

# Exercises - Class 2; Objects and classes

Sander Bossier

2023-10-10

## 1.

Create the following vectors with the function `rep()` (or its variants described on its help page):

```
rep(3, times = 5)
```

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

```
rep(c(3, 1, 2), each = 2)
```

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

```
rep(c(3, 1, 2), length.out = 5)
```

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

```
rep(c(3, 1, 2), times = c(3, 1, 2))
```

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

## 2.

Create the following vectors with the function `seq()` (or its variants described on its help page):

```
seq(from = 1, to = 9, by = 2)
```

```
## [1] 1 3 5 7 9
```

```
seq(from = 1, to = 10, along.with = c(3, 2, 1))
```

```
## [1] 1.0 5.5 10.0
```

```
seq(from = 1, to = 10, length.out = 7)
```

```
## [1] 1.0 2.5 4.0 5.5 7.0 8.5 10.0
```

```
seq(from = 10, to = 2, by = -2)
```

```
## [1] 10 8 6 4 2
```

## 3.

The function `replicate()` is sometimes wrongly used instead of `rep()` to repeat values. Use the function `system.time()` to measure the performance of both functions in repeating the value 3 for a total of  $1e6$  times.

```
system.time(  
  rep(1, times = 1e6)  
)
```

```
##      user  system elapsed
##        0        0        0
```

```
system.time(
  replicate(1e6, 1)
)
```

```
##      user  system elapsed
##    0.65    0.02    1.51
```

#### 4.

The difference between `integer` and `double` (precision floating point number) values is the byte size with which R stores them in memory. Byte size is also dependent on the attributes of the object. Use the function `object.size()` to show this for: List these six objects in increasing order of byte size in your R code.

```
obj01 <- 1:10
obj02 <- seq(1, 10, by = 1)
obj03 <- matrix(1:10, nrow = 5)
obj04 <- matrix(seq(1, 10, by = 1), nrow = 5)
obj05 <- 1:10
class(obj05) <- "humpty"
obj06 <- seq(1, 10, by = 1)
class(obj06) <- "dumpty"

print(c("obj01 size:", object.size(obj01)))
```

```
## [1] "obj01 size:" "96"
```

```
print(c("obj02 size:", object.size(obj02)))
```

```
## [1] "obj02 size:" "176"
```

```
print(c("obj03 size:", object.size(obj03)))
```

```
## [1] "obj03 size:" "264"
```

```
print(c("obj05 size:", object.size(obj05), "class:", class(obj05)))
```

```
## [1] "obj05 size:" "320"          "class:"          "humpty"
```

```
print(c("obj04 size:", object.size(obj04)))
```

```
## [1] "obj04 size:" "344"
```

```
print(c("obj06 size:", object.size(obj06), "class:", class(obj06)))
```

```
## [1] "obj06 size:" "400"          "class:"          "dumpty"
```

#### 5.

Rank the following objects/vectors according to the byte size. Which result surprises you?

```
object.size(rep(c(TRUE, FALSE), 5))
```

```
## 96 bytes
```

```
object.size(seq(1, 10, by = 1))
```

```
## 176 bytes
```

```
object.size(vector(mode = "logical", length = 10))
```

```
## 96 bytes
```

```
object.size(vector(mode = "numeric", length = 10))
```

```
## 176 bytes
```

```
object.size(vector(mode = "list", length = 10))
```

```
## 176 bytes
```

```
object.size(vector(mode = "character", length = 10))
```

```
## 232 bytes
```

6.

```
object.size(NULL)
```

```
## 0 bytes
```

```
object.size(NA)
```

```
## 56 bytes
```

```
obj07 <- c(label = NULL)
```

```
obj08 <- c(label = NA)
```

```
object.size(obj07)
```

```
## 0 bytes
```

```
object.size(obj08)
```

```
## 280 bytes
```

7.

```
as.numeric(2 * log(1i) / 1i) == pi
```

```
## [1] TRUE
```

8.

```
alphabetS4 <- setClass("alphabetS4", slots = c(symbols = "character", size = "numeric", type = "character"))
```

```
obj09 <- new("alphabetS4", symbols = letters, size = length(letters), type = "roman")
```

```
obj09
```

```
## An object of class "alphabetS4"
```

```
## Slot "symbols":
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
```

```
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
##
```

```
## Slot "size":
```

```
## [1] 26
```

```
##
```

```
## Slot "type":
## [1] "roman"

alphabetRC <- setRefClass("alphabetRC", fields = c(symbols = "character", size = "numeric", type = "character"))

obj10 <- new("alphabetRC", symbols = letters, size = length(letters), type = "roman")

obj10
```

```
## Reference class object of class "alphabetRC"
## Field "symbols":
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
## Field "size":
## [1] 26
## Field "type":
## [1] "roman"
```

## 9.

```
obj11 <- list(
  symbols = letters,
  size = length(letters),
  type = "roman"
)

obj12 <- new.env()
obj12$symbols <- letters
obj12$size <- length(letters)
obj12$type <- "roman"
```

```
object.size(obj09)
```

```
## 2736 bytes
```

```
object.size(obj10)
```

```
## 688 bytes
```

```
object.size(obj11)
```

```
## 2320 bytes
```

```
object.size(obj12)
```

```
## 56 bytes
```

## 10.

```
vecX <- c(symbols = "a", size = "1", type = "roman")
expY <- expression(vecY <- c(symbols = "a", size = "1", type = "roman"))
eval(expY)
vecX == vecY
```

```
## symbols    size    type
##      TRUE    TRUE    TRUE
```

```
identical(vecX, vecY)
```

```
## [1] TRUE
```

```
object.size(vecX)
```

```
## 608 bytes
```

```
object.size(vecY)
```

```
## 608 bytes
```

11.

```
err <- simpleError("Watch out for this!")  
object.size(err)
```

```
## 864 bytes
```

```
object.size("Watch out for this!")
```

```
## 136 bytes
```

```
attributes(err)
```

```
## $names
```

```
## [1] "message" "call"
```

```
##
```

```
## $class
```

```
## [1] "simpleError" "error"          "condition"
```

```
##12.
```

```
is.atomic(err)
```

```
## [1] FALSE
```

```
is.recursive(err)
```

```
## [1] TRUE
```