

# R 语言-2

郑泽靖    zzjstat2023@163.com

北京师范大学统计学院

2025 年 10 月 10 日

# 常量

---

常量是直接写在程序中的值，包括数值、字符串等。

- 数值型常量: 123, 123.45
- 字符型常量: 使用双引号或单引号包围 ("Li Ming")
- 逻辑型常量: 只有 TRUE 和 FALSE
- 缺失值: 用 NA 表示，常见于数据缺失情况。
- NaN (Not a Number): 表示未定义或不可表示的数值，如 0 除以 0。
- Inf: 表示正无穷或负无穷，通常在除以零时出现。

# 变量

---

变量用于保存输入值或计算结果。

- 变量名规则:
  - 由字母、数字、下划线和句点组成，不能以数字开头。
  - 变量名区分大小写，如 `y` 和 `Y` 是不同的变量。
  - 示例: `x`, `x1`, `X`, `cancer.tab`
- 赋值: 使用 `<-` 或 `=` 定义变量。

```
1 x5 <- 6.25  
2 x6 = sqrt(x5)
```

## 重要变量类型：逻辑型

逻辑型变量用于存储真（TRUE）和假（FALSE）的值。在数值计算中，逻辑变量值 *TRUE* 自动转换为 1，*FALSE* 自动转换为 0。

```
1 TRUE + TRUE           # 求和
2 ## [1] 2
3 a <- TRUE             # 赋值
4 is.logical(a)         # 检查类型
5 ## [1] TRUE
6 class(a)              # 查看变量类型
7 ## [1] "logical"
```

## 重要变量类型：数值型

---

数值型变量用于保存数值数据。

```
1  a <- 23                # 将23赋值给变量a
2  is.numeric(a)          # 检查是否为数值型
3  ## [1] TRUE
4
5  is.logical(a)          # 检查是否为逻辑型
6  ## [1] FALSE
7
8  class(a)               # 查看变量类型
9  ## [1] "numeric"
```

## 重要变量类型：字符型

字符型变量用于保存文本数据。

```
1 a <- "My"           # 将"My"赋值给 a
2 a                   # 显示a的值
3 ## [1] "My"
4
5 is.character(a)      # 检查是否为字符型
6 ## [1] TRUE
7
8 class(a)             # 查看变量类型
9 ## [1] "character"
```

## 变量的输出

---

- 在 R 中，直接输入变量名并运行代码，可以查看变量的值。
- 使用函数 `print()` 也可以输出变量的值。

```
1 > a <- 56
2 > a
3 [1] 56
4 > print(a)
5 [1] 56
```

# 变量的输出

cat() 函数可以更灵活地输出变量值。例如：

```
1 > x <- 5
2 > y <- 10
3 > cat("x的值是:", x, ", y的值是:", y, "\n")
4 x的值是: 5 , y的值是: 10
```

```
1 > name <- "Alice"
2 > age <- 30
3 > cat("姓名:", name, "\n", #\n用来换行
4 +     "年龄:", age, "岁\n", sep = "")
5 姓名:Alice
6 年龄:30岁
```



# 数据集

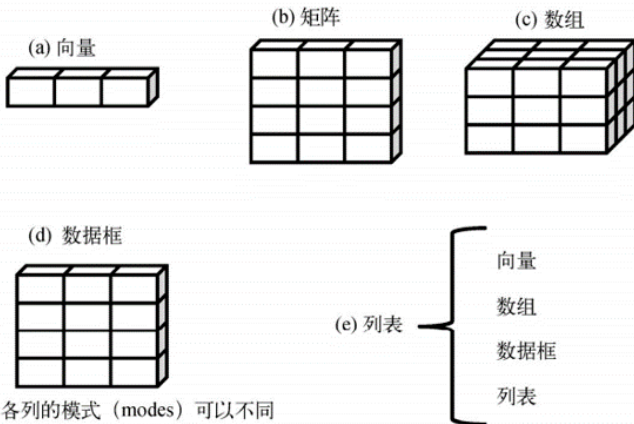
表2-1 病例数据

病人编号 (PatientID)	入院时间 (AdmDate)	年龄 (Age)	糖尿病类型 (Diabetes)	病情 (Status)
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

请观察图中的数据集，思考以下问题：

- 数据集中包含哪些变量？
- 每个变量的数据类型是什么？（例如：数值型、字符型、逻辑型等）

# 数据结构



# 向量

向量是将若干个基础类型（数值型、字符型或逻辑型）相同的值存储在一起的结构。各个元素可以通过序号进行访问。

向量的创建方法（1） - 组合功能函数：

```
1 # 数值型向量
2 x1 <- c(1, 2) # 创建第一个数值型向量
3 x2 <- c(3, 4) # 创建第二个数值型向量
4 x <- c(x1, x2) # 合并向量
5 x
6 ## [1] 1 2 3 4
```

# 向量

---

```
1 # 字符型向量
2 b <- c("one", "two", "apple")
3 # 逻辑型向量
4 d <- c(TRUE, TRUE, FALSE, TRUE)
```

## 注意事项:

- 向量中的所有元素必须是相同类型。
- 使用组合函数时，注意不要使用已定义的变量名，例如 “c”。

## 向量的创建

### 方法 (2): 冒号生成数值序列

表达式 "a:b" 从 a 到 b (步长为 1)。默认生成整数序列。示例:

```
1 > 1:6                # 1 到 6
2 [1] 1 2 3 4 5 6
3 > 4:7.6              # 整数部分
4 [1] 4 5 6 7
5 > 1.2:5              # 从 1.2 到 5
6 [1] 1.2 2.2 3.2 4.2
```

# 向量的创建

---

## 方法 (3): **Seq()** 函数

`seq(from, to, by, length.out)`

- `from`: 向量的起始值。
- `to`: 向量的结束值。
- `by`: 步长, 默认值为 1。可为负数以生成递减序列。
- `length.out`: 指定生成的向量长度, 自动计算步长。

# 向量的创建

```
1 > seq(2, 10) # 默认步长
2 [1] 2 3 4 5 6 7 8 9 10
3 > seq(2, 10, by = 2) # 指定步长
4 [1] 2 4 6 8 10
5 # 指定长度 # 生成 5 个数的等差数列
6 > seq(0, 1, length.out = 5)
7 [1] 0.0 0.25 0.5 0.75 1.0
8 # 递减序列
9 > seq(10, 2, by = -2)
10 [1] 10 8 6 4 2
```

## 访问和修改向量元素

正整数下标：使用方括号访问向量元素。示例：

```
1 x <- c(1, 4, 6.25)
2 x[2]                # 取出第二个元素
3 [1] 4
4 x[2] <- 99          # 修改第二个元素
5 print(x)
6 [1] 1.00 99.00 6.25
```



## 访问和修改向量元素

```
1 x[c(1, 3)]      # 取出第1和第3个元素
2 [1] 1.00 6.25
3 # 修改第1和第3个元素
4 x[c(1, 3)] <- c(11, 13)
5 print(x)
6 [1] 11 99 13
7 x[c(1, 3, 1)]   # 重复下标
8 [1] 11 13 11
```

## R 矩阵

矩阵：二维数组。向量可以视为特殊的矩阵（行数为 1 或列数为 1 的矩阵）。

(a) 向量



(b) 矩阵



注意：每个元素必须为相同的数据类型（数值型、字符型或逻辑型）。

# 矩阵的定义

---

使用 `matrix()` 函数定义矩阵。示例如下：

```
1 A <- matrix(11:16, nrow=3, ncol=2)
2 print(A)
3      [,1] [,2]
4 [1,]   11  14
5 [2,]   12  15
6 [3,]   13  16
```

## 矩阵的定义

- `matrix()` 函数接受一个向量作为矩阵元素。
- `nrow` 和 `ncol` 指定行数和列数。
- 默认按列填充，可使用 `byrow=TRUE` 按行填充。

```
1 B <- matrix(c(1, -1, 1, 1),  
2             nrow=2, ncol=2, byrow=TRUE)  
3 print(B)  
4      [,1] [,2]  
5 [1,]    1  -1  
6 [2,]    1    1
```

## 访问矩阵的元素

使用下标访问矩阵元素：

- `A[1,]`：取出第一行，返回普通向量。
- `A[,1]`：取出第一列，返回普通向量。
- `A[c(1, 3), 1:2]`：取出指定行和列的子矩阵。

```
1 A <- matrix(11:16, nrow=3, ncol=2)
2 A
3      [,1] [,2]
4 [1,]   11  14
5 [2,]   12  15
6 [3,]   13  16
```

# 访问矩阵的元素

```
1 A[1,]                # 取出第一行
2 [1] 11 14
3 A[,1]                # 取出第一列
4 [1] 11 12 13
5 # 取出第1和第3行的前2列
6 A[c(1, 3), 1:2]
7      [,1] [,2]
8 [1,]    11    14
9 [2,]    13    16
```

## 矩阵元素的修改

使用下标可以修改矩阵的元素：

- 使用  $A[i, j]$  访问第  $i$  行第  $j$  列的元素。
- 直接赋值可以修改该元素的值。
- 可以使用下标向量同时修改多个元素。

```
1 A[2, 1] <- 99 # 修改单个元素
2 A
3      [,1] [,2]
4 [1,]   11  14
5 [2,]   99  15
6 [3,]   13  16
```

## 矩阵元素的修改

```
1 # 修改多个元素
2 A[c(1, 3), 2] <- c(20, 30)
3 A
4      [,1] [,2]
5 [1,]   11  20
6 [2,]   99  15
7 [3,]   13  30
```



# 数据框

数据框是 R 语言中一种常见的数据结构, 用于存储表格数据。

- 由  $n$  行和  $p$  列组成。
- 各列可以包含不同类型的数据
- 同一列中的数据类型必须相同。

表2-1 病例数据

病人编号 (PatientID)	入院时间 (AdmDate)	年龄 (Age)	糖尿病类型 (Diabetes)	病情 (Status)
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

## 在 R 中创建数据框

---

使用 `data.frame()` 函数来创建数据框。所有输入变量必须是相同维度的向量。

示例代码：

```
1 x <- c(160, 175)
2 y <- c(51, 72)
3 sex <- c("Female", "Male")
4 myData <- data.frame(x, y, sex)
5 class(myData)
6 [1] "data.frame"
```

## 在 R 中创建数据框

---

创建的数据框 myData 的输出：

```
1 > myData
2       x    y    sex
3 1  160  51 Female
4 2  175  72   Male
```

## 在 R 中创建数据框并设置变量名称

---

可以在创建数据框时指定变量的名称。

```
1 myData <- data.frame(height = x,  
2                       weight = y, sex = sex)  
3 myData  
4   height weight      sex  
5 1    160     51 Female  
6 2    175     72   Male
```

## 在 R 中提取数据框的数据

---

可以通过列变量名称提取相应的数据。例如：

```
1 # 提取身高数据
2 heights <- myData$height
3 heights
4 [1] 160 175
5 # 提取性别数据
6 genders <- myData$sex
7 genders
8 Levels: Female Male
```

# 在 R 中查阅数据框列名称

---

## 查阅列名称

在 R 语言中，可以通过 `names()` 函数查阅数据框各列的名称。以下是示例代码：

```
1 myData <- data.frame(x, y, sex)
2 # 查阅各列名称
3 names(myData) # 显示 myData 各列的名称
4 [1] "x" "y" "sex"
```

## 在 R 中修改数据框列名称

修改列名称我们可以使用 `names()` 函数修改数据框的列名称：

```
1 # 重新命名 myData 的列名称
2 names(myData) <- c("h", "w", "Sex")
3 # 显示当前各列名称
4 names(myData)
5 [1] "h" "w" "Sex"
```

现在，数据框 `myData` 的列名称已更改为 `h`, `w` 和 `Sex`。

## 在 R 中索引数据框内容

在 R 语言中，可以像数组和向量一样通过方括号索引或修改数据框中特定位置的内容。

```
1 myData <- data.frame(height=x,  
2                        weight=y,sex=sex)
```

- `myData[1, 1]` 和 `myData$height[1]` 是等价的，它们都代表数据框 `myData` 第 1 行和第 1 列交叉位置的变量值。
- `myData[, 2]` 和 `myData$weight` 也是等价的，它们都表示数据框 `myData` 的第 2 列。



## 在 R 中修改数据框内容

---

可以通过以下代码修改 `myData[1, 1]` 的内容：

```
1 # 将myData[1, 1] 修改为 165
2 myData[1, 1] <- 165
3 myData$height[1] <- 165
```

# 作业

---

1. 定义一个数值变量和一个字符变量，并打印它们的值。
2. 创建一个数值向量和一个字符向量，并打印它们的值；修改数值向量的第三个元素为 10，并打印修改后的向量。
3. 创建一个  $2 \times 3$  矩阵，并打印矩阵；提取矩阵的第一行和第二列，并打印它们；将矩阵的第一行第二列的元素修改为 99，并打印修改后的矩阵。

*Thanks!*