

R 语言基础：练习(一)

Table of Contents

1.数组练习	1
2.缺失值	3
3.字符向量练习	5

```
rm(list = ls(all = TRUE))
options(digits = 4)
setwd("E:\\mywork\\study_R\\R_sim_2016\\R_base_exer\\")
```

1.数组练习

Exercise 1

Create an array (3 dimensional) of 24 elements using the `dim()` function.

```
x1 <- 1:24
dim(x1) <- c(2,3,4)
```

Exercise 2

Create an array (3 dimensional) of 24 elements using the `array()` function.

```
x2 <- array(1:24,dim=c(2,3,4))
```

Exercise 3

Assign some `dimnames` of your choice to the array using the `dimnames()` function.

```
x1 <- 1:24
dim(x1) <- c(2,3,4)
dimnames(x1)<-list(c("a1", "a2"),c("b1", "b2", "b3"),c("c1", "c2", "c3", "c4"
"))
x1
```

Exercise 4

Assign some `dimnames` of your choice to the array using the arguments of the `array()` function.

```
x2 <- array(1:24,dim=c(2,3,4),dimnames =list(c("a1", "a2"),c("b1", "b2", "
b3"),c("c1", "c2", "c3", "c4")))
x2
```

Exercise 5 (不做)

Instead of column-major array, make a row-major array (transpose).

```
a <- array(1:3,c(2,3))
b <- t(a)
a
b
```

Exercise 6

For this exercise, and all that follow, download this file(ex.csv), and read it into R using the read.csv() function, e.g.: temp Copy the column named N into a new variable arr.

```
temp <- read.csv("ex.csv",header = T)
arr <- temp$N
```

Exercise 7

Set dimensions of this variable and convert it into a 3 * 2 * 4 array. Add dimnames.

```
dim(arr) <- c(3,2,4)
dimnames(arr)<-list(c("h1","h2","h3"),c("r1","r2"),c("c1","c2","c3","c4"))
arr
```

Exercise 8

Print the whole array on the screen.

```
arr
```

Exercise 9

Print only elements of height 2, assuming the first dimension represents height, the second rows and the third column.

```
arr[2,,]
```

Exercise 10

Print elements of height 1 and columns 3 and columns 1.

```
arr[1,,c(3,1)]
```

Exercise 11

Print element of height 2, column 4 and row 2.

```
arr[2,2,4]
```

Exercise 12

Repeat the exercises 9-11, but instead of using numbers to reference row, column and height, use dimnames.

```
arr[2,,]  
arr["h2",,,]  
arr[1,,c(3,1)]  
arr[1,,c("c3", "c1")]  
arr[2,2,4]  
arr["h2", "r2", "c1"]
```

2.缺失值

Exercise 1

If `X <- c(22,3,7,NA,NA,67)` what will be the output for the R statement `length(X)`

Answer: 6

Exercise 2

If `X = c(NA, 3, 14, NA, 33, 17, NA, 41)` write some R code that will remove all occurrences of NA in X.

- a. `X[!is.na(X)]`
- b. `X[is.na(X)]`
- c. `X[X==NA]= 0`

Answer: a

Exercise 3

If `Y = c(1, 3, 12, NA, 33, 7, NA, 21)` what R statement will replace all occurrences of NA with 11?

- a. `Y[Y==NA]= 11`
- b. `Y[is.na(Y)]= 11`
- c. `Y[Y==11]= NA`

Answer: b

Exercise 4

If `X = c(34, 33, 65, 37, 89, NA, 43, NA, 11, NA, 23, NA)` then what will count the number of occurrences of NA in X?

- a. `sum(X==NA)`
- b. `sum(X == NA, is.na(X))`
- c. `sum(is.na(X))`

Answer: C

Exercise 5

Consider the following vector `W <- c (11, 3, 5, NA, 6)` Write some R code that will return TRUE for value of W missing in the vector.

```
W <- c (11, 3, 5, NA, 6)
is.na(W)
```

Exercise 6

Load Orange dataset from R using the command `data(Orange)` . Replace all values of `age=118` to NA.

```
data(Orange)
Orange$age[Orange$age == 118] <- NA
```

Exercise 7

Consider the following vector `A <- c (33, 21, 12, NA, 7, 8)` Write some R code that will calculate the mean of A without the missing value.

```
A <- c (33, 21, 12, NA, 7, 8)
mean(A, na.rm = TRUE)
```

Exercise 8

Let:

```
c1 <- c(1,2,3,NA) ;
c2 <- c(2,4,6,89) ;
c3 <- c(45,NA,66,101)
```

If `X <- rbind (c1,c2,c3, deparse.level=1)` , write a code that will display all rows with missing values.

```
c1 <- c(1,2,3,NA)
c2 <- c(2,4,6,89)
c3 <- c(45,NA,66,101)
X <- rbind (c1,c2,c3, deparse.level=1)
X[!complete.cases(X), ]
```

Exercise 9

Consider the following data obtained from `df <- data.frame (Name = c(NA, "Joseph", "Martin", NA, "Andrea"), Sales = c(15, 18, 21, 56, 60), Price = c(34, 52, 21, 44, 20), stringsAsFactors = FALSE)` Write some R code that will return a data frame which removes all rows with NA values in Name column

```
df <- data.frame(Name = c(NA, "Joseph", "Martin", NA, "Andrea"),
Sales = c(15, 18, 21, 56, 60),
Price = c(34, 52, 21, 44, 20),
```

```
stringsAsFactors = FALSE)
df[!is.na(df$Name), ]
```

Exercise 10

Consider the following data obtained from `df <- data.frame(Name = c(NA, "Joseph", "Martin", NA, "Andrea"), Sales = c(15, 18, 21, NA, 60), Price = c(34, 52, 33, 44, NA), stringsAsFactors = FALSE)` Write some R code that will remove all rows with NA values and give the following output:

```
Name Sales Price
2 Joseph 18 52
3 Martin 21 33
```

```
df <- data.frame(Name = c(NA, "Joseph", "Martin", NA, "Andrea"), Sales =
  c(15, 18, 21, NA, 60), Price = c(34, 52, 33, 44, NA), stringsAsFactors =
  FALSE)
df[!(is.na(df$Name) | is.na(df$Sales) | is.na(df$Price)), ]
```

3. 字符向量练习

Exercise 1

If `x <- "Good Morning! "`, find out the number of characters in X

- a. 1
- b. 14
- c. 13

```
x <- "Good Morning! "
nchar(x)
## [1] 13
# (Answer: c)
```

Exercise 2

Consider the character vector `x <- c ("Nature's", "Best ")`, how many characters are there in x?

- a. 12
- b. 13
- c. 8,5

```
x <- c ("Nature's", "Best ")
nchar(x)
## [1] 8 5
# (Answer: c)
```

Exercise 3

If `x <- c("Nature's", " At its best ")`, how many characters are there in x?

- a. 19
- b. 8, 13
- c. 8, 9

```
x <- c("Nature's", " At its best ")
nchar(x)
## [1] 8 13
# (Answer: b)
```

Exercise 4

If `fname <- "James"` and `lname <- "Bond"`, write some R code that will produce the output "James Bond".

```
fname <- "James"
lname <- "Bond"
paste(fname, lname)
## [1] "James Bond"
```

Exercise 5

If `m <- "Capital of America is Washington"` then extract the string "Capital of America" from the character vector `m`.

```
m <- "Capital of America is Washington"
substr(m, start=1, stop=18)
## [1] "Capital of America"
```

Exercise 6

Write some R code to replace the first occurrence of the word "failed" with "failure" in the string "Success is not final, failed is not fatal".

```
x <- "Success is not final, failed is not fatal"
sub("failed", "failure", x)
## [1] "Success is not final, failure is not fatal"
```

Exercise 7

Consider two character vectors:

```
Names <- c("John", "Andrew", "Thomas") and
Designation <- c("Manager", "Project Head", "Marketing Head")
```

Write some R code to obtain the following output:

```
Names Designation
1 John Manager
2 Andrew Project Head
3 Thomas Marketing Head
```

```
Names <- c("John", "Andrew", "Thomas")
Designation <- c("Manager", "Project Head", "Marketing Head")
data.frame(Names, Designation)
##   Names      Designation
## 1  John          Manager
## 2 Andrew    Project Head
## 3 Thomas Marketing Head
```

Exercise 8

Write some R code that will initialise a character vector with fixed length of 10.

```
vector(mode="character", length=10)
## [1] "" "" "" "" "" "" "" "" "" ""
```

Exercise 9

Write some R code that will generate a vector with the following elements, without using loops. "aa" "ba" "ca" "da" "ea" "ab" "bb" "cb" "db" "eb" "ac" "bc" "cc" "dc" "ec" "ad" "bd" "cd" "dd" "ed" "ae" "be" "ce" "de" "ee"

```
c(outer(letters[1:5], letters[1:5], FUN=paste, sep=""))
## [1] "aa" "ba" "ca" "da" "ea" "ab" "bb" "cb" "db" "eb" "ac" "bc" "cc"
## [2] "dc"
## [15] "ec" "ad" "bd" "cd" "dd" "ed" "ae" "be" "ce" "de" "ee"
```

Exercise 10

Let `df <- data.frame(Date = c("12/12/2000 12:11:10"))`. Write some R code that will convert the given date to character values and gives the following output: "2000-12-12 12:11:10 CST"

```
df <- data.frame(Date = c("12/12/2000 12:11:10"))
strptime(df$Date, "%m/%d/%Y %H:%M:%S")
## [1] "2000-12-12 12:11:10 CST"
```

返回 [课程主页](#) 。