# Data types and structures, matrices, data frames and lists

—

B83128 – Introduction to R scripting language (shortened version)

Lubomír Štěpánek[1, 2]

[1]Department of Biomedical Statistics
Institute of Biophysics and Informatics
First Faculty of Medicine
Charles University, Prague


[2]Department of Biomedical Informatics
Faculty of Biomedical Engineering
Czech Technical University in Prague

# Content

# Data structures

- a vector (`vector`)
- a factor (`factor`)
- a matrix (`matrix`)
- a data frame (`data.frame`)
- a list (`list`)

# Matrices' initialization and basic commands

- a matrix (`matrix`) is a two-dimensional array containing values of (only) one data type
- all columns of a matrix are of one length, and all rows of a matrix are of one length
- let

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

- in R, we get the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ by

```
1  A <- matrix(c(1, 2, 3, 4), nrow = 2,
2              ncol = 2)
3  B <- matrix(c(1, 3, 2, 4), nrow = 2,
4              ncol = 2)
5  B <- matrix(c(1, 2, 3, 4), nrow = 2,
6              ncol = 2, byrow = TRUE)
7  # only one of the arguments "nrow" and "ncol" is necessary
```

# Manipulation with matrices

- let

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- in R using

```
C <- matrix(letters[1:12], nrow = 3,
            byrow = T)
```

- some useful commands are

```
is.matrix(C)    # TRUE
class(C)        # "matrix"
mode(C)         # "character"; data type of matrix
                # values
str(C)          # chr [1:3, 1:4] "a" "e" "i" ...
dim(C)          # c(3, 4); dimensions of matrix C
```

# Manipulation with matrices

- let

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- other bunch of useful commands

```
colnames(C) <- c("c1", "c2", "c3", "c4")
rownames(C) <- c("r1", "r2", "r3")
                # adds labels to columns and rows
C <- unname(C)
                # deletes the labels of columns
                # and rows
dimnames(C) <- list(
                    c("r1", "r2", "r3"),
                    c("c1", "c2", "c3", "c4")
                )
                # also adds labels to columns and rows
```

# Manipulation with matrices

- let's have

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- another bunch of useful commands

```
1  rbind(C, c("x", "x", "x", "x"))
2                  # adds a row of c("x", "x", "x", "x")
3                  # to the matrix C
4  cbind(C, c("x", "x", "x"))
5                  # adds a column of c("x", "x", "x")
6                  # to the matrix C
7  C[-1, ]         # deletes the 1-st row of the matrix C
8  C[, -2]         # deletes the 2-nd column
9                  # of the matrix C
```

# Submatrices, indexing, addressing

- let

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- in R by

```
1  C <- matrix(letters[1:12], nrow = 3,
2              byrow = T, dimnames = list(
3              c("r1", "r2", "r3"),
4              c("c1", "c2", "c3", "c4")))
5  C[2, 3]          # "g"; a value of the 2-nd row
6                   # and the 3-st column
7  C["r2", "c3"]    # "g"; a value of the 2-nd row
8                   # and the 3-st column
9  C[1, ]           # c("a", "b", "c", "d");
10                  # a vector of the 1-st row of the matrix C
11                  # with labels
```

# Submatrices, indexing, addressing

- still let's have

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- addressing

```
1  C[, 3]              # c("c", "g", "k");
2                      # a vector of the 3-rd column
3                      # of the matrix C with labels
4  C[c(1, 3), c(2, 4)]
5                      # matrix(c("b", "j", "d", "l"), 2)
6                      # a submatrix of the 1-st and 3-rd rows,
7                      # 2-nd and 4-th column of the matrix C
8                      # with labels
9  C["r2", ]           # c("e", "f", "g", "h");
10                     # a vector of the 2-nd row
11                     # of the matrix C with labels
```

# Submatrices, indexing, addressing

- let's have

$$\boldsymbol{C} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

- addressing

```
C[dim(C)[1], dim(C)[2]]
                # "l"; a general addressing
                # of the right bottom page
C[5]            # "f"; major-column ordering
C[c(8, 9)]      # c("g", "k")
C[13]           # NA
diag(C)         # c("a", "f", "k"); main diagonal
diag(C[, dim(C)[2]:1])
                # c("d", "g", "j"); opposite diagonal
```

# Matrix algebra

- let

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

- in R using

```
A <- matrix(c(1, 2, 3, 4), nrow = 2)
B <- matrix(c(5, 6, 7, 8), nrow = 2)
```

- Hadamard's product (*element-wise*, *pairwise*) $\boldsymbol{A} \circ \boldsymbol{B} = \begin{pmatrix} 5 & 21 \\ 12 & 32 \end{pmatrix}$

```
A * B       # matrix(c(5, 12, 21, 32), 2)
```

- matrix product $\boldsymbol{A} \cdot \boldsymbol{B} = \begin{pmatrix} 23 & 31 \\ 34 & 46 \end{pmatrix}$

```
A %*% B     # matrix(c(23, 34, 31, 46), 2)
```

# Matrix algebra

- let

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

- in R using

```r
A <- matrix(c(1, 2, 3, 4), nrow = 2)
B <- matrix(c(5, 6, 7, 8), nrow = 2)
```

- transposition $\boldsymbol{A}^T = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

```r
t(A)          # matrix(c(1, 3, 2, 4), 2)
```

# Data frames initialization and basic commands

- a `data.frame` is two-dimensional array that consists of columns of values having a same data type
- data types of different columns could be different
- all columns of a matrix are of one length, and all rows of a matrix are of one length
- in R, there are many in-built data.frames

```
mtcars
str(mtcars)
class(mtcars)              # "data.frame"
mode(mtcars)               # "list"
is.data.frame(mtcars)      # TRUE
str(iris)
     # 'data.frame': 150 obs. of 5 variables
     # ...
dim(iris)                  # c(150, 5)
```

# Manipulation with data frames

- a bunch of useful commands

```
1  data <- mtcars
2  colnames(data)
3  colnames(data) <- paste("c",
4                          1:dim(data)[2],
5                          sep = "_")
6  rownames(data) <- paste("r",
7                          1:dim(data)[1],
8                          sep = "_")
9                  # changes labels of rows and commands
10 head(data)      # prints out the first 6 rows
11 head(data, 10)
12                 # prints out the first 10 rows
13 tail(data)      # prints out the last 6 rows
14 tail(data, 10)
15                 # prints out the last 10 rows
```

# Manipulation with data frames

- other useful commands

```
rbind(data, rep(0, dim(data)[2]))
                  # adds a row of c(0, 0, ..., 0)
                  # to data.frame "data"
cbind(data, rep(0, dim(data)[1]))
                  # adds a column of c(0, 0, ..., 0)
                  # to data.frame "data"
data.frame(data,
           "hello" = rep(0, dim(data)[1]))
                  # adds a row of c(0, 0, ..., 0)
                  # with a label "hello" to data.frame
                  # "data"
data[-1, ]        # removes the 1-st row of data.frame
                  # "data"
data[, -1]        # removes the 1-st column of data.frame
                  # "data"
```

# Indexing, addressing

```
data[2, 3]      # 160; a value of the 2-nd row,
                # 3-rd column
data["r_2", "c_3"]
                # 160; a value given by the labels
data[1, ]       # c(21, 6, 160, 110, ...);
                # a vector of the 1-st row
                # of data frame "data" with labels
data[, 2]       # c(6, 6, 4, 6, ...);
                # a vector of the 2-nd column
                # of data frame "data" with labels
data$c_5        # c(3.90, 3.90, 3.85, 3.08, ...);
                # a vector of the 5-th column
                # of data frame "data" with labels
data$c_5[1]     # 3.9;
                # the first value of vector
                # of the 5-th column
                # of data frame "data" with labels
```

# Indexing, addressing

```
data[dim(data)[1], dim(data)[2]]
                # 2; a general way how to address
                # the right bottom values
                # of the data.frame
data[5]         # the 5-th column, not the major-column
                # ordering!
```

# Columns' summaries

- sometimes it is handy to get quickly a summary for all columns of a data.frame

```
1 colSums(data)
2             # a vector of sums for each column
3 apply(data, 2, sum)
4             # the same as above
5 colMeans(data)
6             # a vector of means for each column
7 apply(data, 2, mean)
8             # the same as above
```

# Columns' summaries

```
data <- rbind(data, rep(NA, dim(data)[2]))
          # adds a row of c(NA, NA, ...)
colMeans(data)
          # c(NA, NA, ...); to avoid non-informative
          # output, we have to add an argument
          # na.rm = TRUE
colMeans(data, na.rm = TRUE)
apply(data, 2, mean, na.rm = TRUE)
          # both commands work now
```

# Lists initialization and basic commands

- a `list` is a tuple of items such that each item could have its own data structure or data type
- a list could include one or even more other lists
- lengths of the items in a list could differ

```
my_list <- list("a" = c(1:10),
                "b" = mtcars,
                "c" = matrix(1:8, 2),
                "z" = "hello")
str(my_list)
class(my_list)          # "list"
mode(my_list)           # "list"
is.list(my_list)        # TRUE
```

# Manipulation with lists

- a bunch of useful commands

```
1  names(my_list)
2  names(my_list) <- LETTERS[
3       1:length(my_list)
4  ]              # changing the list's item names
5
6  my_list[[length(my_list) + 1]] <- c(T, F)
7                 # adding a vector of c(T, F)
8                 # to the list "my_list"
9  names(my_list)[length(my_list)] <- "XY"
10                # and adding a name "XY"
11                # to the vector as an item
```

# Indexing, addressing

```
my_list[[2]]   # the 2-nd item of the list
my_list[["B"]]
               # an item of the list labeled by "B"
               # of the list
               # it's the data.frame "mtcars"
               # because of the double brackets
my_list["B"]   # an item of the list labeled by "B"
               # it's still a list
               # because of the single brackets
my_list[c(2, 4)]
               # the 2-nd and 4-th item of the list
my_list$C      # an item of the list labeled by "C"
```

# Indexing, addressing

```
1 my_list[[1]][2]
2                    # 2; the 2-nd value of the 1-st item
3                    # of the list
4 my_list[[2]][3, 5]
5                    # 3.85; the 3-nd and 5-th value
6                    # of the 2-nd item of the list
```

# Indexing, addressing using `lapply()` function

```
1  set.seed(1)
2  my_long_list <- lapply(
3      sample(c(80:120), 100, TRUE),
4      function(x) sample(
5          c(50:150), x, replace = TRUE
6      )
7  )    # a list of vectors of different lengths
8       # populated by random numbers
9       # from range (80, ..., 100)
10
11 lapply(my_long_list, "[[", 14)
12      # prints out the 14-th value of each item
13      # of the list
14      # - this is handy since otherwise
15      # it would require to use a for loop
```

# Items' summaries

- sometimes it is handy to get quickly a summary for each item of a list

```
lapply(my_long_list, mean)
    # printing out a mean of all
    # values for each item of the list

lapply(my_long_list, length)
    # printing out a length
    # for each item of the list
```

# References


Alain F. Zuur, Elena N. Ieno und Erik Meesters. *A Beginner's Guide to R*. Springer New York, 2009. DOI: 10.1007/978-0-387-93837-0. URL: https://doi.org/10.1007/978-0-387-93837-0.

Hadley Wickham. *Advanced R*. Boca Raton, FL: CRC Press, 2015. ISBN: 978-1466586963.

## Thank you for your attention!

lubomir.stepanek@lf1.cuni.cz
lubomir.stepanek@fbmi.cvut.cz

▸ GitHub

https://github.com/LStepanek/B83128_Introduction_to_R_scripting_language