

课后习题二

生信 2001 张子栋 2020317210101

GitHub 地址: [MarkdownNotes/R at main · Bluuur/MarkdownNotes \(github.com\)](https://github.com/Bluuur/MarkdownNotes).

1. 1. 写出元素为 3, -1.5, 3E-8 的向量

```
1 > c(3, -1.5, 3E-8)
2 [1] 3.0e+00 -1.5e+00 3.0e-08
```

2. 写出从 -1 开始每次增加 5, 长度为 10 的向量

```
1 > seq(-1, by=5, length=10)
2 [1] -1 4 9 14 19 24 29 34 39 44
```

3. 写出 c(0, 1, 2) 重复 10 次的向量

```
1 > rep(c(0:2), 10)
2 [1] 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
```

2. 创建向量 `x=rnorm(100, 2, 1)`, 计算其和, 平均值及方差

```
1 > x = rnorm(100, 2, 1)
2 > sum(x) # 求和
3 [1] 207.045
4 > mean(x) # 均值
5 [1] 2.07045
6 > var(x) # 方差
7 [1] 0.8514727
```

1. 用 `tail` 查看其最后 10 个元素

```
1 > tail(x, n=10)
2 [1] 3.1259876 3.0699540 0.8325109 2.2028783 1.9608977 0.7154083
   1.5189217
3 [8] 2.3435189 2.3842001 2.6895156
```

2. 找出其元素大于等于 0, 小于 1 的数

```
1 > x[x >= 0 & x < 1]
2 [1] 0.5628381 0.6643946 0.4719207 0.6193675 0.9748905 0.6987171
   0.5357305
3 [8] 0.3599669 0.9784141 0.6827296 0.3502964 0.8325109 0.7154083
```

3. 找出小于 0 的元素的下标

无小于 0 的元素

```
1 > x[x < 0]
2 numeric(0)
```

3. 查看逻辑操作 `&` 和 `&&` 的帮助文件

1. 两者的区别是什么?

`&` 作用在对象的每一个元素上, `&&` 只作用在对象第一个元素上

2. 各自运行一个 example

```
1 > x[(x>0) & (x<1)]
2 [1] 0.5628381 0.6643946 0.4719207 0.6193675 0.9748905 0.6987171
   0.5357305
3 [8] 0.3599669 0.9784141 0.6827296 0.3502964 0.8325109 0.7154083
4 > x[(x>0) && (x<1)]
5 numeric(0)
6 > x[1]
7 [1] 1.829205
```

4. 对以下函数, 阅读 help 文件, 并试运行其中的 2 个 example

1. `matrix`

```
1 > matrix(1:10)
2      [,1]
3 [1,]    1
4 [2,]    2
5 [3,]    3
6 [4,]    4
7 [5,]    5
8 [6,]    6
9 [7,]    7
10 [8,]    8
11 [9,]    9
12 [10,]   10
13 > is.matrix(as.matrix(1:10))
14 [1] TRUE
```

2. `read.table`

其中两个重要参数 `comment` 和 `header` 分别代表什么

- `header = TRUE` 表示将导入文件的第一行内容作为字段名
- 默认情况下, 该函数将 `#` 作为注释的标识符, 会忽略其后内容
`comment.char =` 可以指定注释标识符
- 运行 example:

```
1 > read.table(header = TRUE, text = "
2 + a b
3 + 1 2
4 + 3 4
5 + ")
6      a b
7 1 1 2
8 2 3 4
9 > test1 <- c(1:5, "6,7", "8,9,10")
10 > tf <- tempfile()
```

```

11 > writeLines(test1, tf)
12 >
13 > read.csv(tf, fill = TRUE) # 1 column
14 x1
15 1 2
16 2 3
17 3 4
18 4 5
19 5 6
20 6 7
21 7 8
22 8 9
23 9 10
24 > ncol <- max(count.fields(tf, sep = ","))
25 > read.csv(tf, fill = TRUE, header = FALSE,
26 +         col.names = paste0("v", seq_len(ncol)))
27 v1 v2 v3
28 1 1 NA NA
29 2 2 NA NA
30 3 3 NA NA
31 4 4 NA NA
32 5 5 NA NA
33 6 6 7 NA
34 7 8 9 10
35 > unlink(tf)

```

3. file.choose

1. 试用 `read.table` 和 `file.choose` 读入一个表格文件

```

1 > read.table(file.choose())
2      v1 v2 v3 v4
3 1      Name Sex Age Height
4 2    Alice  F  13   56.5
5 3    Becka  F  13   65.3
6 4     Gail  F  14   64.3
7 5    Karen  F  12   56.3
8 6    Kathy  F  12   59.8
9 7     Mary  F  15   66.5
10 8    Sandy  F  11   51.3
11 9   Sharon  F  15   62.5
12 10   Tammy  F  14   62.8
13 11  Alfred  M  14   69.0
14 12    Duke  M  14   63.5
15 13   Guido  M  15   67.0
16 14   James  M  12   57.3
17 15 Jeffrey  M  13   62.5
18 16    John  M  12   59.0
19 17  Philip  M  16   72.0
20 18  Robert  M  12   64.8
21 19  Thomas  M  11   57.5
22 20 William  M  15   66.5

```

5. 生成以下矩阵 A (不要用 `c`), 并用 `apply` 命令按行求和

$$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 9 & 11 & 13 & 15 \\ 17 & 19 & 21 & 23 \end{bmatrix}$$

```

1 > A = matrix((0:11+1:12), ncol = 4, byrow = TRUE)
2 > A
3      [,1] [,2] [,3] [,4]
4 [1,]    1    3    5    7
5 [2,]    9   11   13   15
6 [3,]   17   19   21   23
7 > apply(A,1,sum)
8 [1] 16 48 80

```