Introduction to R programming for data science – day 3

Dr. Francesca Finotello
Medical University of Innsbruck, Austria

Data structures (part 2)

Data structures

In R, there are different **data structures**, including:

Data structure	Can contain different data types?
Vector	No
Matrix	No
Factor	No
List	Yes
Data.frame	Yes

Data structure: matrix

A **matrix** is a 2-dimensional object that can be build with the function *matrix*

```
( M <- matrix(1:6, ncol=3, byrow=FALSE) ) # Fill by columns (def.)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
( N <- matrix(1:6, ncol=3, byrow=TRUE) ) # Fill by rows
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

Matrix rows and columns

```
M <- matrix(1:6, nrow=2, byrow=FALSE)
dim(M)
```

```
## [1] 2 3
```

```
ncol(M)
```

```
## [1] 3
```

```
nrow(M)
```

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


Matrix row and column names

```
colnames(M) <- c("Sample_A", "Sample_B", "Sample_C") # Assign
rownames(M) <- c("Gene1", "Gene2") # Assign
M
```

```
##       Sample_A Sample_B Sample_C
## Gene1        1        3        5
## Gene2        2        4        6
```

```
colnames(M) # Query
```

```
## [1] "Sample_A" "Sample_B" "Sample_C"
```

```
rownames(M) # Query
```

```
## [1] "Gene1" "Gene2"
```

Data structure: data.frame

A **data.frame** is a 2-dimensional object ("matrix-like") that can contain different types of data on its colums.

It can be created with the function *data.frame*

```
( DF <- data.frame(name=c("Mary", "John", "Lisa"),
  age=c(19, 30, 20),
  city=c("New York", "Seattle", "New York") ) )
```

```
##   name age     city
## 1 Mary  19 New York
## 2 John  30  Seattle
## 3 Lisa  20 New York
```


Data.frame rows and columns

```
dim(DF)
```

```
## [1] 3 3
```

```
nrow(DF)
```

```
## [1] 3
```

```
ncol(DF)
```

```
## [1] 3
```

```
colnames(DF)
```

```
## [1] "name" "age"  "city"
```

Accessing matrices

```
M <- matrix(1:6, nrow=2, byrow=FALSE)
colnames(M) <- c("Sample_A", "Sample_B", "Sample_C")
rownames(M) <- c("Gene1", "Gene2")
M
```

```
##       Sample_A Sample_B Sample_C
## Gene1        1        3        5
## Gene2        2        4        6
```

```
M[1,2] # Two indices: [row,column]
```

```
## [1] 3
```


Subsetting matrices with numeric indexes

Accessing the first row:

```
M[1,]
```

```
## Sample_A Sample_B Sample_C
##        1        3        5
```

Removing the second column:

```
M[,-2]
```

```
##       Sample_A Sample_C
## Gene1        1        5
## Gene2        2        6
```

Subsetting matrices with row/column names

Elements of matrices and data.frames can be accessed also considering row and column names

```
M["Gene1",]
```

```
## Sample_A Sample_B Sample_C
##        1        3        5
```

```
M[,"Sample_C"]
```

```
## Gene1 Gene2
##     5     6
```


Subsetting matrices: the “drop” option

By default, R transforms one-dimensional objects into vectors, unless we specify drop = FALSE

```
v <- M[1,]

dim(v)
```

```
## NULL
```

```
m <- M[1,, drop = FALSE]

dim(m)
```

```
## [1] 1 3
```

Subsetting data.frames

Data.frames can be handled similarly to matrices

```
( DF <- data.frame(name=c("Mary", "John"), age=c(19, 30) ) )
```

```
##   name age
## 1 Mary  19
## 2 John  30
```

```
DF[,"age"]
```

```
## [1] 19 30
```

```
DF[1,2]
```

```
## [1] 19
```


Data structure: list

A **list** is an object that can contain different objects (including lists!)

It can be created with the function *list*

```
v <- 1:3
M <- matrix(1:4, ncol=2)
( myList <- list(myVec=v, myMat=M) )
```

```
## $myVec
## [1] 1 2 3
##
## $myMat
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

List names and length

```
length(myList)
```

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

```
names(myList)
```

```
## [1] "myVec" "myMat"
```

```
myList <- list(v, M)
names(myList)
```

```
## NULL
```


Accessing lists

Elements in lists can be accessed using double square brackets

```
favorites <- list(colors=c("blue", "purple"),
  cities=c("Venice", "Innsbruck", "New York City"))

favorites[[1]]
```

```
## [1] "blue"   "purple"
```

```
favorites[["cities"]]
```

```
## [1] "Venice"        "Innsbruck"     "New York City"
```

Functions (part 2)

Useful functions: *rbind* and *cbind*

```
(M1 <- matrix(1, ncol=2, nrow=2))
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    1
```

```
(M2 <- matrix(2, ncol=2, nrow=2))
```

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    2    2
```

Useful functions: *rbind* and *cbind*

```
cbind(M1,M2)
```

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

```
rbind(M1,M2)
```

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


Useful functions: *apply*

```
( M <- rbind(c(1,1,1), c(2,2,2)) )
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    2    2    2
```

```
apply(M,1,sum)
```

```
## [1] 3 6
```

```
M <- as.data.frame(M)
apply(M,2,sum)
```

```
## V1 V2 V3
##  3  3  3
```

Useful functions: *lapply*

```
L <- list(group1 = c(1, 1, 2),
          group2 = c(100, 50, 60),
          group3 = c(100, 100, 100, 1000, 2000))

lapply(L, mean)
```

```
## $group1
## [1] 1.333333
##
## $group2
## [1] 70
##
## $group3
## [1] 660
```


Useful functions: *head* and *tail*

```
M3 <- rbind(M1, M2)
tail(M3, 2)
```

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

```
head(L, 2)
```

```
## $group1
## [1] 1 1 2
##
## $group2
## [1] 100  50  60
```