Data Object

Quantide Srl

2016-05-02

Vectors

Exercise 1

a. Create a vector, named `vec1`, containing the following values:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90

```
vec1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90)
```

b. Select the 5-th element of `vec1`.

```
vec1[5]
```

```
## [1] 5
```

c. Select the first 10 elements of `vec1.`

```
vec1[1:10]
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

d. Select all the elements of `vec1` apart from the 2nd and the 6th element.

```
vec1[-c(2,6)]
```

```
##  [1]  1  3  4  5  7  8  9 10 15 20 25 30 35 40 45 50 60 70 80 90
```

Exercise 2

a. Generate a vector, named `vec2`, containing numbers from 1 to 10 and of length 8, using the operator `seq`.

```
vec2 <- seq(from=1, to=10, length.out = 8)
```

b. Select the values of `vec2` which are greater than 4.

```
vec2[vec2>4] # or y > 4; b[y]
```

```
## [1]  4.857143  6.142857  7.428571  8.714286 10.000000
```

c. Select the values of `vec2` which are equal or less than 2 or which are equal or greater than 6.

```
vec2[vec2<=2 | vec2>=6]
```

```
## [1]  1.000000  6.142857  7.428571  8.714286 10.000000
```

Exercise 3

a. Generate the following vector using the operator `rep`:
   `vec3 <- c("one", "two", "one", "two", "one", "two")`

```
vec3 <- rep(c("one", "two"), times=3)
```

b. Generate a new vector, named `vec5`, combining the previous vector, `vec3`, with the following one:

```
vec4 <- c("three", "four")
```

```
vec5 <- c(vec3, vec4)
vec5
```

```
## [1] "one"   "two"   "one"   "two"   "one"   "two"   "three" "four"
```

Matrices

Exercise 1

Generate a matrix, named `mat1`, with 5 rows and 3 columns, using `matrix` function:

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
## [4,]   10   11   12
## [5,]   13   14   15
```

```
mat1 <- matrix(1:15, nrow = 5, ncol = 3, byrow = TRUE)
```

Exercise 2

Starting from the following vector:

```
mat2 <- 1:8
```

Generate a matrix with 2 rows and 4 columns using the function `dim`.

```
dim(mat2) <- c(2,4)
mat2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
```

Exercise 3

a. Generate a matrix, named `mat3`, combining the following columns:

```
a <- 1:3
b <- 7:9
c <- 8:6
```

```
mat3 <- cbind(a,b,c)
mat3
```

```
##      a b c
## [1,] 1 7 8
## [2,] 2 8 7
## [3,] 3 9 6
```

b. Add the following row to `mat3`:

```
d <- 4:6
```

```
mat3 <-rbind(mat3, d)
mat3
```

```
##   a b c
##   1 7 8
##   2 8 7
##   3 9 6
## d 4 5 6
```

Exercise 4

Considering the following matrix, named `mat4`:

```
mat4 <- matrix(1:24, nrow = 6, ncol = 4, byrow = TRUE)
mat4
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
## [4,]   13   14   15   16
## [5,]   17   18   19   20
## [6,]   21   22   23   24
```

a. Select the third and the fifth row of `mat4`.

```
mat4[c(3,5),]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    9   10   11   12
## [2,]   17   18   19   20
```

b. Select all columns of `mat4` apart from the first.

```
mat4[, -1]
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    6    7    8
## [3,]   10   11   12
## [4,]   14   15   16
## [5,]   18   19   20
## [6,]   22   23   24
```

c. Select second and third rows and second and third columns of `mat4`.

```
mat4[2:3, 2:3] # or mat4[c(2,3) , c(2,3)]
```

```
##      [,1] [,2]
## [1,]    6    7
## [2,]   10   11
```

Lists

Exercise 1

a. Generate a list, named `list1` that contains the following R elements:

```
vec <- 1:10
mat <- matrix(1:9, ncol = 3)
name <- "Oscar"
```

```
list1 <- list(vec = 1:10, mat = matrix(1:9, ncol = 3), name = "Oscar")
list1
```

```
## $vec
##  [1]  1  2  3  4  5  6  7  8  9 10
##
## $mat
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
##
## $name
## [1] "Oscar"
```

b. Add to `list1` the following element:

```
letters <- c("a", "b", "c", "d")
```

```
list1$letters <- letters
list1
```

```
## $vec
##  [1]  1  2  3  4  5  6  7  8  9 10
##
## $mat
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
##
## $name
## [1] "Oscar"
##
## $letters
## [1] "a" "b" "c" "d"
```

Exercise 2

Given the following list, named `list2`:

```
list2 <- list(vec = c(1,3,5,7,8), mat = matrix(1:12, ncol = 4),
              sub_list = list(names = c("Veronica", "Enrico", "Andrea", "Anna"),
                              numbers = 1:4))
list2
```

```
## $vec
## [1] 1 3 5 7 8
##
## $mat
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
##
## $sub_list
## $sub_list$names
## [1] "Veronica" "Enrico"   "Andrea"   "Anna"
##
## $sub_list$numbers
## [1] 1 2 3 4
```

a. Entract the first element of `list2`.

```
list2[1]
```

```
## $vec
## [1] 1 3 5 7 8
```

b. Extract the objects contained in the first element of `list2`.

```
list2[[1]]
```

```
## [1] 1 3 5 7 8
```

c. Extract the object named `sub_list` of `list2`.

```
list2$sub_list
```

```
## $names
## [1] "Veronica" "Enrico"   "Andrea"   "Anna"
##
## $numbers
## [1] 1 2 3 4
```

d. Extract the second rows of the matrix included in the second element of `list2`.

```
list2[[2]][2,] # or list2$mat[2,]
```

```
## [1]  2  5  8 11
```

Factors

Exercise 1

Starting from the vector:

```
fac1 <- c("F", "F", "M", "M" , "F")
```

Generate the corresponding factor with two levels: "F" and "M"

```
fac1 <- factor(fac1, levels = c("F", "M"))
fac1
```

```
## [1] F F M M F
## Levels: F M
```

Exercise 2

Starting from the vector:

```
fac2 <- c(1, 1, 1, 2, 2, 2)
```

a. Generate the corresponding factor considering that 1 = "Female", 2 = "Male" e 3 = "Trans".

```
fac2 <- factor(fac2, levels = c(1,2,3), labels = c("Female", "Male", "Trans"))
fac2
```

```
## [1] Female Female Female Male   Male   Male
## Levels: Female Male Trans
```

b. Select the all elements of `fac2` apart from "Male".

```
fac2[fac2!= "Male"]
```

```
## [1] Female Female Female
## Levels: Female Male Trans
```

Data Frames

Exercise 1

a. Generate a data frame, named `df1`, corresponding to:

```
##    id     name class mean
## 1   1     Luca    5A  6.0
## 2   2   Chiara    5A  7.0
## 3   3     Lisa    5A  5.0
## 4   4   Matteo    5A  6.5
## 5   5    Alice    5A  7.5
## 6   6    Marco    5B  4.5
## 7   7 Veronica    5B  9.0
## 8   8   Nicola    5B  8.0
## 9   9    Elena    5B  8.5
## 10 10  Daniele    5B  7.0
```

Remember to maintain character vectors as they are, specifiyng `stringsAsFactors = FALSE`.

```
df1 <- data.frame(id=1:10,
                  name=c("Luca", "Chiara", "Lisa", "Matteo", "Alice", "Marco",
                         "Veronica", "Nicola", "Elena", "Daniele"),
                  class=c(rep("5A", times=5), rep("5B", times=5)),
                  mean= c(6,7,5,6.5,7.5,4.5, 9, 8, 8.5, 7), stringsAsFactors = FALSE)
df1
```

```
##    id     name class mean
## 1   1     Luca    5A  6.0
## 2   2   Chiara    5A  7.0
## 3   3     Lisa    5A  5.0
## 4   4   Matteo    5A  6.5
## 5   5    Alice    5A  7.5
## 6   6    Marco    5B  4.5
## 7   7 Veronica    5B  9.0
## 8   8   Nicola    5B  8.0
## 9   9    Elena    5B  8.5
## 10 10  Daniele    5B  7.0
```

```
# Other solution
id <- 1:10
name <- c("Luca", "Chiara", "Lisa", "Matteo", "Alice", "Marco",
          "Veronica", "Nicola", "Elena", "Daniele")
class <- c(rep("5A", times=5), rep("5B", times=5))
mean <- c(6,7,5,6.5,7.5,4.5, 9, 8, 8.5, 7)
df1 <- data.frame(id, name, class, mean, stringsAsFactors = FALSE)
df1
```

```
##    id     name class mean
## 1   1     Luca    5A  6.0
## 2   2   Chiara    5A  7.0
## 3   3     Lisa    5A  5.0
## 4   4   Matteo    5A  6.5
## 5   5    Alice    5A  7.5
## 6   6    Marco    5B  4.5
## 7   7 Veronica    5B  9.0
## 8   8   Nicola    5B  8.0
## 9   9    Elena    5B  8.5
## 10 10  Daniele    5B  7.0
```

b. Select the first 3 rows of `df1`.

```
df1[1:3,]
```

```
##   id   name class mean
## 1  1   Luca    5A    6
## 2  2 Chiara    5A    7
## 3  3   Lisa    5A    5
```

c. Select the last 6 rows and the first 3 columns of `df1`.

```
df1[5:10, 1:3]
```

```
##    id     name class
## 5   5    Alice    5A
## 6   6    Marco    5B
## 7   7 Veronica    5B
## 8   8   Nicola    5B
## 9   9    Elena    5B
## 10 10  Daniele    5B
```

d. Select the column `class` of `df1`.

```
df1$class
```

```
##  [1] "5A" "5A" "5A" "5A" "5A" "5B" "5B" "5B" "5B" "5B"
```

e. Convert the column `class` of `df1` in a factor with levels: “5A” and “5B”

```
df1$class <- factor(df1$class, levels = c("5A", "5B"))
df1$class
```

```
##  [1] 5A 5A 5A 5A 5A 5B 5B 5B 5B 5B
## Levels: 5A 5B
```

f. How many columns and rows `df1` has?

```
dim(df1) # or ncol(df1) and nrow(df1)
```

```
## [1] 10  4
```

g. Generate another dataframe, named `df2` composed by the columns `name` and `mean` of `df1`, specifying the argument `stringsAsFactors = FALSE`.

```
df2 <- data.frame(name = df1$name, mean=df1$mean, stringsAsFactors = FALSE)
df2
```

```
##        name mean
## 1      Luca  6.0
## 2    Chiara  7.0
## 3      Lisa  5.0
## 4    Matteo  6.5
## 5     Alice  7.5
## 6     Marco  4.5
## 7  Veronica  9.0
## 8    Nicola  8.0
## 9     Elena  8.5
## 10  Daniele  7.0
```

h. Show the first rows and the structure of `df2`.

```
head(df2)
```

```
##     name mean
## 1   Luca  6.0
## 2 Chiara  7.0
## 3   Lisa  5.0
## 4 Matteo  6.5
## 5  Alice  7.5
## 6  Marco  4.5
```

```
str(df2)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ name: chr  "Luca" "Chiara" "Lisa" "Matteo" ...
##  $ mean: num  6 7 5 6.5 7.5 4.5 9 8 8.5 7
```