Pakiety Matematyczne - R Zestaw 2

Data Frames (ramki/zbiory danych)

Data Frames to "macierze" o różnych typach kolumn. W R można znaleźć przykłady wbudowanych zbiorów danych:

```
#przykład wbudowanego zbioru zawierającego dane dotyczące samochodów
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
#przeglądanie danych
#View(mtcars)
#pięć pierwszych obserwacji ze zbioru mtcars
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
```

```
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
#pięć ostatnich obserwacji ze zbioru mtcars
tail(mtcars)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

Tworzenie własnych zbiorów danych

```
#  tworzymy przykładowe wektory, które posłużą do utworzenia zbioru danych
zm1<-c(6.1, 2.2, 1.7, 4.5)
zm2<-c('K', 'M', 'K', 'K')
zm3<-c(T,F,T,F)
# tworzymy zbiór danych zd, w którym każda kolumna jest innego typu
zd<-data.frame(zm1, zm2, zm3)
```

Badanie struktury zbioru danych (liczbę kolumn (zmiennych) i wierszy (obserwacji))

```
str(zd)
```

```
## 'data.frame':    4 obs. of  3 variables:
##  $ zm1: num  6.1 2.2 1.7 4.5
##  $ zm2: chr  "K" "M" "K" "K"
##  $ zm3: logi  TRUE FALSE TRUE FALSE
```

Zauważmy, że druga kolumna utworzona przy pomocy wektora zm2 jest typu Factor. Jest to tzw. typ czynnikowy (nazywany również wyliczeniowym lub kategorycznym). Jest on przydatny do przechowywania wektorów wartości występujących w kilku kategoriach (np. płeć).

Aby wydobyć kolumnę ze zbioru używamy **$**:

```
zd$zm2
```

```
## [1] "K" "M" "K" "K"
```

```
# wybrana kolumna jest typu czynnikowego
is.factor(zd$zm2)
```

```
## [1] FALSE
```

```
# a sam wektor zm2 nie
is.factor(zm2)
```

```
## [1] FALSE
```

Wybieranie ze zbioru danych zmiennych i obserwacji spełniających zadane warunki

```
# wybieramy ze zbioru zd elementy takie, że zm1 jest większa niż 3
subset(zd,zm1>3)
```

```
##   zm1 zm2   zm3
## 1 6.1   K  TRUE
## 4 4.5   K FALSE
```

- **zd[m:n,p:r]** wybiera ze zd obserwacje od m-tej do n-tej dla zmiennych od p-tej do r-tej
- **zd[“nazwa_zmiennej”]** - wybiera ze zd obserwacje dla zmiennej o podanej nazwie
- **zd[m:n,”nazwa_zmiennej”]** - wybiera ze zd obserwacje od m-tej do n-tej dla zmiennej o podanej nazwie

```
# wybieramy ze zbioru same kobiety
kobiety<-subset(zd,zm2=="K")
kobiety
```

```
##   zm1 zm2   zm3
## 1 6.1   K  TRUE
## 3 1.7   K  TRUE
## 4 4.5   K FALSE
```

```
subset(zd[c("zm1","zm3")],zm2=="K")
```

```
##   zm1   zm3
## 1 6.1  TRUE
## 3 1.7  TRUE
## 4 4.5 FALSE
```

```
kobiety$zm1
```

```
## [1] 6.1 1.7 4.5
```

```
kobiety$zm3
```

```
## [1]  TRUE  TRUE FALSE
```

```
#sortujemy rosnąco ze względu na zmienną zm1
ranking<-order(zd$zm1)
ranking
```

```
## [1] 3 2 4 1
```

```
zd[ranking,]
```

```
##   zm1 zm2   zm3
## 3 1.7   K  TRUE
## 2 2.2   M FALSE
## 4 4.5   K FALSE
## 1 6.1   K  TRUE
```

Odczyt/zapis: **write.table, write.csv, read.table, read.csv, scan, save, load, sprintf**

```
write.csv(zd,file="dane.csv")
```

Zadanie 1

Stworzyć zbiór danych **naukowcy** składający się z wektorów:

```
nazwisko <- c("Feynman", "Kosterlitz", "Haldane", "Thompson",
"Hardy", "Smale")
zawod <- c("fizyk", "fizyk", "fizyk","matematyk", "matematyk",
"matematyk")
rok_urodzenia<- c(1918, 1943, 1951, 1932, 1877, 1930)
nobel <- c(TRUE,TRUE,TRUE, FALSE, FALSE, FALSE)
```

1. Używając funkcji **str** sprawdzić strukturę zbioru **naukowcy**.
2. Wyświetlić wszystkie dane o Kosterlitzu.
3. Jaki zawód wykonuje Smale?
4. Wyświetlić naukowców, którzy urodzili się po roku 1939.
5. Wyświetlić tylko naukowców, którzy otrzymali nagrodę Nobla.
6. Posortować naukowców względem kolumny **rok_urodzenia**.
7. Zapisać rezultat do pliku *naukowcy.csv*

Instrukcje warunkowe, pętle i funkcje

If

- **if**(*cond*) *iftrue*

```
x<-5
if( x > 2 ){
  cat(x,"jest większe od 2") }
```

```
## 5 jest większe od 2
```

```
#funkcja cat służy do łączenia ze sobą łańcuchów znaków, wartości
#numerycznych etc. i wyświetlenia ich na ekranie.
cat("Wartość zmiennej x:",x)
```

```
## Wartość zmiennej x: 5
```

```
cat("a","b","c","d")
```

```
## a b c d
```

```
cat("a","b","c","d", sep=", ")
```

```
## a, b, c, d
```

```
cat("a","b","c","d", sep=" <-> ")
```

```
## a <-> b <-> c <-> d
```

- **if**(*cond*) *iftrue* **else** *iffalse*

```
x<-1
if( x > 2 ) {
  cat(x,"jest większe od 2") } else{
# else nie może być w nowej linii!!!
    cat(x,"nie jest większe niż 2") }
```

```
## 1 nie jest większe niż 2
```

Pętla for

for(*i* **in** *indeksy*) *expr*

indeksy nie muszą być liczbami, może to być „w zasadzie dowolny” wektor

```
x = c(0,1,2,3,4,5,6,7,8,9)
for(i in 1:10){
  x[i]=x[i]+2^i }
x
```

```
##  [1]    2    5   10   19   36   69  134  263  520 1033
```

```
x = c("krzesło",NA,"dąb", 3,"nic")
for(i in x){
  cat(paste("Element", i,"\n")) }
```

```
## Element krzesło
## Element NA
## Element dąb
## Element 3
## Element nic
```

Pętla while

while(*cond*) *expr* - pętla wykonuje wyrażenie *expr* dopóki warunek *cond* jest prawdą

```
i <- 0
while(i < 3) {
  cat(paste("juz", i, "\n"))
  i <- i+1 }
```

```
## juz 0
## juz 1
## juz 2
```

Definiowanie funkcji

- *nazwa_funkcji* = **function**(*argumenty*) *kod_funkcji* **return**(*value*)

```
#prosta funkcja f1, można pominąć return
f1 = function(x,y) x^2+y-1
#wywołanie funkcji f1
f1(2,3)
```

```
## [1] 6
```

```
f2 = function(x,n){
  var = 0 # zmienna lokalna
  for(i in rep(x,n)){
    var<-var+i}
  return(var) }

f2(2,10)
```

```
## [1] 20
```

Argumenty w definicji mogą przyjmować wartości domyślne:

```
f3 = function(x,y=0) x^2+y-1
f3(3)
```

```
## [1] 8
```

```r
f3(3,4)
```

```
## [1] 12
```

Inne polecenia warte uwagi:

Do odczytu informacji z konsoli (i nie tylko (help)) służy funkcja **scan**.

```r
{nazwa = scan()}
#  wczytuje kolejne wartości liczbowe do zmiennej
# nazwa, wczytanie pustej linii kończy wczytywanie
{nazwa = scan(what="",sep="\n")}
# wczytuje kolejne łańcuchy tekstowe
# do zmiennej nazwa, wczytanie pustej linii kończy wczytywanie
# nawiasy klamrowe umożliwiają wykonanie kilku funkci scan() naraz
```

Zadanie 2

1. Zapoznać się z funkcjami **pbirthday** i **qbirthday**, a następnie obliczyć prawdopodobieństwo, że dokładnie 2 spośród 30 osób urodziło się tego samego dnia.
2. Napisać skrypt, który prosi użytkownika o podanie liczby osób. Jeśli liczba podana przez użytkownika jest dodatnia, to oblicza i wyświetla na ekranie prawdopodobieństwo, że dokładnie 2 spośród tych osób urodziły się tego samego dnia.
3. (zadanie domowe) Napisać funkcję **prawdopodobienstwa(nmin=1,nmax=366,k=2,filename='file.txt')**, o domyślnych wartościach argumentów **nmin=1, nmax=366, k=2**, która:
   - Dla wszystkich **n = nmin, . . . , nmax** oblicza prawdopodobieństwo **p**, że dokładnie **k** spośród **n** osób urodziło się tego samego dnia.
   - Otrzymane wyniki umieszcza w tablicy
   - Zapisuje tą tablicę do pliku podanego jako ostatni argument **filename**.

Wywołać tę funkcję dwukrotnie: z argumentami domyślnymi oraz z własnymi