第 3 章

R 的变量

数值类型数据可分为常量和变量两大类，在程序运行过程中，其值和类型不能被改变的量称为常量，其值和类型能被改变的量称为变量。变量的标识符（即变量名）、变量的值和变量的数据类型称为变量的三要素。显然，变量是 R 语言的对象之一。

本章涉及的内容包括：

- 变量赋值定义。
- 变量的类型。
- 特殊变量定义。
- 判别与转换变量的函数。
- R 的变量命名规则及特点。

3.1 变量赋值

在 R 语言中，变量即数据对象，对变量的赋值用“<-”或者“->”（用“=”也可以，但很少用。请注意：“=”不是“等于”，在 R 语言中“等于”是用“==”表示）。例如：

```
>x<-3 或者 3->x
>y<-z<-6
```

此时，变量 x 的值为 3，变量 y 和 z 的值为 6。

可以用 ls() 函数查看当前系统中的变量的状况，例如：

```
> ls()
[1]"x" "y" "z"
```

表明此时系统中有 3 个变量，x、y 和 z。

3.2 变量的类型

在 R 语言中变量的类型有：

- 数值型（numeric）：数值型变量还可以再划分为整数（integer）、单精度和双精度（double）3 种。
- 逻辑型（logical）：逻辑型变量的取值只能是 TRUE（或 T），或者是 FALSE（或 F）。
- 字符型（character）：字符型变量是夹在双引号 "" 或单引号 '' 之间的字符串。
- 复数型（complex）：复数型变量具有 $a+bi$ 的形式。
- 原味型（rsw）：原味型变量就是以二进制形式保存的变量。

3.3 特殊变量

在 R 语言中还有一些特殊意义的变量，它们是：

- Inf，其意义为无穷。例如，1/0 的结果为 Inf，与它相反意义的变量为 -Inf，表示负无穷。
- NaN（Not a Number），其意义为不确定。例如，0/0 的结果为 NaN。
- NA（Not Available），其意义为无法得到或缺失时，就给相应的位置赋予 NA，与 NA 变量的任何运算，其结果均为 NA。
- NULL，其意义是空的变量。

3.4 判别与转换变量的函数

在 R 语言中，各种类型的变量可以相互转换，并提供了相应的函数对于变量的类型进行判别。

【实例 3-1】变量可以相互转换示例。

```
> x<-3
> y<-as.character(x)
> y
[1] "3"
> is.numeric(y)
[1] FALSE
```

在这里 x 是一个数值型变量（其值为 3），由 as.character() 函数强制转换为字符型变量（此时为字符 3），并赋予变量 y。用 is.numeric() 函数来判别变量 y 的类型是否为数值型，返回值为假，即说明 y 不是数值型变量。

表 3-1 给出了各种判别与转换变量类型的函数。

表 3-1　R 语言中判别与转换变量类型的函数

类　型	判别函数	转换函数
数值	is.numeric()	as.numeric()
整数	is.integer()	as.integer()
双精度	is.double()	as.double()
复数	is.complex()	as.complex()
字符	is.character()	as.character()
逻辑	is.logical()	as.logical()
无穷	is.infinite()	—
有限	is.finite()	—
不确定	is.nan()	—
缺失	is.na()	—
空	is.null()	as.null()

一般来说，对于开方函数（sqrt()），其自变量只能是正实数或者是复数，负实数是不能做开方运算的。但在 R 语言中，经过特殊处理后，负实数也能做开方运算。

【实例 3-2】求负数平方根示例。

```
> sqrt(-2)
[1] NaN
Warning message:(警告信息)
In sqrt(-2) : 产生了 NaNs
```

如果一定要对负数做开方运算，那只能是在复数意义下的运算，需要先将实数改写成复数，例如：

```
>sqrt(-2+0i)
[1] 0+1.414214i
```

或者将实数强制转换成复数，例如：

```
>sqrt(as.complex(-2))
[1] 0+1.414214i
```

3.5 R 的变量命名规则

R 语言对变量的基本命名规则有：

- 不能用 R 语言的保留字如 function、if、else、while、for、NA、next、TRUE 等作为变量名。
- R 语言区分英文字母大小写，所以 backet 与 Backet、BACKET 会视为 3 个不同的变量名。
- 变量名开头必须是英文字母或点号（“.”），当以点号（“.”）开头时，接着的第二位不能是数字；对于自定义的变量名，建议使用大写英文字母开头，以避免与系统变量混淆。
- 变量名只能包含字母、数字、下画线（“_”）和点号（“.”）。

3.6 R 变量的特点

R 语言变量有着与其他编程语言不同的两大特点。

特点一：在一般编程语言中，变量必须遵循“先定义后使用”的原则，但在 R 语言中，变量在使用时可以不先定义，直接对其赋值。

【实例 3-3】已知圆的半径 R 为 5，求圆的面积。

```
>R<-5
> Area=3.14*R^2
> Area
[1] 78.5
```

特点二：R 是动态赋值语言，变量的类型可以随时改变。

【实例 3-4】变量的类型可以随时改变示例。

```
> x<-5
> x
[1] 5
> x<-c("abc")
> x
[1] "abc"
```

变量 x 先赋予整数 5，后赋予字符串“abc”，程序运行正常。

自我检测

一、判断题

（　　）1. 在 R 语言中，变量在使用时可以不先定义，直接对其赋值。

（　　）2. 在 R 语言中，变量的类型不可以随时改变。

（　　）3. 在 R 语言中，变量的类型除常见的之外，还有复数型和原味型。

（　　）4. R 语言不区分英文字母大小写，所以 backet 与 Backet、BACKET 会视为 3 个相同的变量名。

（　　）5. 在 R 语言中，负数可以在复数意义下做开方运算。

二、单选题

（　　）1. 下列哪一个函数可以在 Console 窗口列出所有变量数据？

A. ls()　　B. wm()　　C. q()　　D. getwd()

（　　）2. 下列哪一个是 R 语言不合法的变量名称？

A. x3　　B. x.3　　C. .x3　　D. 3.x

（　　）3. 下列哪一个不是 R 语言的等号符号？

A. #　　B. =　　C. < —　　D. — >

（　　）4. 执行下列命令后，*y* 的值是下面哪一个？

```
> x<-3
> y<-as.character(x)
```

A. 3　　B. -3　　C. “3”　　D. 无法确定

（　　）5. 若再执行下列命令后，输出结果是下面哪一个？

```
> is.numeric(y)
```

A. [1]3　　B. [1] “3”　　C. [1]TRUE　　D. [1] FALSE

第4章

向　　量

在R语言中，称创建和控制的实体为对象，它可以是向量、矩阵、数组、因子、数据框、列表、字符串和函数等，也可以是由这些实体定义的一般结构（structrure）。

向量（vector）是由相同基本类型的元素构成的数据系列，是R语言中的最常用的对象，同时也可以作为R语言中最基本的数据输入方式。

本章涉及的内容包括：

- 处理向量对象函数的定义。
- 向量对象的数学运算函数。

4.1 简单的处理向量对象函数

简单的处理向量对象的函数主要有 seq()、c()、rep() 等。

1. 建立向量对象函数 seq()

seq() 函数可用于建立一个规则型的数值向量对象，它的使用格式如下：

```
seq(from,to,by=width,length.out=numbers)
```

参数的意义如下。

from：数值向量对象的初始值。

to：数值向量对象的终止值。

by：每个数值向量元素的增值。如果省略 by 参数，同时没有 length.out 参数存在，则数值向量元素的增值是 1 或 −1。

length.out：用于设定 seq() 函数所需建立的数值向量元素的个数。

【实例 4-1】使用 seq() 建立规则型的数值向量对象。

数值向量元素的初始值是 1，终止值是 9，省略了参数 by，同时没有参数 length.out，则数值向量元素的增值是 1，如图 4-1 所示。

```
> seq(1,9)
[1] 1 2 3 4 5 6 7 8 9
```

图 4-1 代码及运行结果之一

数值向量元素的初始值是 1，终止值是 9，参数 by 值为 2，表示数值向量元素的增值是 2，参数 length.out 的默认值为 5，如图 4-2 所示。

```
> seq(1,9,by=2)
[1] 1 3 5 7 9
```

图 4-2 代码及运行结果之二

数值向量元素的初始值是 1，终止值是 9，参数 by 值为 pi=3.141593，所以输出的值分别为：1.000000，4.141593，7.283185。后面一数超过 9，所以不输出，如图 4-3 所示。

```
> seq(1,9,by=pi)
[1] 1.000000 4.141593 7.283185
```

图 4-3 代码及运行结果之三

数值向量元素的初始值是 1.5，终止值是 4.5，参数 by 值为 0.5，参数 length.out 的默认值为 7，如图 4-4 所示。

```
> seq(1.5,4.5,by=0.5)
[1] 1.5 2.0 2.5 3.0 3.5 4.0 4.5
```

图 4-4 代码及运行结果之四

数值向量元素的初始值是 1，终止值是 9，参数 length.out 的值为 5，参数 by 的默认值为 2，如图 4-5 所示。

```
> seq(1,9,length.out = 5)
[1] 1 3 5 7 9
```

图 4-5　代码及运行结果之五

2. 连接向量对象函数 c()

c() 函数中的 c 是 concatenate() 的缩写。这个函数不用于建立向量对象，而用于连接向量对象元素。

【实例 4-2】使用 c() 函数连接并输出一个简单的向量对象。

代码及运行情况如图 4-6 所示。

```
> x<-c(1,3,7,4,9)
> x
[1] 1 3 7 4 9
```

图 4-6　使用 c() 函数连接并输出向量

用 c() 函数连接一个元素值分别为 1、3、7、4、9 的向量对象并赋予数值型变量 x，将其输出。

3. 重复向量对象函数 rep()

如果向量对象内某些元素是重复的，则可以使用 rep() 函数建立这种类型的向量对象，它的使用格式如下。

```
rep(x,times,each,length.out)
```

参数的意义如下。

x：向量对象内可以重复的元素。

times：重复的次数。

each：每次每个元素的重复次数。

length.out：向量的长度。

【实例 4-3】使用 rep() 函数建立向量对象重复元素的应用。

可重复向量元素为 5，共重复 5 次，如图 4-7 所示。

```
> rep(5,5)
[1] 5 5 5 5 5
```

图 4-7　建立向量对象重复元素示例一

可重复向量元素为 5，共重复 5 次，如图 4-8 所示。

```
> rep(5,times=5)
[1] 5 5 5 5 5
```

图 4-8　建立向量对象重复元素示例二

可重复向量元素为 1 ： 5，共重复 3 次，如图 4-9 所示。

```
> rep(1:5,3)
 [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

图 4-9　建立向量对象重复元素示例三

可重复向量元素为 1 ： 3，共重复 3 次，每次每个元素重复出现 2 次，如图 4-10 所示。

```
> rep(1:3,times=3,each=2)
 [1] 1 1 2 2 3 3 1 1 2 2 3 3 1 1 2 2 3 3
```

图 4-10　建立向量对象重复元素示例四

可重复向量元素为 1 ： 3，每个元素重复出现 2 次，向量对象总长度为 8 个向量元素，如图 4-11 所示。

```
> rep(1:3,each=2,length=8)
[1] 1 1 2 2 3 3 1 1
```

图 4-11　建立向量对象重复元素示例五

4.2　向量对象的数学运算函数

向量对象的数学运算函数有常见运算函数、计算元素积函数、累积运算函数、差值运算函数、排序函数、计算向量对象长度函数和基本统计函数等。

1. 常见运算函数

常见运算函数包括计算所有向量对象元素和的函数 sum()、计算所有向量对象元素最大值的函数 max()、计算所有向量对象元素最小值的函数 min() 和计算所有向量对象元素平均值的函数 mean()。分别说明如下。

【实例 4-4】计算某向量对象元素的总和、最大值、最小值和平均值示例。

代码及运行结果如图 4-12 所示。

```
> x<-c(2,6,4,8,12,10)
> sum(x)
[1] 42
> max(x)
[1] 12
> min(x)
[1] 2
> mean(x)
[1] 7
```

图 4-12　计算向量对象元素的总和、最大值、最小值和平均值

2. 计算所有向量对象元素积的函数

计算所有向量对象元素积的函数是 prod()，通常用于计算阶乘。

【实例 4-5】用 prod() 函数计算 5 的阶乘和 10 的阶乘示例。

5！=5×4×3×2×1，1、2、3、4、5 正好是向量对象（1 ： 5）的元素，所以，可以通过 prod（1 ： 5）来计算它们的乘积。同理，10！也可以通过 prod（1 ： 10）来计算。

代码及运行结果如图 4-13 所示。

```
> prod(1:5)
[1] 120
> prod(1:10)
[1] 3628800
```

图 4-13　计算向量对象元素的积

3. 累积运算函数

累积运算函数包括计算所有向量对象元素累积和的函数 cumsum()、计算所有向量对象元素累积积的函数 cumprod()、可返回各向量对象元素从向量起点到该元素位置间所有元素最大值的函数 cummax()、可返回各向量对象元素从向量起点到该元素位置间所有元素最小值的函数 cummin()。

【实例 4-6】计算某向量对象元素的累积和、累积积、累积最大值和累积最小值示例。

代码及运行结果如图 4-14 所示。

```
> x<-c(10,5,9,15,7,11)
> cumsum(x)
[1] 10 15 24 39 46 57
> cumprod(x)
[1]     10     50    450   6750  47250 519750
> cummax(x)
[1] 10 10 10 15 15 15
> cummin(x)
[1] 10  5  5  5  5  5
```

图 4-14　计算向量对象元素的累积和、累积积、累积最大值和累积最小值

cummax(x) 即 cummax(10，5，9，15，7，11)，向量元素起点为 10，与 5 比 10 大，与 10、5、9 比，仍是 10 大，与其 10、5、9、15 比 15 大，与 10、5、9、15、7 比，仍是 15 大，与 10、5、9、15、7、11 比，还是 15 大。所以输出结果是：10 10 10 15 15 15。

cummin(x) 即 cummin(10，5，9，15，7，11)，向量元素起点为 10，与 5 比 5 小，与 10、5、9 比，仍是 5 小，与 10、5、9、15 比 5 小，与 10、5、9、15、7 比，仍是 5 小，与 10、5、9、15、7、11 比，还是 5 小。所以输出结果是：10 5 5 5 5 5。

4. 差值运算函数

计算各元素与下一个元素的差的函数是 diff()，它返回的是各元素与下一个元素的差。

【实例 4-7】用差值运算函数计算上面向量对象元素的差值。

代码及运行结果如图 4-15 所示。

```
> diff(x)
[1] -5  4  6 -8  4
```

图 4-15　用差值运算函数计算向量对象元素的差值

diff(x) 即 diff(10，5，9，15，7，11)，5−10=−5，9−5=4，15−9=6，7−15=−8，11−7=4。由于返回的是每个元素与下一个元素的差值，所以结果向量对象元素必然会比原向量对象元素少一个。

5. 排序函数

用于向量对象元素排序的排序函数是：sort(x,decreasing=FALSE)，默认是从小排到大，即正序。所以，如果是从小排到大，则第 2 个参数 decreasing 可以省略。

用于返回向量对象位置的函数是 rank()，这个函数返回的向量对象元素是原向量对象按从小到大排序后所得向量对象的位置。

用于向量对象元素倒排的函数是 rev()，它可以将向量元素进行倒向排列。

【实例 4-8】分别用 sort()、rank()、rev() 函数对向量对象元素进行正排、逆排、正排中元素排位和颠倒排序示例。

代码及运行结果如图 4-16 所示。

```
> x<-c(10,5,9,15,7,11)
> sort(x)
[1]  5  7  9 10 11 15
> sort(x,decreasing = TRUE)
[1] 15 11 10  9  7  5
> rank(x)
[1] 4 1 3 6 2 5
> rev(x)
[1] 11  7 15  9  5 10
```

图 4-16　对向量对象元素进行排序

下面对 rank(x) 返回结果做一简单解释：原向量对象元素的第 1 个元素 10 排在正序中的第 4 位，第 2 个元素 5 排在正序中的第 1 位，第 3 个元素 9 排在正序中的第 3 位，第 4 个元素 15 排在正序中的第 6 位，第 5 个元素 7 排在正序中的第 2 位，第 6 个元素 11 排在正序中的第 5 位。

rev(x) 是颠倒排序函数，即将原数 (10，5，9，15，7，11) 按顺序倒排成 (11，7，15，9，5，10)，不是按数值大小倒排，所以与 sort(x,decreasing=TRUE) 有别。

6. 计算向量对象长度的函数

计算向量对象长度的函数是 length()，它可以计算向量对象的长度，即向量对象元素的个数。

【实例 4-9】用计算向量对象长度的函数计算向量对象元素的个数示例。

代码及运行结果如图 4-17 所示。

```
> x<-c(10,5,9,15,7,11)
> length(x)
[1] 6
```

图 4-17　计算向量对象元素的个数

7. 基本统计函数

基本统计函数有 sd() 和 var() 两个，sd() 函数用于计算样本的标准偏差，var() 函数用于计算样本的变异数（或称方差）。

【实例 4-10】基本统计函数的使用示例。

代码及运行结果如图 4-18 所示。

```
> sd(c(11,15,18))
[1] 3.511885
> var(14:16)
[1] 1
```

图 4-18　基本统计函数的使用

样本的标准偏差和样本的变异数是统计学中的两个基本统计量，标准偏差简称标准差，其值等于方差的平方根，方差是由统计学中规定的专门用于度量数据集分散程度的统计量。方差的计算公式是：全部数据与其平均值差的平方的和除以数据个数减 1 的值。方差越大，则说明数据离均值越远，数据越分散。由于方差的值一般很大，所以再将其开平方得到标准差。以上面数据为例，11、15、18 的和为 44，平均值为 14.6667。11 与 14.6667 的差为 −3.6667，15 与 14.6667 的差为 0.3333，18 与 14.6667 的差为 3.3333，它们的平方和为 24.666672，得到方差为 12.333336，开平方得到标准差为 3.511885。

向量对象（14 ： 16）的元素为（14，15，16），14、15、16 的和为 45，平均值为 15。14 与 15 的差为 −1，15 与 15 的差为 0，16 与 15 的差为 1，它们的平方和为 2，得到方差为 1。

【实例 4-11】对向量的综合运算示例。

代码及运行结果如图 4-19 所示。

```
> x<-c(1:5)
> y<-c(6:10)
> z<-c(x,y,11,12)
> z
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
> z<-(x+y)
> z
[1]  7  9 11 13 15
> min(z)
[1] 7
> range(z)
[1]  7 15
> sum(z)
[1] 55
> var(z)
[1] 10
```

图 4-19　向量的综合运算

第 1 条代码向量 x 中存储了序列 1，2，3，4，5，第二条代码向量 y 中存储了序列 6，7，8，9，10，第 3 条代码将向量 x、y 和数字 11、12 拼接成一个更长的向量 z，第 4 条代码显示了向量 z 存储的内容。第 5、6 条代码将向量 x 和 y 相加后赋给了向量 z，并查看了 z。由于向量 x 和 y 的长度相同，因此 z 中存储的内容即为向量 x 和 y 元素一一对应相加后的结果。在 R 语言中，可以直接对向量执行加、减、乘、除等基本运算，但首先要保证执行运算的两个向量的长度相等，否则将会得到难以解释或者错误的结果。在执行了第 5、6 条代码之后，向量 z 中存储的内容已经被替换为新的序列 7，9，11，13，15，这便是赋值过程中的“新冲旧”。

作为一个统计软件，R 语言中内置的函数十分丰富，写起来也十分方便。上述第 7 条代码用 min() 函数对 z 取最小值为 7，用 range() 函数给出了 z 的范围为 7 ～ 15，用 sum() 函数计算出了 z 的和为 55，用 var() 函数计算出了 z 的方差为 10。上面介绍过方差的计算公式是：全部数据与其平均值差的平方的和除以数据个数减 1 的值。5 个数的和为 55，5 个数的平均值为 11，全部数据与 11 的差分别为 −4、−2、0、2、4，其平方和为 40，除以（5−1）=4，所以得到方差为 10。

自我检测

一、判断题

（　　）1. 有如下两个命令：

```
>x<- -2.5:-3.9
>length(x)
```

上面命令的执行结果为如下所示。

```
[1]2
```

（　　）2. 有如下两个命令：

```
>x<-1:3
>y<-x+9:11
>y
```

上面命令的执行结果为如下所示。

```
[1]10 11 12
```

（　　）3. 有如下两个命令：

```
>x<-1:5
>x[-(2:5)]
```

上面命令的执行结果为如下所示。

```
[1]1
```