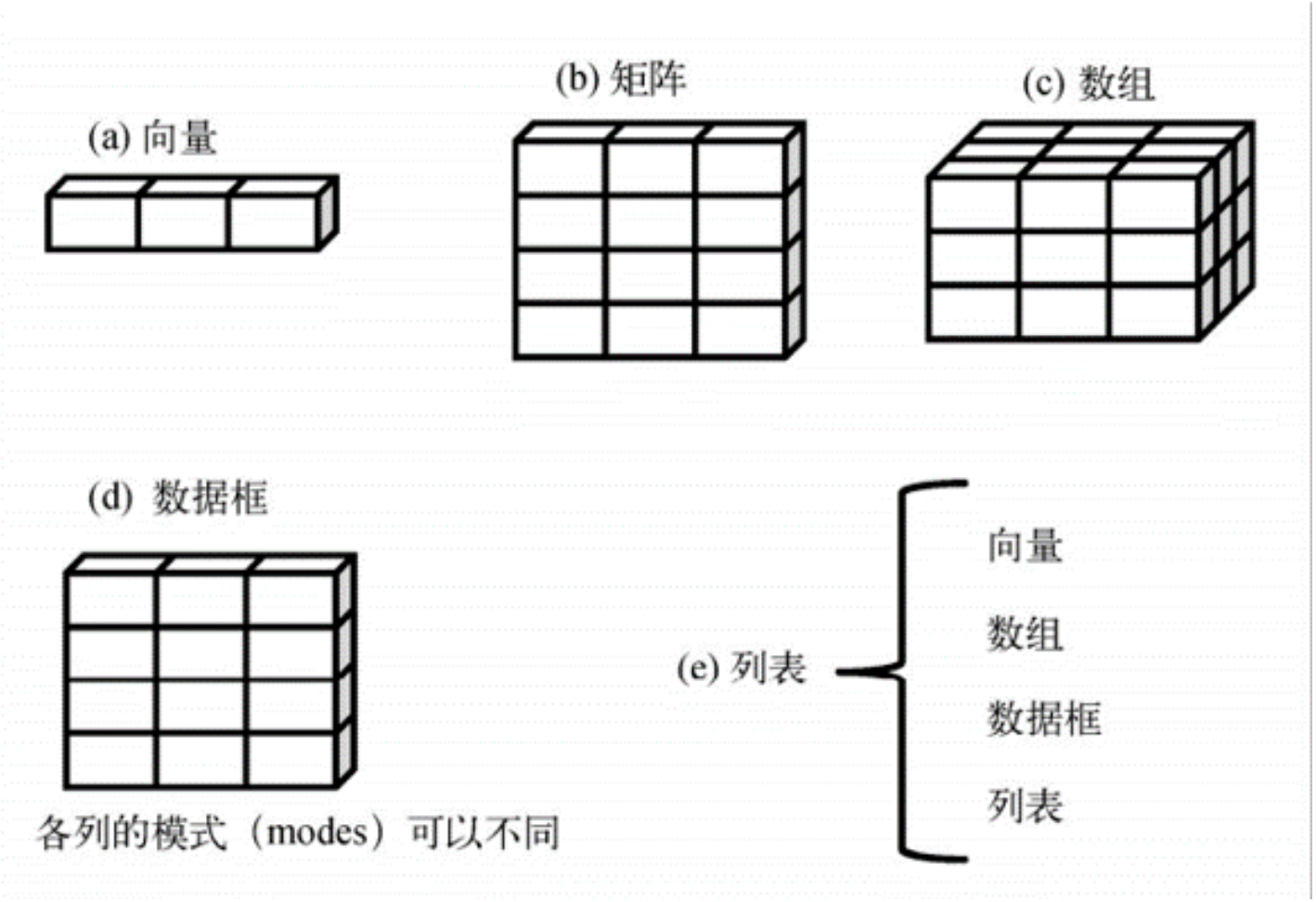
R 语言-3

郑泽靖　zzjstat2023@163.com

北京师范大学统计学院

2025 年 11 月 30 日

数据结构



列表的定义

列表可以看成向量结构的另一种推广。它允许其各个分量是任意的 R 语言结构，例如:

- 向量
- 矩阵
- 数据框
- 其他列表

列表结构能够将不同的对象以简单的形式组合在一起，方便编程者调用。

列表的特点

- 列表中的每个元素可以是不同类型的 R 对象，具有更大的灵活性。
- 列表元素可以通过索引或名称进行访问。
- 许多 R 语言函数的计算结果都是以列表的方式表达的。

生成列表

在 R 语言中，可以通过函数 `list()` 来生成列表数据。以下是示例代码：

```
1 #  生成一个列表
2 x <- list(u = 2, v = "abcd")   #  列表
3 x #  显示列表x的内容
4 $u
5 [1] 2
6 $v
7 [1] "abcd"
```

索引列表分量

在 R 语言中，可以通过 [[i]] 或 $ 来索引和修改列表的分量。

```
# 列表有两个分量：u 和 v
x <- list(u = 2, v = "abcd")
# 索引x的第一分量
x$u     # 或者使用 x[[1]]
[1] 2
# 索引x的第二分量
x[[2]]
[1] "abcd"
```

修改列表分量

```
1 # 将x的第一分量修改为1:3
2 x$u <- 1:3  # 修改为向量
3 # 显示x的第一分量的内容
4 x[[1]]
5 [1] 1 2 3
```

names 函数

在 R 语言中，函数 names 不仅能用于数据框，也能用于列表，查阅或修改各个分量的名称。

```
# 创建列表
x <- list(u = 2, v = "abcd")
names(x)
[1] "u"      "v"
names(x) <- c("num", "string")
names(x)
[1] "num"      "string"
```

函数

为了方便使用者，R 语言将具有特定功能的程序代码封装在函数中。下面简单介绍一些常用函数的功能和使用方法。

1. 求和函数

函数 sum 计算向量的各个分量之和，矩阵的各行（列）元素之和使用函数 rowSums 和 colSums。

```
# 计算向量c(1, 2, 3)的各分量之和
sum(c(1, 2, 3))  # 结果为 6
a <- matrix(1:6, 2, 3, byrow = TRUE)
# 计算 a 的各列之和，结果为3维行向量
b <- colSums(a)
# 计算 a 的各行之和，结果为2维行向量
c <- rowSums(a)
```

2. 函数的输入变量

通常，R 语言的数学计算函数支持数据形式的输人变量（自变量），也支持向量形式的输入变量，还支持矩阵形式的输入变量. 例如，运行程序代码

```
x <- seq(1, 11, 2)
sin(x)
[1]  0.8414710  0.1411200 -0.9589243
[4]  0.65698660 0.4121185 -0.9999902
```

3. 绘图功能

R 语言提供了方便的制图功能，可以绘制任何函数的图像。函数 plot 提供了一种绘制平面图形的功能，可以通过问号 ? 来获取该函数的在线帮助。

```
1 # 创建从0到10的数值序列，步长为0.1
2 x <- seq(0, 10, 0.1)
3 # 计算x的正弦值
4 y <- sin(x)
5 # 绘制函数曲线
6 plot(x,y,type="l",main="y=sin(x)",
7 xlab = "x", ylab = "y")
```

关系运算符

在 R 语言中，关系运算符的计算结果是逻辑数据，表明参与运算的两个变量是否满足特定的关系。不仅两个数之间可以进行关系运算，向量和数之间、矩阵和数之间也可以进行关系运算。

```
# 将向量 1:5 赋值给 X
X <- 1:5
# 判断 X 的各个分量是否小于或等于 4
X <= 4
[1] TRUE TRUE TRUE TRUE FALSE
```

关系运算符的应用

可以利用关系运算符显示向量中满足特定条件的分量，或将这些分量统一改为特定的值。

```
X <- 1:5
# 显示 X 中小于 3 的分量构成的子向量
X[X < 3]
[1] 1 2
# 将 X 中小于 3 的分量都改为 8
X[X < 3] <- 8
X     # 再次显示 X 的结果
[1] 8 8 3 4 5
```

其他关系运算示例

```
1  # 将字符 "abc" 赋值给 b
2  b <- "abc"
3  # 将 b 赋值给 c
4  c <- b
5  # 判断b中的内容是否和字符"abc"相等
6  b == "abc"
7  [1] TRUE
8  # 判断b存储的内容是否和c的相等
9  b == c
10 [1] TRUE
```

离散均匀分布

若随机变量 X 满足:

$$\mathrm{P}(X=k)=\frac{1}{n},\quad k=1,2,\ldots,n$$

则称 X 服从以 n 为参数的离散均匀分布。

以 n 为参数的离散均匀分布随机变量的观测值称为以 n 为参数的离散均匀分布随机数。

离散均匀分布在 R 语言中的模拟

可以使用函数 sample 来模拟离散均匀分布随机数。其简单调用方式为:

```
sample(x, # 等可能地选择 x 的分量
size,  # size为要选取的变量的个数
replace = FALSE)
# replace为真, 有放回抽样;
#否则, 无放回抽样
```

逻辑运算符

逻辑运算符用于逻辑变量的运算。常用的逻辑运算符及其含义如下:

- !: 非运算
- &: 与运算
- |: 或运算

逻辑运算符示例

```
1 !TRUE      #  非运算结果为  FALSE
2 !FALSE     #  非运算结果为  TRUE
3 TRUE & TRUE    #  与运算结果为  TRUE
4 TRUE & FALSE   #  与运算结果为  FALSE
5 FALSE & FALSE  #  与运算结果为  FALSE
6 TRUE | TRUE    #  或运算结果为  TRUE
7 TRUE | FALSE   #  或运算结果为  TRUE
8 FALSE | FALSE  #  或运算结果为  FALSE
```

逻辑矩阵的示例

```
> x
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

> a <- x <= 3
> a
      [,1]  [,2]  [,3]
[1,]  TRUE  TRUE FALSE
[2,]  TRUE FALSE FALSE
```

逻辑矩阵的示例

```
> b <- x > 2
> b
      [,1]  [,2]  [,3]
[1,] FALSE FALSE  TRUE
[2,]  TRUE  TRUE  TRUE

> !a
      [,1]  [,2]  [,3]
[1,] FALSE FALSE  TRUE
[2,] FALSE  TRUE  TRUE
```

逻辑矩阵的示例

```
> a & b
      [,1]  [,2]  [,3]
[1,] FALSE FALSE FALSE
[2,]  TRUE FALSE FALSE

> a | b
      [,1]  [,2]  [,3]
[1,]  TRUE  TRUE  TRUE
[2,]  TRUE  TRUE  TRUE
```

短路逻辑或

|| 是短路逻辑或（OR）运算符，也用于将两个逻辑表达式组合在一起。它与 | 的区别在于，只要第一个表达式为 TRUE，就会返回 TRUE，并且不会评估第二个表达式。

```
1 x <- 2
2 a <- x <= 3
3 b <- x > 2
4 result <- a && b
```

短路逻辑和

&& 是短路逻辑与（AND）运算符，也用于将两个逻辑表达式组合在一起。它与 & 的区别在于，只要第一个表达式为 FALSE，就会返回 FALSE，并且不会评估第二个表达式。

```
1 x <- 2
2 a <- x <= 3
3 b <- x > 2
4 result <- a || b
```

作业

1. 使用 R 语言创建一个数据框，包含以下列：
   - ID：1 到 10 的整数
   - Name：随机给定 10 个名字（如“学生 1”，“学生 2”）
   - Score：随机生成 10 个 1 到 100 之间的整数
2. 计算数据框中所有学生的平均分，并输出平均分。
3. 从数据框中筛选出分数大于等于 70 的学生，并输出这些学生的信息。
4. 从数据框中随机抽取 5 个学生的 Score，使用不放回的方式抽样，并打印结果。

Thanks!