第一章 R语言介绍

1. R是一种区分大小写的解释型语言。你可以在命令提示符（>）后每次输入并执行一条命令，或者一次性执行写在脚本文件中的一组命令。R中有多种数据类型，包括向量、矩阵、数据框（与 数据集类似）以及列表（各种对象的集合）
2. R语句由函数和赋值构成。R使用<-，而不是传统的=作为赋值符号，使用=赋值并不常见

   x <- rnorm(5)与rnorm(5) -> x等价
3. 在命令行中运行demo( )可以了解R能够作出何种图形
4. 可以通过下列函数查看帮助文档

表1-2 R中的帮助函数

函 数	功 能
help.start()	打开帮助文档首页
help("foo")或?foo	查看函数 foo 的帮助（引号可以省略）
help.search("foo")或??foo	以 foo 为关键词搜索本地帮助文档
example("foo")	函数 foo 的使用示例（引号可以省略）
RSiteSearch("foo")	以 foo 为关键词搜索在线文档和邮件列表存档
apropos("foo", mode="function")	列出名称中含有 foo 的所有可用函数
data()	列出当前已加载包中所含的所有可用示例数据集
vignette()	列出当前已安装包中所有可用的 vignette 文档
vignette("foo")	为主题 foo 显示指定的 vignette 文档

   help.start()：打开一个浏览器窗口，我们可在其中查看入门和高级的帮助手册、常见问题集，以及参考材料

   RSiteSearch()：在在线帮助手册和R-Help邮件列表的讨论存档中搜索指定主题，并在浏览器中返回结果

   vignette()函数返回的vignette文档一般是 PDF格式的实用介绍性文章。不过，并非所有的包都提供了vignette文档
5. 工作空间（workspace）就是当前R的工作环境，它存储着所有用户定义的对象（向量、矩阵、 函数、数据框、列表）
6. 当前的工作目录（working directory）是R用来读取文件和保存结果的默认目录。我们可以使用函数getwd()来查看当前的工作目录，或使用函数setwd()设定当前的工作目录。如果需要读入一个不在当前工作目录下的文件，则需在调用语句中写明完整的路径。

表1-3　用于管理R工作空间的函数

函　数	功　能
getwd()	显示当前的工作目录
setwd("*mydirectory*")	修改当前的工作目录为 *mydirectory*
ls()	列出当前工作空间中的对象
rm(*objectlist*)	移除（删除）一个或多个对象
help(*options*)	显示可用选项的说明
options()	显示或设置当前选项
history(#)	显示最近使用过的#个命令（默认值为 25）
savehistory("*myfile*")	保存命令历史到文件 *myfile* 中（默认值为.Rhistory）
loadhistory("*myfile*")	载入一个命令历史文件（默认值为.Rhistory）
save.image("*myfile*")	保存工作空间到文件 *myfile* 中（默认值为.RData）
save(*objectlist*, file="*myfile*")	保存指定对象到一个文件中
load("*myfile*")	读取一个工作空间到当前会话中（默认值为.RData）
q()	退出 R。将会询问你是否保存工作空间

7. 输入与输出

   输入：函数source("*filename*")可在当前会话中执行一个脚本。如果文件名中不包含路径，R将假设此脚本在当前工作目录中。举例来说，source("myscript.R")将执行包含在文件myscript.R中的R语句集合

   输出：函数sink("*filename*")将输出重定向到文filename中。默认情况下，如果文件已经存在，则它的内容将被覆盖。使用参数append=TRUE可以将文本追加到文件后，而不是覆盖它。split=TRUE可将输出同时发送到屏幕和输出文件中。

   图形输出：使用下列函数，最后使用dev.off()将输出返回到终端。

表1-4　用于保存图形输出的函数

函　数	输　出
bmp("*filename*.bmp")	BMP 文件
jpeg("*filename*.jpg")	JPEG 文件
pdf("*filename*.pdf")	PDF 文件
png("*filename*.png")	PNG 文件
postscript("*filename*.ps")	PostScript 文件
svg("*filename*.svg")	SVG 文件
win.metafile("*filename*.wmf")	Windows 图元文件

8. 包是R函数、数据、预编译代码以一种定义完善的格式组成的集合。计算机上存储包的目录称为库（library）。函数.libPaths()能够显示库所在的位置， 函数library()则可以显示库中 有哪些包。命令search()可以告诉你哪些包已加载并可使用。
9. 第一次安装一个包，使用命令install.packages()，update.packages()可以更新已经安装的包，要查看已安装包的描述，可以使用installed. packages()命令，
10. 用library()命令载入这个包，在一个会话中，包只需载入一次

第二章 创建数据集

1. R中有许多用于存储数据的结构，包括标量、向量、数组、数据框和列表
2. R可以处理的数据类型（模式）包括数值型、字符型、逻辑型（TRUE/FALSE）、复数型（虚数）和原生型（字节）
3. 向量

   向量是用于存储数值型、字符型或逻辑型数据的一维数组。执行组合功能的函数c()可用来创建向量，单个向量中的数据必须拥有相同的类型或模式（数值型、字符型或逻辑型）

a <- c(1, 2, 5, 3, 6, -2, 4)

b <- c("one", "two", "three")

c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)

4. 矩阵

   矩阵是一个二维数组，只是每个元素都拥有相同的模式（数值型、字符型或逻辑型）。可通过函数matrix()创建矩阵。一般使用格式为：

   myymatrix* <- matrix(*vector*, nrow=*number_of_rows*, ncol=*number_of_columns*, byrow=*logical_value*,dimnames=list(char_vector_rownames*, *char_vector_colnames*))

   其中*vector*包含了矩阵的元素，nrow和ncol用以指定行和列的维数，dimnames包含了可选的、以字符型向量表示的行名和列名。选项byrow则表明矩阵应当按行填充（byrow=TRUE）还是按列填充（byrow=FALSE），默认情况下按列填充。

5. 数组

   数组（array）与矩阵类似，但是维度可以大于2。数组可通过array函数创建，形式如下：

   myarray <- array(*vector*, *dimensions*, *dimnames*)

   其中*vector*包含了数组中的数据，*dimensions*是一个数值型向量，给出了各个维度下标的最大值，而*dimnames*是可选的、各维度名称标签的列表。

6. 数据框

   数据框可通过函数data.frame()创建：

   mydata <- data.frame(*col1*, *col2*, *col3*,...)

   其中的列向量*col1*、*col2*、*col3*等可为任何类型（如字符型、数值型或逻辑型）。每一列的名称可由函数names指定。

7. 因子

   变量可归结为名义型、有序型或连续型变量

   名义型变量是没有顺序之分的类别变量

   有序型变量表示一种顺序关系，而非数量关系

   连续型变量可以呈现为某个范围内的任意值，并同时表示了顺序和数量

   类别（名义型）变量和有序类别（有序型）变量在R中称为因子（factor）

8. 列表

9. 数据输入

   用R内置的文本编辑器和直接在代码中嵌入数据

   使用read.table()从带分隔符的文本文件中导入数据，还有其他各种数据的导入方式