

# Biostatistics I: Introduction to R

## Common objects

Eleni-Rosalina Andrinopoulou

Department of Biostatistics, Erasmus Medical Center

✉ [e.andrinopoulou@erasmusmc.nl](mailto:e.andrinopoulou@erasmusmc.nl)

🐦 [@erandrinopoulou](https://twitter.com/erandrinopoulou)

## In this Section

---

- ▶ Objects in R
- ▶ Data types
- ▶ Data structures
- ▶ A lot of practice

# Objects in R

---

- ▶ In R Everything (data, results, ...) is an object
- ▶ In order to list the created objects use the following functions

```
objects()  
ls()
```

# Objects in R

---

- ▶ In R Everything (data, results, ...) is an object
- ▶ In order to list the created objects use the following functions

```
objects()  
ls()
```

- ▶ In order to remove objects

```
rm()  
rm(list=ls(all=TRUE))
```

# Objects in R

---

- ▶ To investigate a specific object (e.g. pbc)

```
str(pbc[,c("id", "time", "status", "trt", "age", "sex", "bili", "chol")])
```

```
'data.frame':  418 obs. of  8 variables:  
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...  
 $ time    : int  400 4500 1012 1925 1504 2503 1832 2466 2400 51 ...  
 $ status  : int  2 0 2 2 1 2 0 2 2 2 ...  
 $ trt     : int  1 1 1 1 2 2 2 2 1 2 ...  
 $ age     : num  58.8 56.4 70.1 54.7 38.1 ...  
 $ sex     : Factor w/ 2 levels "m","f": 2 2 1 2 2 2 2 2 2 2 ...  
 $ bili    : num  14.5 1.1 1.4 1.8 3.4 0.8 1 0.3 3.2 12.6 ...  
 $ chol    : int  261 302 176 244 279 248 322 280 562 200 ...
```

# Data Types

---

The simplest data types are:

- ▶ **numeric** : quantitative data
- ▶ **character** : qualitative data
- ▶ **integer** : whole numbers
- ▶ **logical** : TRUE or FALSE
- ▶ **factors** : qualitative data (levels)

## Data Types in R

---

To find out what type of object you have, you can use the following function

```
mode(pbc$age)
```

```
[1] "numeric"
```

```
str(pbc$age)
```

```
num [1:418] 58.8 56.4 70.1 54.7 38.1 ...
```

# Data Structures

---

The most important data structures are:

- ▶ **Scalar** a single element
- ▶ **Vectors** have the same type of elements
- ▶ **Matrices** have the same type of elements with the same length
- ▶ **Arrays** have the same type of elements with the same length but can store the data in more than two dimensions
- ▶ **Data frames** have elements of different type with the same length
- ▶ **Lists** have elements of different type and length



# Data Structures

---

**How do these data structures look like?**

# Data Structures

---

- ▶ Differences between **vector**, matrix, array, data.frame and list

```
pbcc[1:6, c("age")]
```

```
[1] 58.76523 56.44627 70.07255 54.74059 38.10541 66.25873
```

# Data Structures

---

- ▶ Differences between vector, **matrix**, array, data.frame and list

```
pbcc[1:6, c("age", "bili", "chol")]
```

	age	bili	chol
1	58.76523	14.5	261
2	56.44627	1.1	302
3	70.07255	1.4	176
4	54.74059	1.8	244
5	38.10541	3.4	279
6	66.25873	0.8	248

# Data Structures

---

- ▶ Differences between vector, matrix, **array**, data.frame and list

```
pbcc[1:3, c("age", "bili", "chol")]
```

```
      age bili chol
1 58.76523 14.5  261
2 56.44627  1.1  302
3 70.07255  1.4  176
```

```
pbcc[4:6, c("age", "bili", "chol")]
```

```
      age bili chol
4 54.74059  1.8  244
5 38.10541  3.4  279
6 66.25873  0.8  248
```

# Data Structures

---

- ▶ Differences between vector, matrix, **array**, data.frame and list

```
pbcc[1:2, c("prottime", "time")]
```

```
      prottime time
1      12.2  400
2      10.6 4500
```

```
pbcc[3:4, c("prottime", "time")]
```

```
      prottime time
3      12.0 1012
4      10.3 1925
```

# Data Structures

---

- ▶ Differences between vector, matrix, array, **data.frame** and list

```
pbcc[1:6, c("id", "sex", "bili", "chol")]
```

	id	sex	bili	chol
1	1	f	14.5	261
2	2	f	1.1	302
3	3	m	1.4	176
4	4	f	1.8	244
5	5	f	3.4	279
6	6	f	0.8	248

# Data Structures

---

- ▶ Differences between vector, matrix, array, data.frame and **list**

```
pbcc[1:6, c("sex")]
```

```
[1] f f m f f f  
Levels: m f
```

```
pbcc[1:2, c("sex", "bili")]
```

```
  sex bili  
1  f 14.5  
2  f  1.1
```

```
pbcc[1:4, c("age")]
```

```
[1] 58.76523 56.44627 70.07255 54.74059
```

# Data Structures in R

---

**Let's now create different data structure in R!**



# Data Structures in R

---

## Create a vector

```
vec <- c(1, 2, 3, 4, 5)
```

```
vec
```

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

```
vec <- c(1:5)
```

```
vec
```

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

# Data Structures in R

---

## Create a matrix

```
vec <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
mat <- matrix(data = vec,
              nrow = 3, ncol = 3)
mat
```

	[,1]	[,2]	[,3]
[1,]	1	4	7
[2,]	2	5	8
[3,]	3	6	9

# Data Structures in R

## Create a matrix

```
vec <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
mat <- matrix(data = vec,
              nrow = 3, ncol = 3)
mat
```

	[,1]	[,2]	[,3]
[1,]	1	4	7
[2,]	2	5	8
[3,]	3	6	9

```
vec <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
mat <- matrix(data = vec,
              nrow = 3, ncol = 3,
              byrow = TRUE)
mat
```

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	4	5	6
[3,]	7	8	9

# Data Structures in R

---

## Create an array

```
ar <- array(data = c(1, 2, 3, 4, 5, 6, 7, 8), dim = c(2, 2, 2))  
ar
```

```
, , 1
```

```
      [,1] [,2]  
[1,]    1    3  
[2,]    2    4
```

```
, , 2
```

```
      [,1] [,2]  
[1,]    5    7  
[2,]    6    8
```

# Data Structures in R

---

## Create an array

```
ar <- array(data = c(1, 2, 3, 4), dim = c(2, 2, 1))  
ar
```

```
, , 1
```

```
      [,1] [,2]  
[1,]    1    3  
[2,]    2    4
```

# Data Structures in R

---

## Create a data frame

```
dtf <- data.frame(pbc[, "sex"],  
                  pbc[, "age"])  
dtf[1:3,]
```

	pbccolsex	pbccolage
1	f	58.76523
2	f	56.44627
3	m	70.07255

# Data Structures in R

---

## Create a data frame

```
dtf <- data.frame(pbc[, "sex"],  
                  pbc[, "age"])  
dtf[1:3,]
```

	pbccccsex..	pbccccage..
1	f	58.76523
2	f	56.44627
3	m	70.07255

```
dtf <- data.frame(Gender = pbc[, "sex"],  
                  Age = pbc[, "age"])  
dtf[1:3,]
```

	Gender	Age
1	f	58.76523
2	f	56.44627
3	m	70.07255

# Data Structures in R

---

## Create a list

```
list1 <- list(vec = c(1:5), mat = pbc[1:2, c("age", "sex")])  
list1
```

```
$vec
```

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

```
$mat
```

```
      age sex  
1 58.76523  f  
2 56.44627  f
```



# Summary

---

## Data types

- ▶ `is.numeric() / as.numeric()`
- ▶ `is.character() / as.character()`
- ▶ `is.integer() / as.integer()`
- ▶ `is.logical / as.logical()`
- ▶ `is.factor() / as.factor()`
- ▶ `str(), mode()`

## Data structures

- ▶ `c()`
- ▶ `matrix()`
- ▶ `array()`
- ▶ `data.frame()`
- ▶ `list()`

## Other

- ▶ `ls(), objects()`