x=x position, y= y position, legend=c(„a”, „b”), fill=rainbow(2))`

kolory (n – ile kolorów z zestawu wygenerować)

`rainbow(n); heat.colors(n); terrain.colors(n); topo.colors(n); cm.colors(n); grey(n)`

Bazy danych wykorzystane na zajęciach

Baza litters, pakiet „DAAG”, zmienne:

lsize – wielkość miotu

bodywt – masa ciała

brainwt – masa mózgu

Baza Melanoma, pakiet „MASS”, zmienne:

time – czas przeżycia w dniach (od operacji do czasu badania)

status – 1 = śmierć z powodu czerniaka, 2 = żyje, 3 = śmierć z innych przyczyn

sex – 1 = mężczyzna 0 = kobieta

age – wiek

year – rok operacji

thickness – grubość guza w mm

ulcer – obecność lub brak wrzodu

Baza survey, pakiet „MASS”, opis: <http://www.stats4stem.org/r-survey-data.html>

Baza mammals, pakiet „MASS”, opis: <http://www.stats4stem.org/r-mammals-data.html>

Statystyka – przykładowe testy

Test na proporcje – mała próba:

`binom.test(x=9,n=20,p=0.2)` – 9 sukcesów w 20 próbach, prawdopodobieństwo sukcesu 20%

Test na proporcje – duża próba (odpowiednik testu U):

`prop.test(x=2000,n=100000,p=0.2)` -) – 2000 sukcesów w 100k próbach, prawdopodobieństwo sukcesu 20%

Test na jednorodność wariancji: `var.test(X,Y)`

Test na rozkład normalny: `shapiro.test(X)`

Test na średnią – zmienna ma rozkład normalny

`t.test(X,mu=3)` # jedna populacja

`t.test(X,Y)` # dwie populacje, założeniem jest jednorodność wariancji

dodatkowe argumenty: *alternative*=“greater” lub „less” – dla hipotezy alternatywnej odpowiednio > lub <

Test chi-kwadrat na zgodność rozkładu

`kostka=c(6,12,9,11,15,7)`

`pr=c(1/6,1/6,1/6,1/6,1/6,1/6)`

`chisq.test(kostka,p=pr)`

Test chi-kwadrat na niezależność

`a<-c(21,12,32)`

`b<-c(33,22,33)`

`chisq.test(bind(a,b))`

Korelacja

`cor(X,Y)` – zwraca wsp. korelacji liniowej dla zmiennych X i Y

`cor.test(X,Y)` – zwraca wsp. korelacji liniowej dla zmiennych X i Y razem z testem istotności

Prosta regresja liniowa:

`model<-lm(y1~y2)`

`summary(model)` # wyniki

`plot(y1,y2); abline(model)` #wykres

Źródła i polecana literatura:

Biecek P. 2014. Przewodnik po pakiecie R. Oficyna wydawnicza GIS. – Rozdział 1 wersji ze starszego wydania dostępny za darmo w Internecie: <http://cran.r-project.org/doc/contrib/Biecek-R-basics.pdf>

Komsta Ł. 2004. Wprowadzenie do R. Wersja PDF dostępna za darmo w Internecie: <http://cran.r-project.org/doc/contrib/Komsta-Wprowadzenie.pdf>

Rybiński M. 2011. Krótkie wprowadzenie do R dla programistów, z elementami statystyki opisowej. WMIM UW. Dostępny za darmo w Internecie: <http://www.mimuw.edu.pl/~trybik/edu/0809/rps/r-skrypt.pdf>

Paradis E. 2005. R for Beginners. Dostępny za darmo w Internecie: http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf