

Laborator 03

Petculescu Mihai-Silviu

Laborator 03

Petculescu Mihai-Silviu

Liste

Creare liste prin vector()

Accesare prin index

Adăugare în liste

Ștergerea din listă

Concatenare liste

Data frame-uri

Metode de indexare

Metode de manipulare

Funcția order

Funcția merge

Funcția aggregate

Aplicație

Liste

```
> a1 = list(nume = "Ione1", salariu = 1500, apartenenta = T)
> a2 = list("Ione1", 1500, T)

> a1
$nume
[1] "Ione1"
$salariu
[1] 1500
$apartenenta
[1] TRUE

> a2
[[1]]
[1] "Ione1"

[[2]]
[1] 1500

[[3]]
[1] TRUE

# structura listei
> str(a1)
List of 3
 $ nume      : chr "Ione1"
 $ salariu   : num 1500
 $ apartenenta: logi TRUE
> str(a2)
List of 3
 $ : chr "Ione1"
```

```
$ : num 1500
$ : logi TRUE
```

Creare liste prin vector()

```
> z = vector(mode="list")
> z
list()

> z[["a"]] = 3
> z
$a
[1] 3
```

Accesare prin index

```
> a1[1] # []
$nume
[1] "Ionel"

> a1[[1]] # [[]]
# a1$nume
[1] "Ionel"

> str(a1[1]) # []
List of 1
 $ nume: chr "Ionel"

> str(a1[[1]]) # [[]]
# str(a1$nume)
chr "Ionel"
```

Adăugare în liste

```
# Prin folosirea numelui componenteii
> z = list(a = "abc", b = 111, c = c(TRUE, FALSE))
> z$d = "un nou element"
> z
$a [1] "abc"
$b [1] 111
$c [1] TRUE FALSE
$d [1] "un nou element"

# Prin indexare vectoriala
> z[[5]] = 200
> z[6:7] = c("unu", "doi")
> z
# (...)
[[5]] [1] 200
[[6]] [1] "unu"
[[7]] [1] "doi"
```

Ștergerea din listă

```
# atribuim valoarea NULL
> z[4] = NULL
> z
$a [1] "abc"
$b [1] 111
$c [1] TRUE FALSE
[[4]] [1] 200
[[5]] [1] "unu"
[[6]] [1] "doi"
```

Concatenare liste

```
# prin functia c()
> l1 = list(1:10, matrix(1:6, ncol = 3), c(T, F))
> l2 = list(c("Ionel", "Maria"), seq(1,10,2))
> l3 = c(l1, l2)
> length(l3)
[1] 5
> str(l3)
List of 5
 $ : int [1:10] 1 2 3 4 5 6 7 8 9 10
 $ : int [1:2, 1:3] 1 2 3 4 5 6
 $ : logi [1:2] TRUE FALSE
 $ : chr [1:2] "Ionel" "Maria"
 $ : num [1:5] 1 3 5 7 9
```

Data frame-uri

```
> survey = data.frame( "index" = c(1, 2, 3, 4, 5), "sex" = c("m", "m", "m", "f",
"f"), "age" = c(99, 46, 23, 54, 23) )
> survey
  index sex age
1     1  m  99
2     2  m  46
3     3  m  23
4     4  f  54
5     5  f  23
```

```
# Structura initiala
> str(survey)
'data.frame': 5 obs. of 3 variables:
 $ index: num 1 2 3 4 5
 $ sex : Factor w/ 2 levels "f","m": 2 2 2 1 1
 $ age : num 99 46 23 54 23

> survey = data.frame("index" = c(1, 2, 3, 4, 5),
                      "sex" = c("m", "m", "m", "f", "f"),
                      "age" = c(99, 46, 23, 54, 23),
                      stringsAsFactors = FALSE)

# Structura de dupa
> str(survey)
'data.frame': 5 obs. of 3 variables:
```

```
$ index: num 1 2 3 4 5
$ sex : chr "m" "m" "m" "f" ...
$ age : num 99 46 23 54 23
```

```
> data() # vedem ce seturi de date exista
> ?mtcars # alegem setul de date mtcars
> str(mtcars) # structura setului de date
'data.frame': 32 obs. of 11 variables:
 $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num 160 160 108 258 360 ...
 $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num 16.5 17 18.6 19.4 17 ...
 $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
 $ am : num 1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

> 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

> 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

> rownames(mtcars)
 [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
 [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
 [7] "Duster 360"         "Merc 240D"           "Merc 230"
[10] "Merc 280"           "Merc 280C"           "Merc 450SE"
[13] "Merc 450SL"         "Merc 450SLC"         "Cadillac Fleetwood"
[16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
[19] "Honda Civic"        "Toyota Corolla"      "Toyota Corona"
[22] "Dodge Challenger"   "AMC Javelin"         "Camaro Z28"
[25] "Pontiac Firebird"   "Fiat X1-9"           "Porsche 914-2"
[28] "Lotus Europa"       "Ford Pantera L"      "Ferrari Dino"
[31] "Maserati Bora"      "Volvo 142E"

> names(mtcars)
 [1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"
[11] "carb"

> view(mtcars) # Afiseaza o fereastră cu un tabel ce contine toate informatiile
```

Metode de indexare

- Indexarea structurilor de tip dataframe se face la fel ca și indexarea listelor.

```
> mtcars[1,1:4]
      mpg cyl disp  hp
Mazda RX4   21   6  160 110

> mtcars[c(1,2),2]
[1] 6 6

> mtcars$mpg
[1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4

# DataFrame-urile indexare logica
> mtcars[mtcars$mpg > 25, ]
      mpg cyl  disp  hp drat   wt  qsec vs am gear carb
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
# .....

> mtcars[(mtcars$mpg > 25) & (mtcars$wt < 1.8), ]
      mpg cyl  disp  hp drat   wt  qsec vs am gear carb
Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52 1  1    4    2
Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90 1  1    5    2

# O alta metoda de indexare este prin folosirea functiei subset().
> subset(x = mtcars, subset = mpg < 12 & cyl > 6, select = c(displ, wt))
      displ   wt
Cadillac Fleetwood  472 5.250
Lincoln Continental  460 5.424
```

Metode de manipulare

Funcția order

```
# Afisarea celor mai usoare 10 masini
> cars_increasing = rownames(mtcars[order(mtcars$wt),])
> cars_increasing[1:10]
[1] "Lotus Europa"    "Honda Civic"    "Toyota Corolla" "Fiat x1-9"
[5] "Porsche 914-2"  "Fiat 128"       "Datsun 710"     "Toyota Corona"
[9] "Mazda RX4"      "Ferrari Dino"

# Afisarea celor mai grele 10 masini
> cars_decreasing = rownames(mtcars[order(mtcars$wt, decreasing = TRUE), ])
> cars_decreasing[1:10]
[1] "Lincoln Continental" "Chrysler Imperial" "Cadillac Fleetwood"
[4] "Merc 450SE"         "Pontiac Firebird"  "Camaro Z28"
[7] "Merc 450SLC"        "Merc 450SL"        "Duster 360"
[10] "Maserati Bora"

# Ordonarea a mai mult de o coloană
> mtcars[order(mtcars$cyl, mtcars$wt), 1:6]
```

```

      mpg cyl  disp  hp drat   wt
Lotus Europa    30.4   4  95.1 113 3.77 1.513
Honda Civic     30.4   4  75.7  52 4.93 1.615
Toyota Corolla  33.9   4  71.1  65 4.22 1.835
Fiat X1-9       27.3   4  79.0  66 4.08 1.935
Porsche 914-2   26.0   4 120.3  91 4.43 2.140
# .....

```

Funcția merge

```

> stat_course = data.frame(note_stat = c(9, 8, 5, 7, 9),
  student = c("Ionel", "Maria", "Gigel", "Vasile", "Ana"))
> alg_course = data.frame(note_alg = c(10, 8, 9, 7, 9),
  student = c("Maria", "Ana", "Gigel", "Ionel", "Vasile"))
> combined_courses = merge(x = stat_course, y = alg_course, by = "student")
> combined_courses
  student note_stat note_alg
1     Ana         9         8
2    Gigel         5         9
3    Ionel         9         7
4    Maria         8        10
5   Vasile         7         9

```

Funcția aggregate

```

# Structura generala
aggregate(formula = dv ~ iv, # dv este data, iv este grupul
  FUN = fun, # functia pe care vrem sa o aplicam
  data = df # setul de date ce contine coloanele dv si iv
)

```

```

# Fara functia aggregate
> mean(Chickweight$weight[Chickweight$Diet == 2])
[1] 122.6167
> mean(Chickweight$weight[Chickweight$Diet == 3])
[1] 142.95
> mean(Chickweight$weight[Chickweight$Diet == 4])
[1] 135.2627

# Cu ajutorul functiei aggregate
> aggregate(formula = weight ~ Diet, FUN = mean, data = Chickweight)
  Diet  weight
1    1 102.6455
2    2 122.6167
3    3 142.9500
4    4 135.2627

# Pentru restrictii, apelam la functia subset
> aggregate(formula = weight ~ Diet, FUN = mean, subset = Time < 10, data =
Chickweight)
  Diet  weight
1    1 58.03093
2    2 63.40000
3    3 65.94000
4    4 69.36000

```

```
# Folosind mai multe variabile independente
> aggregate(formula = weight ~ Diet + Time, FUN = mean, data = Chickweight)
  Diet Time    weight
1     1   0 41.40000
2     2   0 40.70000
3     3   0 40.80000
4     4   0 41.00000
5     1   2 47.25000
# .....
```

Aplicație

Considerați setul de date `mtcars`. Calculați:

- Greutatea medie în funcție de tipul de transmisie.

```
# wt - weight (greutate)
# am - transmission (transmisie)
> aggregate(formula = wt ~ am, FUN = mean, data = mtcars)
  am    wt
1  0 3.768895
2  1 2.411000
```

- Greutatea medie în funcție de numărul de cilindrii.

```
# cyl - Number of cylinders (numar de cilindri)
> aggregate(formula = wt ~ cyl, FUN = mean, data = mtcars)
  cyl    wt
1   4 2.285727
2   6 3.117143
3   8 3.999214
```

- Consumul mediu în funcție de numărul de cilindrii și tipul de transmisie.

```
> aggregate(formula = wt ~ cyl + am, FUN = mean, data = mtcars)
  cyl am    wt
1   4  0 2.935000
2   6  0 3.388750
3   8  0 4.104083
4   4  1 2.042250
5   6  1 2.755000
6   8  1 3.370000
```