普通高等院校数据科学与大数据技术专业“十三五”规划教材

R语言程序设计基础

文必龙　高雅田 ■ 主编

R YUYAN CHENGXU SHEJI JICHU

華中科技大學出版社
http://www.hustp.com

普通高等院校数据科学与大数据技术专业“十三五”规划教材

R 语言程序设计基础

主　编　文必龙　高雅田
副主编　李　治　李　萌

华中科技大学出版社
中国·武汉

内 容 简 介

本书全面且系统地讲述了R语言的基础知识、图形工具及在机器学习算法中的应用。教材分为三个部分。第一部分是基础部分，由第1～5章组成，介绍R语言的基本语法、数据类型、控制流程、数据的输入输出及图形编程等。第二部分是高级部分，由第6～11章组成，介绍R语言的数据预处理、数据处理、数据统计性分析，以及回归分析、聚类分析、支持向量机、人工神经网络等常用数据分析算法的原理及应用编程。第三部分即第12章，是应用案例，以图书馆大数据分析为业务背景，逐步介绍了采用R语言进行数据探索、数据分析、数据可视化等的具体方法和实现过程。

本书按照“教、学、做”一体化的模式编写，本书可以作为高等院校R语言课程的教材，也可以作为软件开发人员学习R语言的参考书籍。

图书在版编目(CIP)数据

R语言程序设计基础/文必龙，高雅田主编. —武汉：华中科技大学出版社，2019.4
普通高等院校数据科学与大数据技术专业“十三五”规划教材
ISBN 978-7-5680-5006-7

Ⅰ. ①R… Ⅱ. ①文… ②高… Ⅲ. ①程序语言-程序设计-高等学校-教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2019)第062428号

R语言程序设计基础　　文必龙　高雅田　主编
R Yuyan Chengxu Sheji Jichu

策划编辑：李　露　廖佳妮
责任编辑：李　露
封面设计：原色设计
责任校对：李　琴
责任监印：徐　露
出版发行：华中科技大学出版社(中国·武汉)　　电话：(027)81321913
武汉市东湖新技术开发区华工科技园　　邮编：430223
录　　排：华中科技大学惠友文印中心
印　　刷：武汉华工鑫宏印务有限公司
开　　本：787mm×1092mm　1/16
印　　张：18
字　　数：441千字
版　　次：2019年4月第1版第1次印刷
定　　价：42.80元

前 言

PREFACE

近年来，国家明确提出实施国家大数据战略，大数据已然成为一个非常时尚、热门的概念。随着大数据变得越来越流行，对大数据的探索、分析、预测已经成为大数据分析领域的基本技能之一。“大数据”直到今天都是一个宽泛的概念，但是对其中“大”的概念已经达成了共识，那就是“数量庞大的数据集”。在“数量庞大的数据集”面前，传统的数据挖掘软件已经力不从心，在这种背景下由统计学家编写的一款免费的数据挖掘软件 R 语言进入了大数据研究者的视野中。R 语言提供可以进行交互式数据分析和数据探索的强大平台。经过近些年的发展，R 语言被越来越多的大数据研究者接受并使用。

目前关于 R 语言的参考书籍并不多，可以找到的 R 语言教材大多是由统计学家撰写的，在内容上偏向统计学理论的介绍，对 R 语言本身的讲解不多，并且统计学家的讲解方式对现今大量涌入大数据研究行业的计算机相关专业的学生来说友好度过低，造成教学者选材困难，使用者学习困难。由此编者产生了编写这本适合数据科学与大数据技术以及计算机相关专业学生使用的 R 语言教材的想法，并且完成了这本 R 语言教材。本书包括以下特色。

(1) 本书的作者全部为来自教学一线的计算机相关专业的高校教师，教学经历丰富，对计算机语言讲授方式把握精准，所编写的内容适合相关专业的学生学习。

(2) 本书从无到有一步一步地教读者使用 R 语言。

(3) 本书对统计学基础的要求较低，在循序渐进的学习过程中，读者可掌握相关的统计学知识。

(4) 语言表达简洁、严谨、流畅，内容通俗易懂、重点突出、实例丰富，可作为高校各专业程序设计语言课程的教材，也可以作为非计算机专业公共基础课教材。

(5) 本书对常用的 R 语言的语法与使用方法做了详尽的讲解。

(6) 书中包含大量的实例与代码，让读者在学习时事半功倍。

(7) 本书以一个完整的、易于理解的“图书馆大数据分析”为案例，运用大数据分析方法和书中介绍的 R 语言知识逐步实现需求分析、数据理解、数据处理、数据分析、数据可视化等全部过程。

本书第 1、3、6、7 章由李萌编写，第 2、4、5 章由李治编写，第 8、9、10、11 章由高雅田编写，第 12 章由文必龙编写。

本书提供全套教学课件和源代码，源代码均在 RStudio 中测试通过。

由于时间仓促，作者水平有限，书中难免出现疏漏，不足之处敬请读者批评指正。

目 录

CONTENTS

第1章 进入R的世界 /1

1.1 R及RStudio的下载与安装 /1
1.2 认识RStudio开发环境 /2
1.3 第一次使用R /4
1.4 R包 /6

第2章 R语言基础 /8

2.1 基本概念 /8
2.2 向量 /14
2.3 矩阵和数组 /28
2.4 数据框 /39
2.5 列表 /46

第3章 R函数与流程控制 /50

3.1 编写自己的R函数 /50
3.2 分支结构 /62
3.3 循环结构 /70

第4章 数据的输入输出 /75

4.1 基本的输入输出功能 /75
4.2 与其他软件的数据交换 /88

第5章 基本图形 /92

5.1 散点图 /92
5.2 曲线图 /94
5.3 图形的更多修饰 /100
5.4 常用图形 /113

第6章 数据预处理 /119

6.1 字符串的处理 /119
6.2 日期和时间的处理 /129
6.3 数据清洗 /133

第7章 数据处理与描述性统计 /140

7.1 apply()函数族 /140
7.2 数据处理 /151
7.3 描述性统计 /158

第8章 广义线性回归 /163

8.1 回归的基本模型 /163
8.2 检验多元回归模型 /175
8.3 评估与选择回归模型 /177
8.4 预测回归模型 /180
8.5 实践与练习 /181

第9章 聚类分析 /185

9.1 聚类分析的概念及常用函数 /185
9.2 K-means 算法 /188
9.3 高斯混合模型 /192
9.4 层次聚类法 /197
9.5 实践与练习 /201

第 10 章 支持向量机 /209

10.1 支持向量机概述 /209
10.2 线性可分及非线性支持向量机 /212
10.3 实践与练习 /217

第 11 章 人工神经网络 /233

11.1 人工神经网络概述 /233
11.2 神经网络中的感知机 /235
11.3 剖析神经网络 /239
11.4 神经网络实践与练习 /244

第 12 章 应用案例——图书馆大数据分析 /254

12.1 案例背景 /254
12.2 数据探索 /257
12.3 数据分析 /263
12.4 数据可视化 /271

参考文献 /277

第1章 进入R的世界

本章学习目标

- 掌握R及RStudio的下载与安装
- 认识RStudio开发环境
- 掌握R语言的基本开发过程
- 了解R语言的扩展包

R语言是针对统计分析、图形可视化、报告的完美工具，它在广泛的领域中都有着完美的表现。R语言的核心是解释计算机语言，其允许分支和循环，以及使用函数的模块化编程。R语言可以与C,C++,Java,Python等语言集成以提高效率。相信通过本章的学习，读者会对R环境有充分的认识，并且可以了解到与R的扩展包有关的知识。

R语言最初是由新西兰奥克兰大学统计系的Ross Ihaka和Robert Gentleman在S语言的基础上开发的。R语言于1993年首次亮相。一大群人通过发送代码和错误报告对R语言做出了贡献。1997年，R语言的核心开发团队成立，当然Ross Ihaka和Robert Gentleman是这个开发团队的成员，另外，S语言的开发者John Chambers也是这个开发团队的一员。这些成员拥有修改R源代码的权限。

1.1 R及RStudio的下载与安装

1.1.1 R的下载与安装

使用R语言的第一步是下载R程序。R程序是R语言核心团队免费提供给大家的，我们只需要访问R的网站https://www.r-project.org/，从“维护者”网站(CRAN)来下载需要的R程序。CRAN代理网站有很多，大家可以按照自己的实际情况来选择，本书选择国内的CRAN代理网站https://mirrors.ustc.edu.cn/CRAN/来下载R程序。

R语言需要依靠安装在操作系统中的R程序来运行。Windows系统、Mac OS系统、Linux系统都支持R程序的运行，当然前提是已下载了相应的R程序。本书中的所有例子

都在 Windows 环境下执行，所以本书选择 Windows 环境下的 R 程序。

首先选择下载 Windows 版本的 R 程序，然后选择 Base 即可在其中选择下载最新版本的 R 程序。成功下载 R 程序后需要安装 R 程序。找到下载的 R 程序，然后按照安装向导将 R 程序安装到指定目录，需要注意的是，安装的目录尽量不要存在空格或者中文。安装的过程中会询问用户安装哪些组件，其中包含 32 位和 64 位的两种程序，请用户按照自己的实际需求选择其中一个程序。安装完成后，双击 R 图标就可以打开 R 程序的标准界面，在这个界面中可以在 Console 区域中编写、运行 R 代码，同时代码运行结果也可以在 Console 区域中查看。

可以在 Console 区域中编写、运行 R 代码，同时查看代码运行结果的这一特征正是 R 的魅力所在。R 不用像 Java 等编程语言那样需要将源代码编译后运行才可以看到结果，R 的运行结果在输入相应的命令后会自动显示出来。

1.1.2 RStudio 的下载与安装

R 语言充满了魅力，有无数的程序员为 R 的发展无私地贡献了自己的力量。他们为 R 编写了大量的集或开发环境(IDE)，现在公认最好的是 JJ Allaire 小组设计的 RStudio 。本书中的例子使用 RStudio 来完成，下面来完成 RStudio 的下载与安装。

可以在 RStudio 的官网 https://www.rstudio.com/中来选择需要的 RStudio 版本来下载。由于本书是在 Windows 系统环境下使用 R，所以选择支持 Windows 的 RStudio 版本来下载。

成功下载需要的 RStudio 程序后，下一步需要找到下载的 RStudio 程序，然后按照安装向导将 RStudio 安装到指定目录中。需要注意的是，RStudio 安装成功的前提条件是已经成功安装了 R 程序，也就是说 RStudio 需要 R 程序来支持。

1.2 认识 RStudio 开发环境

安装并成功启动 RStudio 后，第一次打开 RStudio 可以看到 3 个区域。选择 File 标签中 New File 中的 R Script 创建一个新的 R 脚本，此时呈现出一个包含 4 个区域的 RStudio 窗体，这是 RStudio 程序常用的界面。如图 1-1 所示。

在这里简单介绍一下这四个区域。

1. Source Editor 区域

Source Editor 区域位于 RStudio 窗体的左上角，这个部分是 R 脚本的编辑区，在这里可以编写 R 语言程序代码，也可以保存并运行编写好的 R 程序代码。

2. Console 区域

Console 区域位于 RStudio 窗体的左下角。这个区域是 R 语言的主界面，可以在此直接输入指令并获得执行结果。这部分功能与 R 语言的类似，运用上下键可以切换上次运行的函数，特别的是，在 RStudio 窗体中按 Ctrl 键＋向上键可以显示最近运行的函数的历史列表。如果想重复运行前面刚运行过的程序，可利用该操作方式很方便地进行。

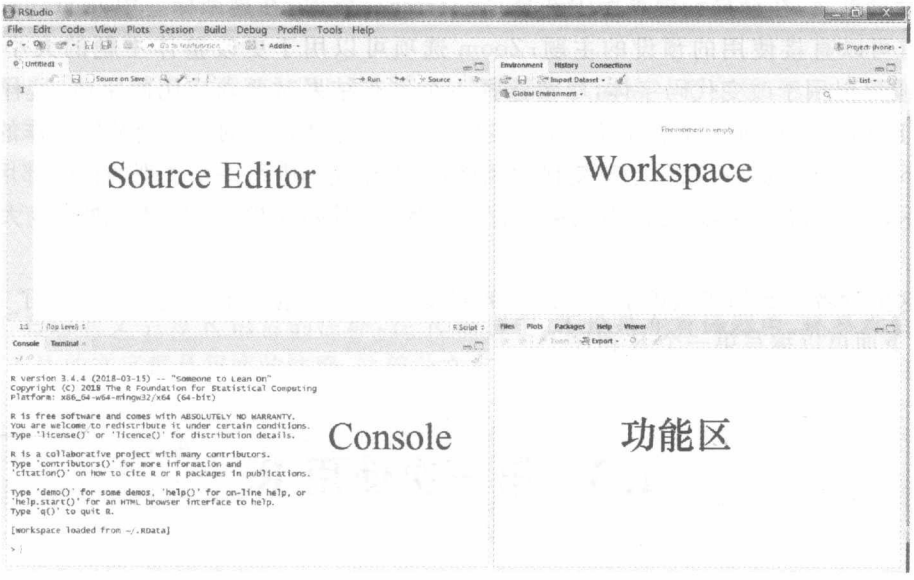

图 1-1　RStudio 窗口各区域图

3. Workspace 区域

Workspace 区域位于 RStudio 窗体的右上角。该部分的核心标签为 Environment 标签和 History 标签。其中，Environment 标签可以用于查看当前 RStudio 环境中所存在的变量名称和变量值；History 标签可以用于查看在 Console 区域中所有执行过的指令。

4. 功能区

功能区位于 RStudio 窗体的右下角。该部分包含 Files 标签、Plots 标签、Packages 标签、Help 标签等，部分标签的功能如下。

(1) Files 标签：这个区域可以对工作区的文件进行操作，可以显示工作区内的所有文件，单击“New Folder”按钮可以新建文件，单击“Delete”按钮可以删除一个文件，单击“Rename”按钮可以对文件重命名。当然，做这些操作之前要先勾选被操作的文件前面的复选框。More 选项则提供了其他功能。

(2) Plots 标签：这个区域可以显示 Console 区域要求输出的图。

(3) Packages 标签：在这个区域可以看到当前 RStudio 的所有扩展包，单击包名，可以在 Help 区域中查阅选中扩展包的说明文档，同时也可以在这个区域下载并安装新的扩展包。

(4) Help 标签：在这个区域可以看到你希望看到的说明文档。

R 自带的脚本编写工具是一款优秀的脚本编写工具，但是与 RStudio 相比，大多数程序员会选择使用 RStudio。因为 RStudio 存在丰富的可以满足各种需求的插件，并且 RStudio 支持用户自定义自己的程序编写风格，可以让程序员使用自己熟悉的编程风格来完成自己的程序。RStudio 初始主题为官方的主题，这个主题对于函数、注释等一些字体的颜色并没有显著的标注，并且字体大小、样式都不让笔者满意。下面介绍如何按照自己的习惯来修改 RStudio 的主题。

首先在菜单中选择 Tools 命令，在 Tools 中选择 Global Options 选项，进入到 RStudio

的全局设置界面，在全局设置界面中选择 Appearance 标签，在这个标签中，RStudio Theme 选项中有可以直接使用的预设的主题；Zoom 选项可以用于更改整体主题的大小；Editor Font 选项可以用于改变代码字体，尽量选择一个发布结果时要求应用的字体，这样会减轻工作量；Editor Font Size 选项可以用于改变字体大小，选择合适的字体会减轻工作的疲劳程度；Editor Theme 选项可以用于改变代码风格，可以让代码中每一个类别的代码都用不同的颜色表示，一个程序员熟悉的代码风格会让编写速度、检错速度，以及代码准确度大幅度提高。

到这里读者已经初步认识了 RStudio 的开发环境，并且按照自己的习惯设置了 RStudio 的风格，下面可以编写第一个 R 程序了。

1.3 第一次使用 R

下面编写经典的入门程序“Hello Word!”。由于本书主要程序都是在 RStudio 中完成的，所以第一个入门程序“Hello Word!”也用 RStudio 来编写。与 R 不同，RStudio 包含一个特色功能——项目功能，也就是说 RStudio 可以把每一次的工作放在不同的项目里，方便用户来管理自己的 R 程序。

开始一个新项目比较简单，选择 File 菜单中的 New Project 命令来创建一个新的项目。新项目的建立有 3 个可选项：New Directory、Existing Directory、Version Control。它们的功能分别是在一个新的目录中建立一个新的项目、在一个现存目录下建立一个新的项目、从一个版本库中查看一个项目。在这里，入门程序采用第一种方式建立一个新项目，首先为项目起一个名字，单击“Create Project”按钮，新的 RStudio 项目就建立好了，可以看到在 Files 窗体中会生成一个新的.Rproj 文件，这个文件用于存放结果目录并追踪整个项目，如图 1-2 所示。

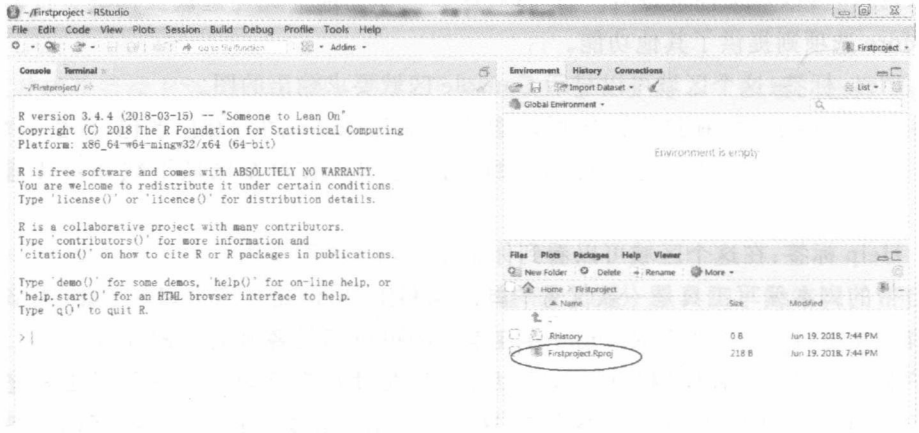

图 1-2 在 RStudio 中创建好的新项目

项目创建成功后，下面正式开始编写“Hello Word!”程序，这个程序的目的是打印“Hello Word!”这句话。R 是一种支持直译器的语言，所以编写“Hello Word!”这句话非常

容易，只需要在 Console 区域中直接输入命令“print("Hello Word!")”就可以得到想要的结果。

在 Console 区域中也可以把“Hello Word!”字符串赋值给一个变量，然后将变量打印出来，程序如下。

```
>#  Hello Word 程序
>fistString <-"Hello Word!"
>print(fistString)
[1] "Hello Word!"
```

“#”右边的文字为 R 语言的注释，在 R 语言中，不论是在直译器中，还是在脚本程序中，“#”右边的文字都是程序的注释，这部分文字只是增加了 R 语言的可读性，不会参与程序的运行。当然还有另一种更像程序员的实现方式，那就是在 Source Editor 编辑器中编写程序，然后通过执行操作来实现“Hello Word!”。

创建一个新的 R 脚本，在 File 菜单中选择创建一个 R Script，然后把刚才的程序写在新建的 Source Editor 编辑器中，选中编写的程序，单击“Run”按钮来运行程序（如果想运行编辑器中的所有代码，直接单击“Source”按钮即可，程序运行结果会显示在 Console 窗口中。

下面介绍在 Source Editor 编辑器中编写实现“Hello Word!”的程序。在 Source Editor 编辑器中编写程序是可以保存下来的，单击保存按钮为保存的程序取一个名字。此例命名为 hello，保存后在 Files 窗口会生成一个新的 hello. R 文件（R 语言默认的文件扩展名为 .R），如图 1-3 所示，这个文件就是已保存好的“Hello Word!”程序。

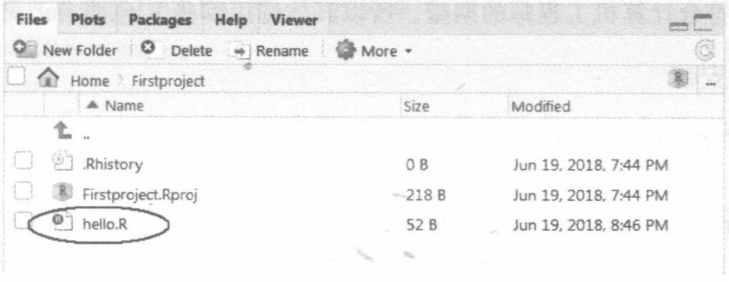

图 1-3　保存好的 R 程序

到这里，入门程序“Hello Word!”就编写完成了，需要注意的是，这个时候 Source Editor 编辑器中的文件名称已经变成了 hello. R，如果想编辑新的程序并且不想改变现有的“Hello Word!”程序，只需要关闭 hello. R 窗口，然后单击 Console 窗口右上角的按钮，就可以生成一个新的未命名的程序编辑窗口，如图 1-4 所示。

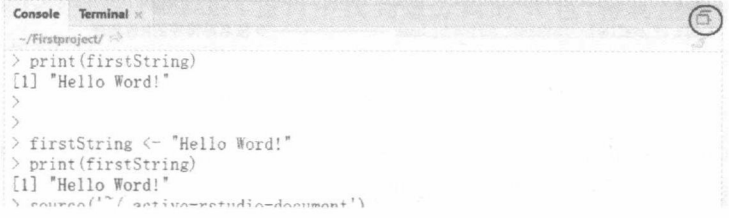

图 1-4　生成一个新的未命名的程序编辑窗口

1.4 R包

R语言现在越来越流行，主要原因是大量的程序员无私地奉献出了自己编写的程序。R语言收集这些程序员自己编写的程序的方式就是依靠包来实现的。截至2018年6月，在R的官方网站中公开发布的可用R包达到了12630个。这些R包几乎囊括了统计学的方方面面，有效地利用这些包可以减少程序员自己的编写量。并且这些包很多都是由某一领域的专家学者编写的，他们有丰富的专业知识，他们编写的程序算法更合理、更高效。

那么什么是R包呢？R包本质上来说其实就是事先编辑好的并实现了一系列功能的代码库，我们可以把它看作R的功能扩展，它包含了大量的函数。不是所有函数都存在于基础的R中的，例如，项目需要利用R绘图所用的Lattice绘图系统和ggplot2绘图系统中的函数来绘制辅助分析图，现阶段基础的R中就不包含这两个绘图系统所使用的函数。那么如果项目需要使用这些函数，就需要加载相应的包，Lattice绘图系统需要加载lattice包，ggplot2绘图系统需要加载ggplot2包。

需要注意的是，并不是所有发布在官网（这里指CRAN代理网站，后同）上的包都是稳定的、高质量的包，官网中的包都是程序员自己发到网上的，有些包可能会存在一些问题。更重要的是，大部分的R包是由统计学家编写的解决统计学问题的R包，这些R包虽然正确，但并不一定符合计算机工程师的期望。所以具体选用哪些R包要自己测试。

1.4.1 R包的安装

R语言的扩展包除了在其官网上可以找到以外，还可以在Bioconductor和GitHub等平台上找到。

本书主要介绍如何在R的官网上找到并安装R包。首先需要打开R的官网，选择链接Packages程序，Packages页面中的第一部分包含现在网站中可用R包的个数，以及两个链接：Table of available packages, sorted by date of publication链接和Table of available packages, sorted by name链接。这两个链接分别为按照时间排序的可用包列表和按照名称排序的可用包列表。具体如图1-5所示。

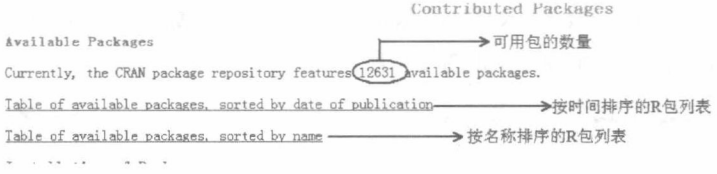

图1-5 CRAN中的R包列表

选择Table of available packages, sorted by name链接后就会显示按照名称排序的平台上现有的所有R包，选择L字母就可以找到Lattice绘图系统需要加载的lattice包，单击后，可以看到lattice包中的相关信息。

找到相关的R包后，需要下载和安装用户需要的R包。在网站下载扩展包，在软件中

自行安装的常规编程语言的扩展包的安装方式也适用于R语言。但是R程序员通常不会采用这种方式来下载并安装R包，R程序员通常采用以下两种方式来下载和安装R包。

第一种为使用RStudio自带的图形安装界面来下载并安装R包。首先选择Packages窗口，单击窗口左上角的“Install”按钮。在名称部分填入想下载并安装的包的名字，例如Lattice绘图系统需要加载lattice包，当然如果你想加载多个包，只需要把包名称都写到输入栏中并用“,”分隔开即可，如果需要同时下载与该包相关联的其他必要包，则可点选Install Dependencies复选框，选择好后，直接单击“Install”按钮，R程序会自动下载并安装用户需要的包。第二种方式为通过在控制台输入命令来安装R包。如果想在CRAN中下载并安装某一个R包，只需要在Console窗口中输入“install.packages(" ")”命令即可，在英文引号中填写想安装的R包的名称。以下载并安装Lattice绘图系统需要加载的lattice包为例，需要在Console窗口中输入以下代码：

```
>install.packages("lattice")
```

这就自动实现了下载并安装R包的功能。

1.4.2　R包的加载

在R语言中，R的扩展包安装好之后，其中的函数是不可以直接调用的。如果想使用下载的R的扩展包，首先需要把这个扩展包加载到R中，有两种命令可供选择，分别为library命令和require命令。它们都可以完成加载包的需求，区别在于require命令在加载成功的时候会返回一个“TRUE”信息，在加载失败的时候会返回一个“FALSE”信息。因此如果想要加载lattice包只需要执行以下任意一条代码即可：

```
>library(lattice)
>require(lattice)
```

需要注意的是，一个包仅在启动一个新的项目时需要加载，一旦加载过之后就一直存于R程序的工作空间中，直到项目关闭或该包被分离为止。

另外一种包的加载方式是在Packages窗口中找到想加载的R包，点选其左侧的复选框，也可以加载想要的包。

1.4.3　R包的分离

有的时候可能需要卸载掉已经加载的包，这个操作在R中称为包的分离。包的分离同样有两种方法，一种是使用命令，可以使用detach命令来分离加载过的包，在这个命令中，需要在包名前面加“package:”作为参数。例如分离ggplot2这个包需要执行以下代码：

```
>detach("package:ggplot2")
```

另一种分离方式是在Packages窗口中找到想分离的R包，将其左侧的复选框取消选择，即可分离该R包。

本章小结及习题

第2章 R语言基础

本章学习目标

- 掌握R语言中变量和函数的特点
- 掌握向量的构建、元素提取及运算方法
- 掌握矩阵的构建、元素提取及运算方法，了解数组的特点
- 掌握数据框的构建、元素提取及运算方法
- 掌握列表的构建及元素提取方法

程序的任务是对数据进行处理，对象即程序处理的各类数据，而各种运算符和函数则用来指明对数据的处理方式。数据按运算方式的不同，可以分为数值、逻辑和字符等三类。根据结构的不同，在R语言中，数据可还分为向量、矩阵、数组、数据框和列表等不同类型。本章首先结合R语言的特点，介绍程序中涉及的一些基本概念。然后根据不同结构，对常用数据类型的特点及相关的运算符和函数进行介绍。

2.1 基本概念

在程序中，数据按其表现形式，可以分为常量和变量等两类，其中变量的正确使用在程序编写中至关重要。此外，在R语言中，函数几乎无处不在，各种各样的函数极大地降低了编程的难度。因此，本节将对变量和函数的基础知识进行介绍。

2.1.1 变量及其赋值

在一些简单情况下，数据可以直接以原始形式，即常量的形式表现，如：

```
>print(3.1415)
[1] 3.1415
```

上面这行代码使用了一个函数print()，其功能是根据括号内数据的特点选择合适的显示方式，并将其显示在屏幕上。在这里，该函数的作用是将括号内的数字直接显示在屏幕上。

这种直接使用常量的形式，至少存在以下几个缺点。第一，当需要在多个地方重复使用这个常量时，有可能会在多次输入的过程中出现录入错误，尤其是在常量比较长、重复使用又比较多的情况下容易出错。第二，如果程序中多处使用这个常量，则要对这个常量进行修改时，就需要对多处同时进行修改，但当对程序本身不够了解时，容易出错。第三，当某个常量需要随情况不同而变化时，无法使用这种形式。由于存在以上缺点，因此直接使用常量的情况比较少见。在绝大多数情况下，数据都需要采用变量的形式来呈现。

1. 变量名

一个变量可以视为一个装载数据的容器。在 R 语言中，变量的类型和大小随所装载数据的类型和大小而变化。为了能够在程序运行过程中随时调用这个容器，需要对这个容器进行命名，即对每一个变量指定一个变量名。

在 R 语言中，变量名可以包含字母、数字、下划线和句点，但不能以数字和下划线开头。另外，变量名中的大小写字母表示着不同的变量，如 a、A、a1、A1、a_1、a.1、.a 都是合法的不同的变量名，且 a 和 A 代表两个不同变量。但需要注意的是，以句点开头的变量比较特殊，默认不显示，如：

```
>x1=1; x2=3; .x=5          # 可以利用";"将多条命令写在一行
>ls()                      # 显示当前工作环境下的变量名称
[1] "x1" "x2"
>ls(all.names=T)           # 强制显示以句点开头的变量
[1] ".x" "x1" "x2"
>rm(list=ls())             # 删除当前工作环境中的所有变量
>ls()
character(0)
>ls(all.names=T)
[1] ".x"
```

如上所示，虽然命名了一个变量".x"，但使用 ls()函数显示当前运行环境中的变量时默认不显示，必须指定显示以句点开头的变量才会显示。同样，在删除当前运行环境中的变量时，该变量也默认不被删除。

虽然，理论上只要符合命名规则的变量名都可以使用，但在实际使用时仍然要注意不要随意命名变量。例如，理论上在程序中可以使用 x1、x2、x3 等对所有变量进行命名，但当程序较长、所用变量较多时，很容易在变量调用时出错。甚至，如果采用 f6nVIz、fOUd9V、0o0o0o 这样的形式来命名变量，那么出错几乎就是不可避免的了。因此，程序中所用的变量名应该尽量能够反映其内容。例如，使用 height、weight 来命名变量，则不容易在调用时发生混淆。但当程序中所用变量较多，而一个单词不足以表示其内容时，直接拼接多个单词可能并不容易识别，此时可以采用首字母大写、加下划线或句点进行分隔的方式。例如，需命名一个变量表示心率(heart rate)时，可以采用 HeartRate、heart_rate、heart.rate 等形式表示。另外，虽然有时可以采用首字母缩写的方式来表示多个单词的变量名，如使用 HR 表示 heart rate，但当这种缩写形式不太常用时，应该进行注释，否则很容易隔一段时间之后就完全忘了该变量所代表的含义。最后，变量名不能和 R 语言中的保留字相冲突。可以使用"? reserved"命令查询 R 语言所用的保留字。

2. 变量赋值

在指定变量名之后，就可以往这个容器里装载数据了，这个过程称为赋值。R语言中的赋值符号为“<－”，但绝大多数情况下也可以使用“＝”。例如，以下两句均可实现对x的赋值：

```
>x <-1
>x=1
```

但二者在使用时需要注意一些细节。

1）关系比较时的陷阱

大多数常用数值运算符并不强求在其前后加空格，因此很多时候虽然数值表达式省略了空格，但不影响计算结果，如：

```
>1+1              # 计算 1+1
[1] 2
>x=1              # 给 x 赋值为 1
>x+2              # 计算 x+2
[1] 3
>1 <3             # 判断是否 1 小于 3
[1] TRUE
>1 <-2            # 判断是否 1 小于-2
Error in 1 <-2 : invalid (do_set) left-hand side to assignment
```

对于以上五句代码，前四句的运行结果都符合预期，但第五句运行出错。原因在于“<”后面紧接一个负数时，R语言会将“<”和“－”优先连接在一起，导致第五句被解释为“给1赋值为2”，但由于1是一个常量，不可以被赋值，因此出错。正确写法如下：

```
>1 <  -2
[1] FALSE
```

需要注意的是，本例中由于1是常量，所以会有出错提示。但在要判断变量x是否小于－2时，若没有在“<”后添加空格，则会直接给x进行赋值，而不会有出错提示，如：

```
>x=1
>x <  -2          # 判断 x 是否小于-2
[1] FALSE
>x <-2            # 若没有添加空格，则直接给 x 进行赋值
>x
[1] 2
```

2）函数中参数传递时的误用

这个错误涉及的原因比较复杂，将在下一小节做详细解释。

2.1.2 函数和参数

在R语言中，函数的使用形式如下：

函数名(参数1，参数2，…)

每一个函数都会根据其参数返回相应的运算结果。为方便调用，每一个函数都有一个函数名，其命名规则和变量名的相同，二者的差别在于后面是否连接圆括号，如：