



Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

Programiranje v R-ju

4. Sestavljeni podatki

Vladimir Batagelj

Univerza v Ljubljani, FMF, Matematika

Finančna matematika
Ljubljana, februar 2009
19. november 2012



Kazalo

Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

- 1 Seznami
- 2 Razporeditve in matrike
- 3 Razpredelnice



Seznami

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Seznami so zaporedja podatkov. Podatki so lahko tudi različnih zvrsti in poimenovani. Seznam ustvarimo s funkcijo `list`

`list(n1=v1, n2=v2, ... nk=vk)`

kjer se n_i imena, v_i pa vrednosti. Posamezna imena (skupaj z enačajem) lahko tudi opustimo. Vrednosti so lahko poljubni podatki – tudi sezname.

Do i -tega podatka iz seznama s pridemo z izrazom `s[[i]]` ali, če je člen poimenovan, tudi z izrazom `s$ni` oziroma `s[["ni"]]`.

Z izrazi oblike `s[i]`, kjer so i indeksi ali imena, dobimo podsezname seznama s .

Funkcija `length(s)` vrne število členov v seznamu s ; funkcija `names(s)` pa imena členov. Sezname lahko stikamo s funkcijo `c`.

Seznami nam pridejo prav tudi v definicijah funkcij, kadar je potrebno vrniti več rezultatov. Zberemo jih v seznam

`return(list(n1=v1, n2=v2, ... nk=vk))`



Seznami

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

```
> s <- list(a=3,b="Ljubljana",TRUE)
> names(s)
[1] "a" "b" ""
> length(s)
[1] 3
> s
$a
[1] 3
$b
[1] "Ljubljana"
[[3]]
[1] TRUE
> s[[3]]
[1] TRUE
> s$b
[1] "Ljubljana"
> s[["a"]]
[1] 3
> s[2]
$b
[1] "Ljubljana"
> s[c('b','a')]
$b
[1] "Ljubljana"
$a
[1] 3
```

```
> s$b <- list("LJ",d=3,e=5)
> s
$a
[1] 3
$b
$b[[1]]
[1] "LJ"
$b$d
[1] 3
$b$e
[1] 5
[[3]]
[1] TRUE
> s$b[c(1,3)]
[[1]]
[1] "LJ"
$e
[1] 5
> names(s$b)
[1] "" "d" "e"
> names(s$b)[1] <- "f"
> names(s$b)
[1] "f" "d" "e"
> s$b$f
[1] "LJ"
```



Razporeditve

Sestavljeni
podatki

V. Batagelj

Seznam

Razporeditve
in matrike

Razpredelnice

Naj bo v vektor razsežnosti $d_1 \cdot d_2 \cdots d_k$. Tedaj prireditveni stavek
 $\text{dim}(v) \leftarrow c(d_1, d_2, \dots, d_k)$

predela vektor v v k razsežno *razporeditev* (array).

Enak učinek bi lahko dosegli tudi s funkcijo `array`
 $\text{array}(v, \text{dim}=c(d_1, d_2, \dots, d_k))$

Vektor se zloga v razporeditev najprej po stolpcih, ...

Iz razporeditve a dobimo nazaj vektor s funkcijo `as.vector(a)`.

Do posameznega člena razporeditve a pridemo tako, da navedemo
njegove "koordinate" – *indekse*

$a[i_1, i_2, \dots, i_k]$

Če na nekem mestu indeks izpustimo, to pomeni vse možne njegove
vrednosti.

Za indekse lahko uporabimo tudi poljubno indeksno razporeditev –
razporeditev razsežnosti $m \times k$ katere členi so indeksi.



Razporeditve

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

```

> a <- 1:24
> dim(a) <- c(3,2,4)
> class(a)
[1] "array"
> dim(a)
[1] 3 2 4
> a
, , 1
  [,1] [,2]
[1,]  1   4
[2,]  2   5
[3,]  3   6
, , 2
  [,1] [,2]
[1,]  7  10
[2,]  8  11
[3,]  9  12
, , 3
  [,1] [,2]
[1,] 13  16
[2,] 14  17
[3,] 15  18
, , 4
  [,1] [,2]
[1,] 19  22
[2,] 20  23
[3,] 21  24

```

```

> a[2,2,3]
[1] 17
> a[3,2,]
[1] 6 12 18 24
> a[, ,4]
  [,1] [,2]
[1,]  19  22
[2,]  20  23
[3,]  21  24
> p <- trunc(runif(24,1,10))
> b <- array(p,dim=c(3,2,4))
> class(b)
[1] "array"
> as.vector(b)
[1] 4 9 3 2 2 6 7 6 4 1 8 3 2 7 6 6
> j <- 1:5
> i <- array(c(j,j),dim=c(5,2))
> d <- array(0,dim=c(5,5))
> d[i] <- j^2
> d
  [,1] [,2] [,3] [,4] [,5]
[1,]  1   0   0   0   0
[2,]  0   4   0   0   0
[3,]  0   0   9   0   0
[4,]  0   0   0  16   0
[5,]  0   0   0   0  25

```



Razporeditve

Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

Funkcija $\text{aperm}(a, p)$ preuredi razsežnosti razporeditve a tako kot določa permutacija p . Na primer

```
ap <- aperm(a, c(2, 3, 1))
```

V dvorazsežnem primeru temu ustreza transpozicija $t(a)$. Velja $t(a) \equiv \text{aperm}(a, c(2, 1))$

Za računanje (običajne operacije in funkcije) veljajo za razporeditve podobna pravila kot za vektorje – računanje poteka po (istoležnih) členih.

Funkcija $\text{outer}(a, b, \text{FUN}=f, \dots)$ določi **zunanji produkt** razporeditev (vektorjev) a in b . Ta je nova razporeditev A razsežnosti $c(\dim(a), \dim(b))$ in s členi $A[c(ia, ib)] = f(a[ia], b[ib], \dots)$.

V posebnem primeru, ko je funkcija množenje, lahko uporabimo okrajšavo

```
a %o% b ≡ outer(a, b, "*")
```



Razporeditve

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Ustvarimo tabelo operacije $x \circ y \equiv (x \cdot y) \bmod m$

```
> mprod <- function(a,b,m)
+   return((a*b)%m)
> u <- 0:6
> names(u) <- u
> outer(u,u,mprod,m=7)
  0 1 2 3 4 5 6
0 0 0 0 0 0 0
1 0 1 2 3 4 5 6
2 0 2 4 6 1 3 5
3 0 3 6 2 5 1 4
4 0 4 1 5 2 6 3
5 0 5 3 1 6 4 2
6 0 6 5 4 3 2 1
>
```




Matrike

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Dvorazsežnim razporeditvam pravimo *matrike*. Zanje obstaja posebna funkcija `matrix`

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
```

S funkcijama `nrow(A)` in `ncol(A)` izvemo število vrstic in število stolpcev matrike A .

Vrstice/stolpci matrike so v bistvu vektorji. Funkciji `rbind` in `cbind` omogočata zlaganje enako dolgih vrstic/stolpcev v matriko.

Funkcija `det(A)` določi vrednost determinante matrike A .

Operacija $A \%*\% B$ ustreza matričnemu produktu. Matriki A in B morata biti usklajeni – `ncol(A) = nrow(B)`. Pri množenju vektorjev in matrik R pretvori vektor v ustrezno matriko. Zato kvadratno formo $x^T A x$ zapišemo `x \%*\% A \%*\% x`.

Na voljo je tudi funkcija `crossprod(A,B) ≡ t(A) \%*\% B`.

Tako lahko skalarni produkt dveh vektorjev zapišemo na več načinov `crossprod(u,v) = rbind(u) \%*\% cbind(v) = u \%*\% v`.



Matrike

Sestavljeni podatki

V. Batagelj

Seznam

Razporeditve in matrike

Razpredelnice

```
> (A <- matrix(1:12,nrow=3))
  [,1] [,2] [,3] [,4]
[1,]   1   4   7  10
[2,]   2   5   8  11
[3,]   3   6   9  12
> (B <- matrix(1:12,nrow=3,byrow=TRUE))
  [,1] [,2] [,3] [,4]
[1,]   1   2   3   4
[2,]   5   6   7   8
[3,]   9  10  11  12
> rownames(A) <- c('a','b','c')
> colnames(A) <- paste('x',1:4,sep="")
> A
  x1 x2 x3 x4
a  1  4  7 10
b  2  5  8 11
c  3  6  9 12
> A['b',]
x1 x2 x3 x4
2  5  8 11
> u <- trunc(runif(4,1,10))
> v <- trunc(runif(4,1,10))
> z <- trunc(runif(4,1,10))
> w <- trunc(runif(4,1,10))
> C <- rbind(u,v,z)
> nrow(C)
[1] 3
```

```
> ncol(C)
[1] 4
> dim(C)
[1] 3 4
> class(C)
[1] "matrix"
> C
  [,1] [,2] [,3] [,4]
u     8     9     3     2
v     5     6     3     1
z     2     3     6     4
> D <- rbind(C,w)
> colnames(D) <- rownames(D)
> D
  u v z w
u 8 9 3 2
v 5 6 3 1
z 2 3 6 4
w 6 5 5 8
> t(D)
  u v z w
u 8 5 2 6
v 9 6 3 5
z 3 3 6 5
w 2 1 4 8
> det(D)
[1] -48
```



Matrike

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

```
> D
  u v z w
u 8 9 3 2
v 5 6 3 1
z 2 3 6 4
w 6 5 5 8
> diag(D)
  u v z w
8 6 6 8
> diag(D) <- -(1:4)
> D
  u v z w
u -1 9 3 2
v 5 -2 3 1
z 2 3 -3 4
w 6 5 5 -4
> (E <- diag(4))
      [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
>
```

```
> D[2,]
u v z w
5 6 3 1
> D[2,,drop=FALSE]
  u v z w
v 5 6 3 1
> as.vector(D[2,])
[1] 5 6 3 1
```



Matrike – reševanje sistemov linearnih enačb

Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

Rešitev x sistema linearnih enačb $Ax = b$, kjer je A kvadratna matrika in b vektor, nam vrne funkcija `solve`

```
x <- solve(A,b)
```

Izraz `solve(A)` vrne obratno matriko A^{-1} matrike A .

Kvadratno formo $x^T A^{-1} x$ izračunamo z izrazom

```
x %*% solve(A,x).
```

Funkcija `solve` temelji na razcepu LU matrike A . Matriko A je mogoče zapisati v obliki $A = LU$, kjer je L spodnje trikotna matrika z enotsko diagonalo in U zgornje trikotna matrika z diagonalo. Enačbo $Ax = b$ lahko sedaj rešimo v dveh korakih $Ly = b$ in $Ux = y$.



Matrike – reševanje sistemov linearnih enačb

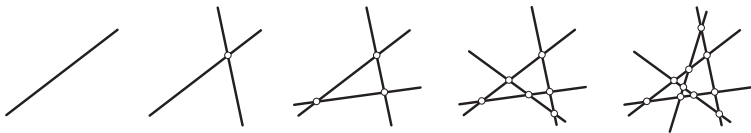
Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice



Določimo obrazec, ki mu zadoščajo členi zaporedja a_k števila presečišč k premic v splošni legi: $a_1 = 0$, $a_2 = 1$, $a_3 = 3$, $a_4 = 6$, $a_5 = 10$, ...

Recimo, da je $a_k = P(k)$, kjer je $P(k)$ polinom. Če je $P(k)$ stopnje n , je $P(k+1) - P(k)$ stopnje $n-1$. Torej je za naše zaporedje druge stopnje

$$a_k = Ak^2 + Bk + C$$

Od tu dobimo (metoda nedoločenih koeficientov) sistem

$$a_1 = A1^2 + B1 + C = 0$$

$$a_2 = A2^2 + B2 + C = 1$$

$$a_3 = A3^2 + B3 + C = 3$$



Matrike – reševanje sistemov linearnih enačb

Sestavljeni podatki

V. Batagelj

Seznam

Razporeditve in matrike

Razpredelnice

```
> vr <- function(k) c(k^2,k,1)
> vr(3)
[1] 9 3 1
> (a <- rbind(vr(1),vr(2),vr(3)))
      [,1] [,2] [,3]
[1,]    1    1    1
[2,]    4    2    1
[3,]    9    3    1
> b <- c(0,1,3)
> (x <- solve(a,b))
[1] 5.000000e-01 -5.000000e-01 -1.665335e-16
> x <- c(1/2,-1/2,0)
> vr(4) %*% x
      [,1]
[1,]    6
> vr(5) %*% x
      [,1]
[1,]   10
> vr(10) %*% x
      [,1]
[1,]   45
> solve(a)
      [,1] [,2] [,3]
[1,]  0.5  -1  0.5
[2,] -2.5   4 -1.5
[3,]  3.0  -3  1.0
```

```
> s <- 1:3
> cbind(s^2,s,1)
      s
[1,]  1  1  1
[2,]  4  2  1
[3,]  9  3  1
>
```



Trikotne matrice

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrice

Razpredelnice

Funkciji `lower.tri(A)` in `upper.tri(A)` ustvarita indeksni matriki za spodnji/zgornji trikotnik matrice A . Če dodamo argument `diag=TRUE`, je v matriko vključena tudi diagonala.

```
> A <- matrix(1:25,byrow=TRUE,nrow=5)
> B <- A
> B[upper.tri(A,diag=TRUE)] <- 0
> B
      [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    0
[2,]    6    0    0    0    0
[3,]   11   12    0    0    0
[4,]   16   17   18    0    0
[5,]   21   22   23   24    0
> A[lower.tri(A)]
[1]  6 11 16 21 12 17 22 18 23 24
```



Razcep QR

Sestavljeni
podatki

V. Batagelj

Seznam

Razporeditve
in matrice

Razpredelnice

Pri **razcepu QR** matriko A reda $n \times m$ zapišemo v obliki $A = QR$, kjer je Q ortogonalna, $Q^T Q = I$, matrika reda $n \times n$ in R zgornje trikotna matrika reda $n \times m$. Enačbo $Ax = b$ lahko nadomestimo z enakovredno $Rx = Q^T b$.

Razcep QR dobimo s funkcijo `qr(A)`. Iz razcepa `qr` dobimo matriko Q s funkcijo `qr.Q(qr)`, matriko R pa s funkcijo `qr.R(qr, complete=TRUE)`. Brez dodatnega argumenta funkcija `qr.R` vrne kvadratno matriko reda $n \times n$.

Funkcija `qr.solve(A, b)` reši sistem $Ax = b$ z uporabo razcepa QR.

Obstaja še nekaj funkcij `qr.*`, ki se uporabljajo predvsem pri reševanju optimizacijskih nalog.



Razcep Choleskega

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Razcep Choleskega izrazi nenegativno definitno simetrično matriko A v obliki $A = U^T U$, kjer je U zgornje trikotna matrika. Matriko U nam vrne funkcija `chol(A)`.

Rešitev x zgornje trikotnega sistema $Ux = b$ nam da funkcija `backsolve(U, b)`; za spodnje trikotni sistem $Lx = b$ pa funkcija `forwardsolve(L, b)`. Rešitev sistema $Ax = b$ torej dobimo z

```
> U <- chol(A)
> x <- backsolve(U, forwardsolve(t(U), b))
```

Obratno matriko A^{-1} dobimo s funkcijo `chol2inv(U)`.



Lastne vrednosti in vektorji

Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

Število $\lambda \in \mathbb{C}$ imenujemo lastna vrednost matrike A , če obstaja tak neničelni vektor x , da je $Ax = \lambda x$. Vektor x imenujemo lastni vektor za A pri lastni vrednosti λ .

V R-ju dobimo lastne vrednosti in pripadajoče lastne vektorje s funkcijo `eigen(A)`, ki nam vrne seznam s členoma `values` – vektor lastnih vrednosti, in `vectors` – matrika lastnih vektorjev (stolpci).

Če potrebujemo samo lastne vrednosti, dodamo argument `only.values=TRUE`.



Razcep SVD

Sestavljeni
podatki

V. Batagelj

Seznami

Razporeditve
in matrike

Razpredelnice

Matriko A reda $n \times m$ lahko zapišemo tudi v obliki $A = UDV^T$, kjer D diagonalna matrika, U in V redov $n \times \min(n, m)$ in $m \times \min(n, m)$ pa sta matriki z ortonormalnimi stolpci – $U^{-1} = U^T$ in $V^{-1} = V^T$.

Ker je $A^T A = V^{-1} D^2 V$, je D^2 podobna $A^T A$ – kvadrati d_i^2 so lastne vrednosti $A^T A$.

Funkcija $\text{svd}(A)$ vrne seznam s člani: vektor d , ter matriki u in v .



Še nekaj matričnih funkcij

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Funkcija $\kappa(A)$ vrne pogojenostno število matrike A .

Funkcija $\text{kronecker}(A, B)$ določi posplošeni Kroneckerjev produkt matrik A in B .

Funkcija $\text{apply}(A, i, f)$ uporabi funkcijo f na vrsticah ($i = 1$) / stolpcih ($i = 2$) dane matrike A . Na primer

```
> A <- matrix(1:24,nrow=4)
> A
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    5    9   13   17   21
[2,]    2    6   10   14   18   22
[3,]    3    7   11   15   19   23
[4,]    4    8   12   16   20   24
> apply(A,1,sum)
[1] 66 72 78 84
> apply(A,2,sum)
[1] 10 26 42 58 74 90
```

Podobni funkciji za sezname sta lapply in sapply .

$\text{lapply}(s, f, \dots)$ vrne seznam katerega člani so vrednosti, ki jih dobimo z uporabo funkcije f nad ustreznim členom seznama s . Pri funkciji $\text{sapply}(s, f, \dots)$ lahko uporabimo še določilo $\text{simplify} = \text{TRUE}$, ki vrne vektor ali matriko.



Razpredelnice

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Razpredelnice so najpogostejša oblika podatkov v analizi podatkov. V R-ju jih podpira struktura `data.frame`, ki je v bistvu posebne vrste seznam, v katerem so vsi členi (stolpci/spremenljivke) s_i enako dolgi vektorji. Imena stolpcev se ustvarijo samodejno.

```
R <- data.frame(s1, s2, ..., sk, row.names=enote)
```

Do i -te spremenljivke pridemo z izrazom `R$si` ali `R$[[i]]`.

Ime razpredelnice lahko izpustimo, če pred tem uporabimo funkcijo `attach(R)`, ki doda razpredelnico v seznam okolij po katerih R išče imena spremenljivk. Seznam dobimo s funkcijo `search()`. Razpredelnico odpnemo s funkcijo `detach(R)`.

Razpredelnico R lahko ustvarimo tudi interaktivno s stavkom

```
R <- edit(data.frame())
```

Trenutno delovno področje nam pove `getwd()`; spremenimo pa ga s `setwd(pot)`, npr. `setwd("c:/Users/Batagelj/test/R")`. Uporabimo lahko tudi možnost `Change dir ...` v izbiri File.

Vrednosti izbranih spremenljivk shranimo na datoteko `podatki.R` (v delovnem področju) z zahtevo `dump(c("v1", "v2", ..., "vk"), "podatki.R")`. Ko jih ponovno potrebujemo, jih lahko vnesemo s `source("podatki.R")`.



Razpredelnice – vgrajeni podatki

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

Podobne učinke imata tudi funkciji `save` in `load`.

Če razpredelnico pripravimo v Excelu, jo shranimo s `File/Save as ... in v Save as Type` izberemo `Text (Tab delimited)`. Excel sam doda podaljšek `.txt`. V R lahko razpredelnico vnesemo z

```
R <- read.table("datoteka",header=TRUE)
```

Posebna različica `read.csv` je pripravljena za branje datotek `.csv` (comma separated values).

Da bodo stvari tekle gladko, v razpredelnici ne uporabljamo imen, ki vsebujejo presledke. Nadomestimo jih s podčrtajem ali piko ali sploh opustimo. Seznam spremenljivk v razpredelnici dobimo z `names(R)`. V R-jevih paketih najdemo veliko že pripravljenih testnih razpredelnic.

Njihov seznam dobimo z zahtevo

```
data(package=.packages(all.available=TRUE))
```

Izbrano razpredelnico `R` naložimo z zahtevo `data(R)` in si njen začetek ogledamo z zahtevo `head(R)`.



Razpredelnice – vgrajeni podatki

Sestavljeni
podatki

V. Batagelj

Seznam

Razporeditve
in matrike

Razpredelnice

S funkcijo `subset(R, pogoj)` lahko dobimo iz razpredelnice R podrazpredelnico sestavljeno iz vrstic, ki zadoščajo danemu *pogoj*. Na razpredelnico R lahko pogledamo tudi kot na matriko z (morda) raznovrstnimi vrednostmi. To nam omogoča indeksno doseganje njenih sestavin:

$R[i,]$ – opis i -te enote

$R[, j]$ – j -ta spremenljivka

Uporabimo lahko vse, kar smo z indeksi vajeni početi. Na primer, z $R[pogoj,]$ izberemo samo tiste enote, ki zadoščajo *pogoj*; z $R[, c("s_1", "...", "s_k")]$ pa samo izbrane spremenljivke. Z izrazom $R[, sapply(R, is.numeric)]$ izberemo iz razpredelnice vse številske spremenljivke.

Z $R[order("s_1", "...", "s_k"),]$ preuredimo razpredelnico glede na vrednosti izbranih spremenljivk. Če želimo obratni vrstni red, nad `order` uporabimo še funkcijo `rev`.

S `summary(R)` dobimo osnovne značilnosti spremenljivk razpredelnice R .



Razpredelnice – primeri

Sestavljeni podatki

V. Batagelj

Seznam

Razporeditve in matrike

Razpredelnice

```
> data(package=.packages(all.available=TRUE))
```

```
> Orange
```

```
  Tree  age  circumference
```

```
1      1  118             30
2      1  484             58
3      1  664             87
4      1 1004            115
```

```
.....
```

```
32     5 1004            125
```

```
33     5 1231            142
```

```
34     5 1372            174
```

```
35     5 1582            177
```

```
> (b <- subset(Orange,age<1000))
```

```
  Tree  age  circumference
```

```
1      1  118             30
```

```
2      1  484             58
```

```
3      1  664             87
```

```
8      2  118             33
```

```
.....
```

```
31     5  664             81
```

```
> data(animals,package="cluster")
```

```
> animals
```

```
  war fly ver end gro hai
```

```
ant  1  1  1  1  2  1
```

```
bee  1  2  1  1  2  2
```

```
.....
```

```
spi  1  1  1  NA  1  2
```

```
wha  2  1  2  2  2  1
```

```
> class(animals)
```

```
[1] "data.frame"
```

```
> names(animals)
```

```
[1] "war" "fly" "ver" "end" "gro" "hai"
```

```
> row.names(animals)
```

```
[1] "ant" "bee" "cat" "cpl" "chi" "c"
```

```
[7] "duc" "eag" "ele" "fly" "fro" "h"
```

```
[13] "lio" "liz" "lob" "man" "rab" "s"
```

```
[19] "spi" "wha"
```

```
> animals$war
```

```
[1] 1 1 2 1 2 2 2 2 1 1 1 2 1 1 2
```

```
> attributes(animals)
```

```
$names
```

```
[1] "war" "fly" "ver" "end" "gro" "hai"
```

```
$class
```

```
[1] "data.frame"
```

```
$row.names
```

```
[1] "ant" "bee" "cat" "cpl" "chi" "c"
```

```
.....
```

```
> war
```

```
Error: object "war" not found
```

```
> attach(animals)
```

```
> war
```

```
[1] 1 1 2 1 2 2 2 2 1 1 1 2 1 1 2
```




Razpredelnice – primeri

Sestavljeni podatki

V. Batagelj

Seznam

Razporeditve in matrike

Razpredelnice

```

> animals[3,]
  war fly ver end gro hai
cat  2  1  2  1  1  2
> animals[,1]
[1] 1 1 2 1 2 2 2 2 2 1 1 1 2 1 1 2...
> animals[4,1]
[1] 1
> animals[20,1]
[1] 2
> animals[animals$war==2,]
  war fly ver end gro hai
cat  2  1  2  1  1  2
chi  2  1  2  2  2  2
cow  2  1  2  1  2  2
duc  2  2  2  1  2  1
eag  2  2  2  2  1  1
ele  2  1  2  2  2  1
lio  2  1  2  NA  2  2
man  2  1  2  2  2  2
rab  2  1  2  1  2  2
wha  2  1  2  2  2  1
> animals[animals$war==2,c(2,3)]
  fly ver
cat  1  2
chi  1  2
...
wha  1  2
> animals[animals$war==2,c("fly","ver")]
> attach(animals)

```

```

> animals[war==2,c("fly","ver")]
> na.omit(animals)
  war fly ver end gro hai
ant  1  1  1  1  2  1
bee  1  2  1  1  2  2
cat  2  1  2  1  1  2
cpl  1  1  1  1  1  2
chi  2  1  2  2  2  2
cow  2  1  2  1  2  2
duc  2  2  2  1  2  1
eag  2  2  2  2  1  1
ele  2  1  2  2  2  1
fly  1  2  1  1  1  1
her  1  1  2  1  2  1
liz  1  1  2  1  1  1
man  2  1  2  2  2  2
rab  2  1  2  1  2  2
wha  2  1  2  2  2  1
> complete.cases(animals)
[1] TRUE TRUE TRUE TRUE T
[13] FALSE TRUE FALSE TRUE T
> animals[complete.cases(animals)
  war fly ver end gro hai
ant  1  1  1  1  2  1
...
liz  1  1  2  1  1  1
man  2  1  2  2  2  2
rab  2  1  2  1  2  2
wha  2  1  2  2  2  1

```



Razpredelnice – primeri

Sestavljeni podatki

V. Batagelj

Seznami

Razporeditve in matrike

Razpredelnice

```
> help(Orange)
> Orange
  Tree age circumference
1     1  118             30
2     1  484             58
3     1  664             87
4     1 1004            115
5     1 1231            120
6     1 1372            142
7     1 1582            145
8     2  118             33
...
33    5 1231            142
34    5 1372            174
35    5 1582            177
> b <- Orange[22:35,]
> b
  Tree age circumference
22    4  118             32
23    4  484             62
24    4  664            112
...
34    5 1372            174
35    5 1582            177
> attach(b)
```

```
> b[order(age),]
  Tree age circumference
22    4  118             32
29    5  118             30
23    4  484             62
...
34    5 1372            174
28    4 1582            214
35    5 1582            177
> b[order(age, circumference),]
  Tree age circumference
29    5  118             30
22    4  118             32
30    5  484             49
...
27    4 1372            209
35    5 1582            177
28    4 1582            214
> b[rev(order(age, circumference)),]
  Tree age circumference
28    4 1582            214
35    5 1582            177
27    4 1372            209
...
30    5  484             49
22    4  118             32
29    5  118             30
```



Razpredelnice – primeri

Sestavljeni podatki

V. Batagelj

Seznam

Razporeditve in matrike

Razpredelnice

```
> apply(b[c(2,3)],2,sum)
      age circumference
12910                1753
```

```
> animals
      war fly ver end gro hai
ant   1   1   1   1   2   1
bee   1   2   1   1   2   2
cat   2   1   2   1   1   2
cpl   1   1   1   1   1   2
chi   2   1   2   2   2   2
cow   2   1   2   1   2   2
duc   2   2   2   1   2   1
eag   2   2   2   2   1   1
ele   2   1   2   2   2   1
fly   1   2   1   1   1   1
fro   1   1   2   2   NA  1
her   1   1   2   1   2   1
lio   2   1   2   NA  2   2
liz   1   1   2   1   1   1
lob   1   1   1   1   NA  1
man   2   1   2   2   2   2
rab   2   1   2   1   2   2
sal   1   1   2   1   NA  1
spi   1   1   1   NA  1   2
wha   2   1   2   2   2   1
```

```
> duplicated(animals)
[1] FALSE FALSE FALSE FALSE FALSE ...
[15] FALSE TRUE TRUE FALSE FALSE TR
```

```
> animals[duplicated(animals),]
      war fly ver end gro hai
man   2   1   2   2   2   2
rab   2   1   2   1   2   2
wha   2   1   2   2   2   1
```

```
> unique(animals)
      war fly ver end gro hai
ant   1   1   1   1   2   1
...
liz   1   1   2   1   1   1
lob   1   1   1   1   NA  1
sal   1   1   2   1   NA  1
spi   1   1   1   NA  1   2
```