R 练习（二）

王宁宁

2015 年 11 月 27 日

```
rm(list = ls(all = TRUE))
options(digits = 4  )
```

第 1 题

第 n 个三角形数表示为 n * (n + 1) / 2。创建一个包含前 20 个三角形数的序列。R 有一个内置常数 letters，它包含小写的罗马字母。使用前 20 个英文字母来给你刚刚创建的向量命名。

```
##   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q 
  r 
##   1   3   6  10  15  21  28  36  45  55  66  78  91 105 120 136 153 
171 
##   s   t 
## 190 210 
```

第 2 题

使用函数 diag 和, 以序列 10 到 0 到 10（即 10,...,1,0,1,...,10）为对角元素创建一个 21×21 的矩阵。

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [,13]
##  [1,]   10    0    0    0    0    0    0    0    0     0     0     0
    0
##  [2,]    0    9    0    0    0    0    0    0    0     0     0     0
    0
##  [3,]    0    0    8    0    0    0    0    0    0     0     0     0
    0
##  [4,]    0    0    0    7    0    0    0    0    0     0     0     0
    0
##  [5,]    0    0    0    0    6    0    0    0    0     0     0     0
    0
```

```
##  [6,]    0    0    0    0    0    5    0    0    0     0     0     0
     0
##  [7,]    0    0    0    0    0    0    4    0    0     0     0     0
     0
##  [8,]    0    0    0    0    0    0    0    3    0     0     0     0
     0
##  [9,]    0    0    0    0    0    0    0    0    2     0     0     0
     0
## [10,]    0    0    0    0    0    0    0    0    0     1     0     0
     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     1
     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0
     2
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
##  [1,]     0     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0     0
## [11,]     0     0     0     0     0     0     0     0
## [12,]     0     0     0     0     0     0     0     0
## [13,]     0     0     0     0     0     0     0     0
## [14,]     3     0     0     0     0     0     0     0
## [15,]     0     4     0     0     0     0     0     0
## [16,]     0     0     5     0     0     0     0     0
## [17,]     0     0     0     6     0     0     0     0
## [18,]     0     0     0     0     7     0     0     0
## [19,]     0     0     0     0     0     8     0     0
## [20,]     0     0     0     0     0     0     9     0
## [21,]     0     0     0     0     0     0     0    10
```

第 3 题

创建一个主对角线元素都为 1 的 20×21 的矩阵。

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [,13]
##  [1,]    1    0    0    0    0    0    0    0    0     0     0     0
     0
##  [2,]    0    1    0    0    0    0    0    0    0     0     0     0
     0
##  [3,]    0    0    1    0    0    0    0    0    0     0     0     0
     0
##  [4,]    0    0    0    1    0    0    0    0    0     0     0     0
     0
##  [5,]    0    0    0    0    1    0    0    0    0     0     0     0
     0
##  [6,]    0    0    0    0    0    1    0    0    0     0     0     0
     0
##  [7,]    0    0    0    0    0    0    1    0    0     0     0     0
     0
##  [8,]    0    0    0    0    0    0    0    1    0     0     0     0
     0
##  [9,]    0    0    0    0    0    0    0    0    1     0     0     0
     0
## [10,]    0    0    0    0    0    0    0    0    0     1     0     0
     0
## [11,]    0    0    0    0    0    0    0    0    0     0     1     0
     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     1
     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0
     1
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0
```

```
     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
##  [1,]     0     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0     0
## [11,]     0     0     0     0     0     0     0     0
## [12,]     0     0     0     0     0     0     0     0
## [13,]     0     0     0     0     0     0     0     0
## [14,]     1     0     0     0     0     0     0     0
## [15,]     0     1     0     0     0     0     0     0
## [16,]     0     0     1     0     0     0     0     0
## [17,]     0     0     0     1     0     0     0     0
## [18,]     0     0     0     0     1     0     0     0
## [19,]     0     0     0     0     0     1     0     0
## [20,]     0     0     0     0     0     0     1     0
```

在此矩阵之上加一行全零元素来创建一个 21×21 的方阵，原来主对角线上的全 1 元素现在全体向下偏移一行。

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [,13]
##  [1,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##  [2,]    1    0    0    0    0    0    0    0    0     0     0     0
     0
##  [3,]    0    1    0    0    0    0    0    0    0     0     0     0
     0
##  [4,]    0    0    1    0    0    0    0    0    0     0     0     0
     0
##  [5,]    0    0    0    1    0    0    0    0    0     0     0     0
     0
##  [6,]    0    0    0    0    1    0    0    0    0     0     0     0
     0
##  [7,]    0    0    0    0    0    1    0    0    0     0     0     0
     0
##  [8,]    0    0    0    0    0    0    1    0    0     0     0     0
     0
##  [9,]    0    0    0    0    0    0    0    1    0     0     0     0
     0
```

```
## [10,]    0    0    0    0    0    0    0    0    1     0     0     0
     0
## [11,]    0    0    0    0    0    0    0    0    0     1     0     0
     0
## [12,]    0    0    0    0    0    0    0    0    0     0     1     0
     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     1
     0
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0
     1
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##        [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
##  [1,]     0     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0     0
## [11,]     0     0     0     0     0     0     0     0
## [12,]     0     0     0     0     0     0     0     0
## [13,]     0     0     0     0     0     0     0     0
## [14,]     0     0     0     0     0     0     0     0
## [15,]     1     0     0     0     0     0     0     0
## [16,]     0     1     0     0     0     0     0     0
## [17,]     0     0     1     0     0     0     0     0
## [18,]     0     0     0     1     0     0     0     0
## [19,]     0     0     0     0     1     0     0     0
## [20,]     0     0     0     0     0     1     0     0
## [21,]     0     0     0     0     0     0     1     0
```

创建另一个类似的矩阵，使主对象线上的全 1 元素往上偏移一行。

```
identity_21_by_20 <- diag(rep.int(1, 20), 21, 20)
above_the_diagonal <- cbind(0, identity_21_by_20)
above_the_diagonal
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [,13]
##  [1,]    0    1    0    0    0    0    0    0    0     0     0     0
     0
##  [2,]    0    0    1    0    0    0    0    0    0     0     0     0
     0
##  [3,]    0    0    0    1    0    0    0    0    0     0     0     0
     0
##  [4,]    0    0    0    0    1    0    0    0    0     0     0     0
     0
##  [5,]    0    0    0    0    0    1    0    0    0     0     0     0
     0
##  [6,]    0    0    0    0    0    0    1    0    0     0     0     0
     0
##  [7,]    0    0    0    0    0    0    0    1    0     0     0     0
     0
##  [8,]    0    0    0    0    0    0    0    0    1     0     0     0
     0
##  [9,]    0    0    0    0    0    0    0    0    0     1     0     0
     0
## [10,]    0    0    0    0    0    0    0    0    0     0     1     0
     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     1
     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     0
     1
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
##  [1,]     0     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0     0
## [11,]     0     0     0     0     0     0     0     0
## [12,]     0     0     0     0     0     0     0     0
## [13,]     1     0     0     0     0     0     0     0
## [14,]     0     1     0     0     0     0     0     0
## [15,]     0     0     1     0     0     0     0     0
## [16,]     0     0     0     1     0     0     0     0
## [17,]     0     0     0     0     1     0     0     0
## [18,]     0     0     0     0     0     1     0     0
## [19,]     0     0     0     0     0     0     1     0
## [20,]     0     0     0     0     0     0     0     1
## [21,]     0     0     0     0     0     0     0     0
```

把这两个矩阵相加，然后再与上面 练习中的答案相加。所得的矩阵被称为 Wilkinson 矩阵。计算 Wilkinson 矩阵的特征值。

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [,13]
##  [1,]   10    1    0    0    0    0    0    0    0     0     0     0
     0
##  [2,]    1    9    1    0    0    0    0    0    0     0     0     0
     0
##  [3,]    0    1    8    1    0    0    0    0    0     0     0     0
     0
##  [4,]    0    0    1    7    1    0    0    0    0     0     0     0
     0
##  [5,]    0    0    0    1    6    1    0    0    0     0     0     0
     0
##  [6,]    0    0    0    0    1    5    1    0    0     0     0     0
     0
##  [7,]    0    0    0    0    0    1    4    1    0     0     0     0
     0
##  [8,]    0    0    0    0    0    0    1    3    1     0     0     0
     0
##  [9,]    0    0    0    0    0    0    0    1    2     1     0     0
     0
## [10,]    0    0    0    0    0    0    0    0    1     1     1     0
     0
## [11,]    0    0    0    0    0    0    0    0    0     1     0     1
     0
## [12,]    0    0    0    0    0    0    0    0    0     0     1     1
     1
```

```
## [13,]    0    0    0    0    0    0    0    0    0     0     0     1
     2
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0
     1
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0
     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
##  [1,]     0     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0     0
## [11,]     0     0     0     0     0     0     0     0
## [12,]     0     0     0     0     0     0     0     0
## [13,]     1     0     0     0     0     0     0     0
## [14,]     3     1     0     0     0     0     0     0
## [15,]     1     4     1     0     0     0     0     0
## [16,]     0     1     5     1     0     0     0     0
## [17,]     0     0     1     6     1     0     0     0
## [18,]     0     0     0     1     7     1     0     0
## [19,]     0     0     0     0     1     8     1     0
## [20,]     0     0     0     0     0     1     9     1
## [21,]     0     0     0     0     0     0     1    10
```

```
##  [1] 10.7462 10.7462  9.2107  9.2107  8.0389  8.0389  7.0040  7.0040
##  [9]  6.0002  6.0002  5.0002  4.9998  4.0044  3.9960  3.0431  2.9611
## [17]  2.1302  1.7893  0.9475  0.2538 -1.1254
```

第 4 题

创建一个列表变量，它的第一个元素包含所有从 0 到 9 的平方数，第二个元素为 10 至 19 之内的所有平方数，依此类推，最后一个元素为 90 到 99 之内的平方数。不是平方数的元素也应该被包含在内！

```
## $`0 to 9`
## [1] 0 1 4 9
##
## $`10 to 19`
## [1] 16
##
## $`20 to 29`
## [1] 25
##
## $`30 to 39`
## [1] 36
##
## $`40 to 49`
## [1] 49
##
## $`50 to 59`
## NULL
##
## $`60 to 69`
## [1] 64
##
## $`70 to 79`
## NULL
##
## $`80to 89`
## [1] 81
##
## $`90 to 99`
## NULL

## $`[0,10)`
## [1] 0 1 4 9
##
## $`[10,20)`
## [1] 16
##
## $`[20,30)`
## [1] 25
##
## $`[30,40)`
## [1] 36
##
## $`[40,50)`
## [1] 49
##
## $`[50,60)`
## integer(0)
##
## $`[60,70)`
## [1] 64
##
## $`[70,80)`
## integer(0)
##
## $`[80,90]`
## [1] 81
```

第 5 题

R 有几个内置的数据集，其中包括著名的由安德森和费舍尔在 20 世纪 30 年代收集和分析的 iris（指鸢尾花，而不是虹膜）数据。输入 iris 即可看到数据集。创建一个新的数据框，它由 iris 数据集的数值列组成；计算各列的平均值。

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        5.843        3.057        3.758        1.199
```

第 6 题

beaver1 和 beaver2 数据集包含两个海狸的体温数据。为 beaver1 数据集添加一列名为 id 的列，其值全部为 1。同样，也为 beaver2 添加一个 id 列，值全为 2。垂直拼接两个数据框，并且找到所有活跃着的海狸的子集。

```
##     day time  temp activ id
## 54  346 1730 37.07     1  1
## 68  346 1950 37.10     1  1
## 80  346 2150 37.53     1  1
## 83  346 2230 37.25     1  1
## 86  346 2300 37.24     1  1
## 114 347  340 37.15     1  1
## 153 307 1550 37.98     1  2
## 154 307 1600 38.02     1  2
## 155 307 1610 38.00     1  2
## 156 307 1620 38.24     1  2
## 157 307 1630 38.10     1  2
## 158 307 1640 38.24     1  2
## 159 307 1650 38.11     1  2
## 160 307 1700 38.02     1  2
## 161 307 1710 38.11     1  2
## 162 307 1720 38.01     1  2
## 163 307 1730 37.91     1  2
## 164 307 1740 37.96     1  2
## 165 307 1750 38.03     1  2
## 166 307 1800 38.17     1  2
## 167 307 1810 38.19     1  2
## 168 307 1820 38.18     1  2
## 169 307 1830 38.15     1  2
## 170 307 1840 38.04     1  2
## 171 307 1850 37.96     1  2
## 172 307 1900 37.84     1  2
## 173 307 1910 37.83     1  2
## 174 307 1920 37.84     1  2
## 175 307 1930 37.74     1  2
## 176 307 1940 37.76     1  2
## 177 307 1950 37.76     1  2
## 178 307 2000 37.64     1  2
## 179 307 2010 37.63     1  2
## 180 307 2020 38.06     1  2
## 181 307 2030 38.19     1  2
## 182 307 2040 38.35     1  2
## 183 307 2050 38.25     1  2
## 184 307 2100 37.86     1  2
## 185 307 2110 37.95     1  2
## 186 307 2120 37.95     1  2
## 187 307 2130 37.76     1  2
## 188 307 2140 37.60     1  2
## 189 307 2150 37.89     1  2
## 190 307 2200 37.86     1  2
## 191 307 2210 37.71     1  2
## 192 307 2220 37.78     1  2
## 193 307 2230 37.82     1  2
## 194 307 2240 37.76     1  2
## 195 307 2250 37.81     1  2
## 196 307 2300 37.84     1  2
## 197 307 2310 38.01     1  2
## 198 307 2320 38.10     1  2
## 199 307 2330 38.15     1  2
## 200 307 2340 37.92     1  2
## 201 307 2350 37.64     1  2
## 202 308    0 37.70     1  2
## 203 308   10 37.46     1  2
## 204 308   20 37.41     1  2
## 205 308   30 37.46     1  2
## 206 308   40 37.56     1  2
## 207 308   50 37.55     1  2
## 208 308  100 37.75     1  2
## 209 308  110 37.76     1  2
## 210 308  120 37.73     1  2
## 211 308  130 37.77     1  2
## 212 308  140 38.01     1  2
## 213 308  150 38.04     1  2
## 214 308  200 38.07     1  2
```

第 7 题

显示 pi 的 32 位有效数字

```
## [1] "3.1415926535897931159979634685442"
```

第 8 题

将以下字符串分割成单词，删除任何逗号或连字符：

```
x <- c(
"Swan swam over the pond, Swim swan swim!",
"Swan swam back again - Well swum swan!"
)
```

```
## [[1]]
## [1] "Swan"   "swam"   "over"   "the"    "pond"   "Swim"   "swan"   "swim!"
##
## [[2]]
## [1] "Swan"   "swam"   "back"   "again"  "Well"   "swum"   "swan!"
```

```
## [[1]]
## [1] "Swan"   "swam"   "over"   "the"    "pond"   "Swim"   "swan"   "swim!"
##
## [[2]]
## [1] "Swan"   "swam"   "back"   "again"  "Well"   "swum"   "swan!"
```

第 9 题

这是著名的 sea shells 绕口令：

```
sea_shells <- c(
"She", "sells", "sea", "shells", "by", "the", "seashore",
"The", "shells", "she", "sells", "are", "surely", "seashells",
"So", "if", "she", "sells", "shells", "on", "the", "seashore",
"I'm", "sure", "she", "sells", "seashore", "shells"
)
```

使用 nchar 函数来计算每个单词的字母数。现在，循环遍历所有可能的单词长度，找出所有与其长度相等的单词。例如，长度为 6 的单词应该有 shell 和 surely，它们都有六个字母。

```
## These words have 2 letters:

## [1] "by, So, if, on"

## These words have 3 letters:

## [1] "She, sea, the, The, she, are, I'm"

## These words have 4 letters:

## [1] "sure"

## These words have 5 letters:

## [1] "sells"

## These words have 6 letters:

## [1] "shells, surely"

## These words have 7 letters:

## [1] ""

## These words have 8 letters:

## [1] "seashore"

## These words have 9 letters:

## [1] "seashells"
```

第 10 题

解析披头士的出生日期，并使用“AbbreviatedWeekday DAYOFMONTH Abbreviated-MonthNameTwoDigitYear”的形式（例如，周三 09 十月 40）把它们打印出来。他们的出生日期列于下表。

披头士乐队	成员出生日期
Ringo Starr	1940-07-07
John Lennon	1940-10-09
Paul McCartney	1942-06-18
George Harrison	1943-02-25

```
## [1] "周日 07 七月 40" "周三 09 十月 40" "周四 18 六月 42" "周四 25 二月
 43"
```

第 11 题

编写一个函数 `zodiac_sign` ，它以日期为输入参数，并能返回对应于当天的星座。每个星座的日期范围列于下表。

星座	起始日期	结束日期
白羊座	3 月 21 日	4 月 19 日
金牛座	4 月 20 日	5 月 20 日
双子座	5 月 21 日	6 月 20 日
巨蟹座	6 月 21 日	7 月 22 日
狮子座	7 月 23 日	8 月 22 日
处女座	8 月 23 日	9 月 22 日
天秤座	9 月 23 日	10 月 22 日
天蝎座	10 月 23 日	11 月 21 日
射手座	11 月 22 日	12 月 21 日
摩羯座	12 月 22 日	1 月 19 日
水瓶座	1 月 20 日	2 月 18 日
双鱼座	2 月 19 日	3 月 20 日

例如输入“1473-02-10”，能输出下列：

```
## [1] "水瓶"
```

第 12 题

确保你安装了包：`learningr`

首先从 learningr 包中加载 hafu 数据集。

```r
data(hafu, package = "learningr")
head(hafu[,1:8])
```

```
##   Year         Series     Character Gender   Father    Mother  Eyes
 Hair
## 1 1963 Yuki no Taiyou         Sanae      F Japanese  American  <NA>
 <NA>
## 2 1964     Cyborg 009 Joe Shimamura      M American  Japanese brown
brown
## 3 1967      Lupin III     Lupin III      M  French? Japanese? black
black
## 4 1967   Nekome Kozou  Cat-Eyed Boy      M Japanese   Fantasy brown
 <NA>
## 5 1972      Gatchaman  Jun the Swan      F     <NA>      <NA> green
green
## 6 1974 Great Mazinger      Jun Hono      F American  Japanese black
black
```

在 Father 和 Mother 列中，有一些值在国家名后面带有问号，表明作者不确定这些父母的国籍。在 fafu 数据框中创建两个新的列，分别表示是否在 Father 或 Mother 列中带有问号。从 Father 和 Mother 列中删除那些问号。

```r
head(hafu[,10:11])
```

```
##   FathersNationalityIsUncertain MothersNationalityIsUncertain
## 1                         FALSE                         FALSE
## 2                         FALSE                         FALSE
## 3                          TRUE                          TRUE
## 4                         FALSE                         FALSE
## 5                            NA                            NA
## 6                         FALSE                         FALSE
```

hafu 数据集中每个父母的国籍都有单独的列。把数据框从宽形转换为长形，使一列为父母的国籍，一列为国籍所对应的父母是谁。

```r
colnames(hafu_long)
```

```
##  [1] "Year"                          "Series"
##  [3] "Character"                     "Gender"
##  [5] "Eyes"                          "Hair"
##  [7] "Notes"                         "FathersNationalityIsUncertain"
##  [9] "MothersNationalityIsUncertain" "variable"
## [11] "value"
```

```r
head(hafu_long[,10:11])
```

```
##   variable    value
## 1   Father Japanese
## 2   Father American
## 3   Father   French
## 4   Father Japanese
## 5   Father     <NA>
## 6   Father American
```

写一个函数，它能返回向量中 10 个最常见的值及其次数。尝试把这个函数应用于 hafu 数据集的某些列中（缺失值不计入）。

```
## x
##  Japanese   English  American    French    German   Russian   Fantas
y
##       120        29        23        20        12        10         
8
##   Italian Brazilian   Finnish 
##         4         3         2 
```