

R

Osztályok típusai és a kiválasztás

Csima Judit

BME, VIK,
Számítástudományi és Informatikai Tanszék

2013. február 14.

Alaptípusok

- minden adat, amivel dolgozunk egy objektum
- minden objektumnak van típusa, ettől függ, hogy mit lehet vele csinálni
- atomi típusok: character, numeric, integer, complex, logical
- ezekből felépíthető
 - vector: azonos típusú objektumokat tartalmaz (közös nevezőre hozás, ha lehet)
 - list: tartalmazhat különböző típusú objektumokat
 - matrix
 - data frame

Attribútumok

az objektumoknak lehetnek attribútumaik

- names, dimnames
- dimenzió (pl. mátrix, data frame)
- class
- length

lekérdezhető az `attributes()` függvénnyel

```
> x <- 1                                ##értékadás, nincs kiírás
> x                                     ##auto-printing
[1] 1
> class(x)
[1] "numeric"

> t <- 1:10
> t
[1] 1 2 3 4 5 6 7 8 9 10
> class(t)
[1] "integer"
```

```
> y <- rnorm(13)

> print(y)          ##explicit printing
[1] 1.93863126 -0.93498813 -0.87160105 0.27556586
-1.23191483 0.11921757 0.32511604 0.39737275
0.04851825 -0.82997406
[11] -1.21575261 -0.96806903 -0.91578041

> class(y)          ##milyen atomi típusú?
[1] "numeric"

> str(y)            ##összefoglalás
num [1:13] 1.939 -0.935 -0.872 0.276 -1.232 ...
```

```
> z <- airquality          ##beépített data frame
```

```
> head(z)
```

| | Ozone | Solar.R | Wind | Temp | Month | Day |
|---|-------|---------|------|------|-------|-----|
| 1 | 41 | 190 | 7.4 | 67 | 5 | 1 |
| 2 | 36 | 118 | 8.0 | 72 | 5 | 2 |
| 3 | 12 | 149 | 12.6 | 74 | 5 | 3 |
| 4 | 18 | 313 | 11.5 | 62 | 5 | 4 |
| 5 | NA | NA | 14.3 | 56 | 5 | 5 |
| 6 | 28 | NA | 14.9 | 66 | 5 | 6 |

Vektorok

A `c()` függvénnyel vektorokat lehet csinálni

```
> x<- c(0.5, 0.6)
```

Ha olyan elemek kerülnek a vektorba, amik nem azonos típusuak, akkor az R egységes típusra konvertálja a tagokat: $logical \subseteq integer \subseteq numeric \subseteq complex \subseteq character$

Explicit módon is lehet típust váltani az `as.*` függvényekkel:

```
> x<- 1:6
```

```
> class(x)
```

```
[1] "integer"
```

```
> as.character(x)
```

```
[1] "1" "2" "3" "4" "5" "6"
```

Hiányzó értékek

- jelölése: NA (vagy NaN)
- hiányzó érték vagy ha olyan műveletet hajtunk végre, ami nem teljesen értelmes:

```
> x <- "aa"  
> as.logical(x)  
[1] NA
```

- NaN: NA speciális esete, "not a number"
- lekérdezhető az `is.na()` és `is.nan()` függvényekkel


```
> x <- c(1,2,NA,10,3)
> is.na(x)
[1] FALSE FALSE TRUE FALSE FALSE
> !is.na(x)
[1] TRUE TRUE FALSE TRUE TRUE
> is.nan(x)
[1] FALSE FALSE FALSE FALSE FALSE

> x <- c(1, 2, NaN, NA, 4)
> is.na(x)
[1] FALSE FALSE TRUE TRUE FALSE
> is.nan(x)
[1] FALSE FALSE TRUE FALSE FALSE
```

Mátrixok

A mátrixok olyan vektorok, amiknek van egy *dimension* nevű attribútumuk, a *dimension* egy kettő hosszú vektor: (nrow, ncol)

```
> x <- rpois(10, 1)
```

```
> x
```

```
[1] 0 0 2 1 0 2 0 4 1 0
```

```
> ma <- matrix(x, 2, 5)
```

```
> ma
```

| | [,1] | [,2] | [,3] | [,4] | [,5] |
|------|------|------|------|------|------|
| [1,] | 0 | 2 | 0 | 0 | 1 |
| [2,] | 0 | 1 | 2 | 4 | 0 |

Mátrixok még...

Lehet oszloponként vagy soronként is feltölteni:

```
> x <- 1:3
> y <- 13:11
> cbind(x,y)
      x  y
[1,]  1 13
[2,]  2 12
[3,]  3 11
```

Hasonlóan `rbind()`

Listák

A lista olyan speciális vektor, ahol a tagok lehetnek különböző típusúak

```
> x <- list(1, "aa", TRUE, 1+2i)
```

```
> x
```

```
[[1]]
```

```
[1] 1
```

```
[[2]]
```

```
[1] "aa"
```

```
[[3]]
```

```
[1] TRUE
```

```
[[4]]
```

```
[1] 1+2i
```

Data frame

- Data frame: táblázat, belső reprezentációja olyan speciális lista, aminek tagjai ugyanolyan hosszú vektorok
- A lista tagjai az oszlopok, a tagok közös hossza a sorok száma
- Lehet különböző az oszlopok típusa
- Általában a sorok az egyes események (rekordok), az oszlopok pedig a rekordok tulajdonságainak felelnek meg

Data frame létrehozása

```
> df <- data.frame(ID = 1:3, Acc = c("Y", "Y", "N"))
```

```
> df
```

| | ID | Acc |
|---|----|-----|
| 1 | 1 | Y |
| 2 | 2 | Y |
| 3 | 3 | N |

Kiválasztás, három alapeset

- `[]`: ugyanolyan osztályú dolgot ad vissza, mint amiből kiválasztottam; lehet több elemet is választani egyszerre
- `[[]]`: listából vagy data frame-ből választ ki, az eredmény nem feltétlenül lista vagy data frame; csak egy elemet lehet kiválasztani
- `$`: kábé mint a `[[]]`, név alapján történő kiválasztás

Kiválasztás vektorból

```
> x <- c("a", "b", "c", "c", "d", "a")
> x[1]
[1] "a"
> x[2]
[1] "b"
> x[1:4]
[1] "a" "b" "c" "c"
> x[x > "a"]
[1] "b" "c" "c" "d"
> u <- x > "a"
> u
[1] FALSE TRUE TRUE TRUE TRUE FALSE
> x[u]
[1] "b" "c" "c" "d"
```


Kiválasztás mátrixból és data frame-ből

Mátrixokból egy elemet az (i, j) indexpár megadásával lehet kiválasztani.

```
> x <- matrix(1:6, 2, 3)
```

```
> x[1, 2]
```

```
[1] 3
```

```
> x[2, 1]
```

```
[1] 2
```

Lehet sorokat vagy oszlopokat is kiválasztani:

```
> x[1, ]
```

```
[1] 1 3 5
```

```
> x[, 2]
```

```
[1] 3 4
```

Data frame ugyanígy

Kiválasztás listából

```
> x <- list(lista1 = 1:4, szam = 0.6)
> x[1]
$lista1
[1] 1 2 3 4
> x[[1]]
[1] 1 2 3 4
> x$szam
[1] 0.6
> x[["szam"]]
[1] 0.6
> x["szam"]
$szam
[1] 0.6
```

Hiányos adatok kizárása

- Sokszor kell, hogy csak azok a sorok maradjanak meg, ahol minden érték megvan
- Erre való a `complete.cases` függvény:

```
> good <- complete.cases(airquality)
> class(good)
[1] "logical"
> head(good)
[1] TRUE TRUE TRUE TRUE FALSE FALSE
```

```
> airquality[1:6, ]
```

| | Ozone | Solar.R | Wind | Temp | Month | Day |
|---|-------|---------|------|------|-------|-----|
| 1 | 41 | 190 | 7.4 | 67 | 5 | 1 |
| 2 | 36 | 118 | 8.0 | 72 | 5 | 2 |
| 3 | 12 | 149 | 12.6 | 74 | 5c | 3 |
| 4 | 18 | 313 | 11.5 | 62 | 5 | 4 |
| 5 | NA | NA | 14.3 | 56 | 5 | 5 |
| 6 | 28 | NA | 14.9 | 66 | 5 | 6 |

```
> airquality[good, ][1:6, ]
```

| | Ozone | Solar.R | Wind | Temp | Month | Day |
|---|-------|---------|------|------|-------|-----|
| 1 | 41 | 190 | 7.4 | 67 | 5 | 1 |
| 2 | 36 | 118 | 8.0 | 72 | 5 | 2 |
| 3 | 12 | 149 | 12.6 | 74 | 5 | 3 |
| 4 | 18 | 313 | 11.5 | 62 | 5 | 4 |
| 7 | 23 | 299 | 8.6 | 65 | 5 | 7 |
| 8 | 19 | 99 | 13.8 | 59 | 5 | 8 |