

talk02 练习与作业

目录

0.1	练习和作业说明	1
0.2	talk02 内容回顾	1
0.3	练习与作业：用户验证	2
0.4	练习 1: vector 的基本类型与简单算术	2
0.5	练习 2: vector 操作	6
0.6	练习 3: 逻辑检验和运算	8
0.7	9
0.8	练习 4: matrix、计算及相关函数	9
0.9	练习 5: 特别值	10

0.1 练习和作业说明

将相关代码填写入以 “{r}” 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的”Knit” 按键生成 PDF 文档；

将生成的 PDF 改为：姓名-学号-talk02 作业.pdf，并提交到老师指定的平台/钉群。

0.2 talk02 内容回顾

- R language basic
 - 基本数据类型
 - 简单算术

– 特别值

0.3 练习与作业：用户验证

请运行以下命令，验证你的用户名。

如你当前用户名不能体现你的真实姓名，请改为拼音后再运行本作业！

```
Sys.info()[["user"]]
```

```
## [1] "wchen"
```

```
Sys.getenv("HOME")
```

```
## [1] "/Users/wchen"
```

0.4 练习 1: vector 的基本类型与简单算术

- 用 `class` 命令确定以下 `vector` 的类型；

```
c(100, 20, 30)
```

```
c(" 字符串", " 数组", " 是我")
```

```
c(TRUE, FALSE, TRUE, T, F)
```

```
## 将代码写在此处，并运行，比如：
```

```
class(c(100, 20, 30));
```

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

-
- 用 `class` 命令确定以下 `vector` 的类型；

```
c(45, TRUE, 20, FALSE, -100)
```

```
c("string a", FALSE, "string b", TRUE)
```

```
c("a string", 1.2, "another string", 1e-3)
```

```
## 将代码写在此处，并运行  
class( c(45, TRUE, 20, FALSE, -100) );
```

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

请解释为什么整个 `vector` 的结果与单个成员的类型并不完全一致?

答:

-
- 运行以下代码:

```
x <- c(10,100,1000, 10000);  
  
( y <- sqrt( x ) * 4 + 10 );
```

```
## 代码写在此处并运行
```

问: 第二行代码最外层的括号有什么作用?

答:

-
- 以下两个 `vector`，计算它们的乘积:

```
x <- c(4,6,5,7,10,9,4,15)  
  
y <- c(0,10,1,8,2,3,4,1)
```

```
## 代码写在此处并运行
```

-
- 以下两个 `vector`，计算: `a <= b`:

```
a <- c(1,5,4,3,6)
```

```
b <- c(3,5,2,1,9)
```

```
## 代码写在此处并运行
```

-
- 将函数 `dimis.numericis.characteris.logicallength` 应用到下面的 `vector`，并展示结果；

```
x <- 1:12
```

```
y <- LETTERS[1:12]
```

```
z <- c(F, T, FALSE);
```

```
## 代码写在此处并运行
```

-
- 以下两个 `vector`，计算：`which(!is.finite(x/y))`：

```
x <- c(12:4)
```

```
y <- c(0,1,2,0,1,2,0,1,2)
```

```
## 代码写在此处并运行
```

提问：请解释输出结果的含义？

答：

-
- 以下两个 `vector`，计算：`x > y`：

```
x <- letters[1:10]
```

```
y <- letters[15:24]
```

```
## 代码写在此处并运行
```

-
- 以下 `vector`：

```
x <- c(4,6,5,7,10,9,4,15)
```

计算：

```
x < 7
```

```
x < c( 8, 4 ); ## 与第二个 vector 进行
```

```
## 代码写在此处并运行
```

问：请问第二个 `vector` 成员的循环规则是什么？这种循环在 `R` 里被称为什么？

答：

- 练习阶乘和取余操作：

```
2 ^ 6
```

```
1:10 ^ 2
```

```
5 %% 2
```

```
100:110 %% 2
```

```
## 代码写在此处并运行
```

- 将函数 `is.vector` 应用到以下数据：

```
c( 8, 9, 10)
```

```
T
```

```
7
```

问：后两个的输出结果是什么？`TRUE` or `FALSE`？为什么？

答：

0.5 练习 2: vector 操作

- 合并:

```
a <- 1:3;  
b <- LETTERS[1:3];  
( ab <- c(a,b) );
```

```
## 代码写在此处并运行
```

- 用至少两个函数检测上面生成的变量 `ab` 的数据类型;

```
## 代码写在此处并运行
```

- 取 `vector` 的一部分

先生成一个 `vector`，并对其每个成员进行命名:

```
v <- 1:10;  
names( v ) <- letters[1:10];  
v; ## 显示 v 的内容
```

```
## 代码写在此处并运行
```

取部分操作:

```
v[1]; ## index based method  
v[ 2:5 ];  
v[ c(1,3,9,2,5) ];  
v[ "a" ];  
v[ c( "a", "c", "b") ];
```

注：运行上述代码，并于每次运行后，显示 `v` 的当前值；

```
## 代码写在此处并运行
```

-
- 替换

```
v[ 1 ] <- 100;  
v[2:3] <- 100;  
v[ 3:5 ] <- c( 100, 200 );  
v[ c(1, 5, 3 ) ] <- c(100, 500, 300);
```

注：运行上述代码，并于每次运行后，显示 `v` 的当前值；

```
## 代码写在此处并运行
```

-
- 在 `vector` 的后面增加一个成员；此操作会改变 `vector` 的长度；

```
a <- sample(1:20, 10);  
length(a);  
a[ length(a) + 1 ] <- 666;  
length(a);  
a;
```

```
## 代码写在此处并运行
```

-
- 以下两个 `vector` 相加，并查看结果；

```
p <- c (3, 5, 6, 8)  
q <- c (3, 3, 3)
```

```
## 代码写在此处并运行
```

-
- 取出下面 `vector` 中数据大于 20 的成员，并显示：

```
a <- sample( 1:50, 20 );
```

```
## 代码写在此处并运行
```

0.6 练习 3：逻辑检验和运算

- 用函数 `isTRUE` 计算以下数值或表达式，查看结果；

```
T | F
```

```
T & F
```

```
5 | 0
```

```
5 & 6
```

```
## 代码写在此处并运行
```

问题：为什么 `isTRUE(5)` 为 `FALSE`，`isTRUE(6)` 也为 `FALSE`，但 `isTRUE(5 & 6)` 是 `TRUE`？

答：

-
- 将 `isTRUE` 应用于以下数值，并查看结果：

```
-1
```

```
-100
```

```
0
```

```
1
```


100

```
## 代码写在此处并运行
```

0.7

0.8 练习 4: matrix、计算及相关函数

- 生成一个 matrix，并查看结果，注意 dimnames 的用法：

```
m <- matrix( c(20, 30.1, 2, 45.8, 23, 14), nrow = 2, dimnames  
= list( c("row_A", "row_B"), c("A", "B", "C")));
```

```
## 代码写在此处并运行
```

-
- 在上面生成的变量 m 上运行以下函数：

```
dim
```

```
nrow
```

```
ncol
```

```
range
```

```
summary
```

```
colnames
```

```
rownames
```

```
t
```

```
## 代码写在此处并运行
```

-
- 用代码实现以下操作：

- a. 取第一行
- b. 取第二列
- c. 同时取第三、二列，注意取的顺序；

并且，用 `class` 函数检验得到结果的数据类型；

```
## 代码写在此处并运行
```

-
- 用代码实现以下操作：

- a. 用 1-1000 之间随机数值（用 `sample` 函数取值）替换第一行；
- b. 用 1-1000 之间随机数值（用 `sample` 函数取值）替换第二列；

```
## 代码写在此处并运行
```

0.9 练习 5：特别值

- 用以下函数或命令式检测特别值构成的 `vectorsp`，报告输出结果

```
sp <- (NA, NaN, Inf, -Inf)
```

```
is.finite
```

```
! is.infinite
```

```
is.na
```

```
is.nan
```

```
## 代码写在此处并运行
```