第 1 章

R 语言介绍

R 语言是当前主流的数据分析和统计软件之一，它提供了一个丰富的生态系统，包含了用于数据分析、可视化和统计建模的各种工具和包。更重要的是，它是免费和开源的。对于许多希望发表 SCI 论文的研究人员而言，这款软件可谓是“神兵利器”。

1.1 R 语言概述

本节首先介绍什么是 R 语言，然后介绍临床医生使用 R 语言进行大数据分析的优势。

1.1.1 什么是 R 语言

R 语言常用于统计计算、数据挖掘和机器学习等领域。特别是在医学大数据分析和挖掘方面，R 语言已成为一个非常重要的工具。

之所以要开发 R 语言，是因为如果为了发表一篇科研论文或者教学而去购买付费商业软件，显然是不划算的。于是 1991 年 Robert Gentleman 和 Ross Ihaka 开发了这款免费开源的语言，由于两位开发者的名字都以 R 开头，因此将其命名为 R 语言。

其实，Robert Gentleman 是一位生物学家，并非统计学家或计算机学家。他开发 R 语言的初衷是为了生物统计，因此 R 语言最初就是为了生物统计而设计的。后来，他还开发了专门用于生物信息学分析的工具包 Bioconductor。通过使用 Bioconductor，可以快速地对生物数据和高通量数据进行分析与可视化。

在生物信息学领域，R 语言可以进行大量的分析，包括基本的序列分析、分子进化和比较基因组学；蛋白质结构比对和预测；计算机辅助药物设计；等等。生物信息学已成为 R 语言的一个重要应用领域，近年来 R 语言的迅猛发展在很大程度上也得益于生物信息学的推动。

1.1.2 临床医生使用 R 语言的优势

R 语言在医学科研领域有着丰富的应用场景和巨大的潜力。越来越多的医生希望通过 R 语言来

帮助自己完成科研项目，因为熟练掌握 R 语言的好处非常多！

那么，对于临床医生来说，使用 R 语言有什么优势呢？

1. R 语言完全免费且开源

R 语言完全免费且开源，这意味着任何人都可以从 R 语言社区中获取代码和文档，而且可以轻松地共享自己编写的代码。这有助于降低学习成本，提高编程效率，并更好地服务于科学研究。我们可以在它的网站及其镜像站点下载相关的安装程序、源代码、程序包及其文档资料，并且不断有“大神”级开发者上传新的代码和源文件，使 R 语言社区变得越来越强大。

2. 使用 R 语言可以实现数据的批量预处理、清洗和整理

如果只有几十、几百个数据，我们还可以手动地一个一个地录入、清洗和整理。但是，当数据达到成千上万的量级时，手动整理数据的时间成本将快速升高。使用 R 语言，几个包和几行代码就能帮我们实现结构化数据的整理，节省的时间可以用于更有意义的工作。R 语言可以轻松读取、清洗和处理各种数据类型，包括结构化和非结构化数据。在医学研究领域，往往需要挖掘大量的公共数据库来寻求新的治疗方法或疾病诊断技术，而 R 语言就可以处理各种数据库。

3. 几乎所有医学相关的数据分析、建模和制图都可以通过 R 语言来完成

对于要学习临床科研统计分析的人而言，掌握 R 语言通常就足够了。虽然我们常用的 SPSS 软件用起来也非常方便，但很多近年来流行的分析方法在 SPSS 中难以实现。而 R 语言的更新速度远远超过其他统计软件，最新的统计分析方法和最前沿的 R 包都能在第一时间获取，使得研究更具前沿性。

R 语言的统计分析能力可以帮助我们轻松地完成统计分析、数据挖掘和机器学习等任务。R 语言还拥有丰富的绘图和数据可视化功能，使用户可以将复杂数据转换为更易于理解和可视化的图形。

4. 很多临床数据都有现成的 R 包，可以直接调用

有很多临床医生想发文章，但苦于缺乏数据，也没有那么多的时间和精力去收集数据。实际上，很多大型临床数据已经有现成的 R 包可以调用。

例如，针对之前的热点“新型冠状病毒感染”，通过一个名为 COVID19 的 R 包就可以快速获取全球不同地区新冠病毒感染的历史确诊、住院、重症、死亡、接种人数等数据。我们完全可以利用这些数据来进行研究并发表文章。除了 GitHub 和 Bioconductor 上的资源，目前仅在 CRAN（Comprehensive R Archive Network）上就有 18948 个 R 包，涉及统计学、生物信息学、数据挖掘和数据可视化、机器学习等各个领域。

总之，R 语言在数据分析和统计学习领域是一个非常重要且流行的工具，尤其在医学大数据分析和挖掘方面具有广泛的应用价值。

1.2 R 编程环境的搭建

R 语言对编程环境的要求不高，可以在多种操作系统平台上运行，包括 Windows、macOS 和

Linux。要运行 R 语言，需要安装 R 解释器。可以从 R 语言的官方网站下载和安装最新版本。R 语言还需要一个集成开发环境来编写和运行代码，如 RStudio。

R 和 RStudio 的区别，可以一句话概括为：R 是 R 语言自带的解释器，而 RStudio 是 R 的一个集成开发环境。因此，在安装 RStudio 之前必须安装 R。

相比普通的 R 软件，RStudio 让 R 编程更加方便快捷，更加方便编写、修改和调试代码。此外，RStudio 提高了代码的复用性，更便于查看已有变量的值及数据结构类型，也更便于使用程序包。由于 RStudio 功能强大且易于使用，因此使用 R 语言时一般都会安装 RStudio，这使得 R 编程的学习和实践更加轻松和方便。

1.2.1　R 语言的下载和安装

R 语言支持 Windows、macOS、Linux 操作系统，因此在进入 R 的官方网站（https://cran.r-project.org/，见图 1-1）后，我们需要根据自己计算机上的操作系统选择对应的下载链接。下面以 Windows 系统为例，讲解 R 语言的下载和安装过程。首先，单击“Download R for Windows”链接。

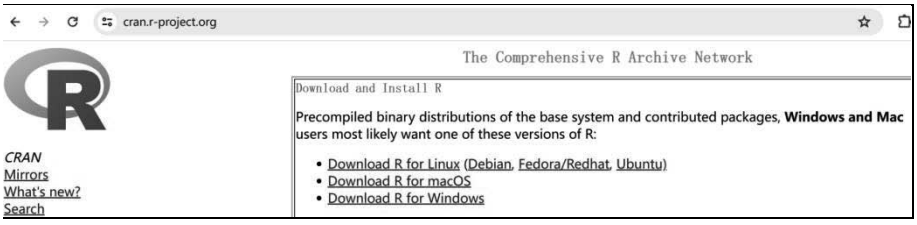

图 1-1

进入下载页面后，再单击“base”链接，如图 1-2 所示。

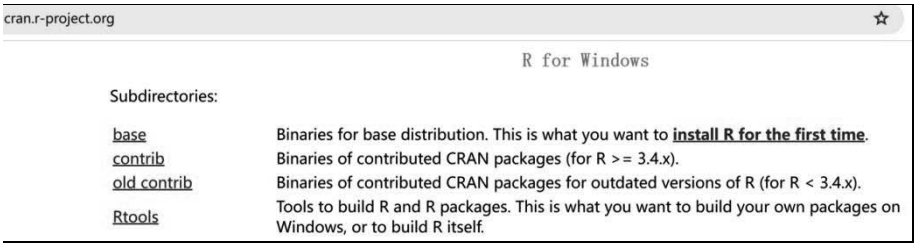

图 1-2

最后，单击“Download R-4.3.2 for windows”链接即可开始下载，如图 1-3 所示。

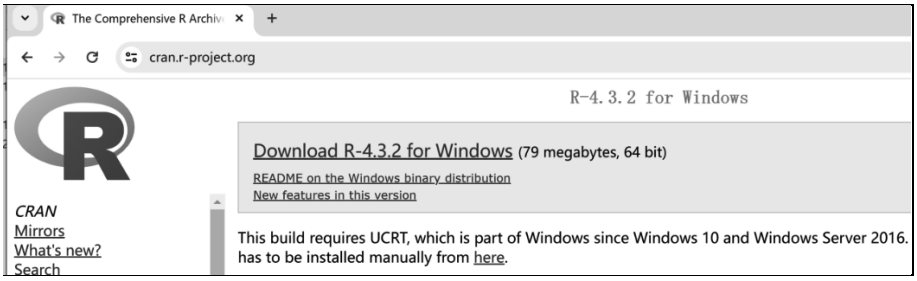

图 1-3

下载完毕后，打开此安装包，出现安装向。安装过程与一般软件类似，直接单击“下一步”按钮即可，如图 1-4 所示。

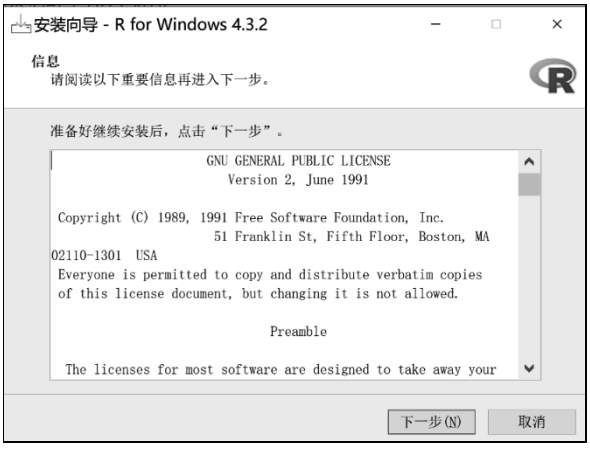

图 1-4

关于软件的安装目录，一般选择默认安装路径即可。选择组件时，也可以选择默认设置。最后，等待 R 安装完成。

安装完成后，双击打开 R 的原生界面，在交互式的命令窗口输入代码进行测试，例如输入 print("Hello, world")，然后按回车键，结果如图 1-5 所示。得到了结果"Hello, world"，说明安装无误。

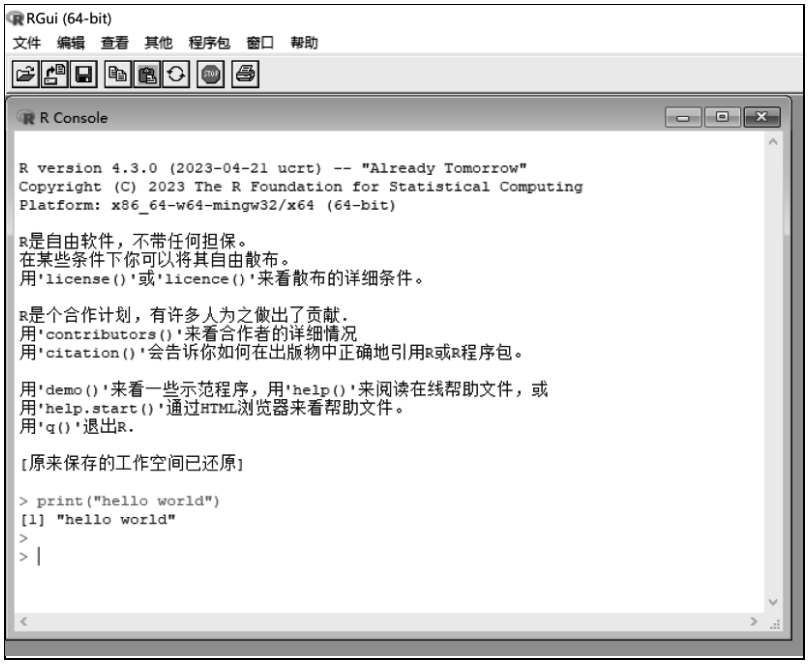

图 1-5

1.2.2　RStudio 的下载和安装

R 语言是一门解释型语言，虽然 R 语言的原生编辑器也可以编写 R 脚本，但通常我们使用功能更强大、界面更美观的 RStudio，它是最受欢迎的 R 语言集成开发环境（Integrated Development Environment，简介 IDE）。需要注意的是，R 语言是 RStudio 的核心组成，安装 RStudio 之前必须安装 R 语言。RStudio 是 R 语言的“盔甲”，为 R 语言提供了一个更强大、更易使用的界面。

RStudio 的官方网站（https://posit.co/download/rstudio-desktop/）如图 1-6 所示，单击“DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS”按钮，即可下载 RStudio 软件安装包。

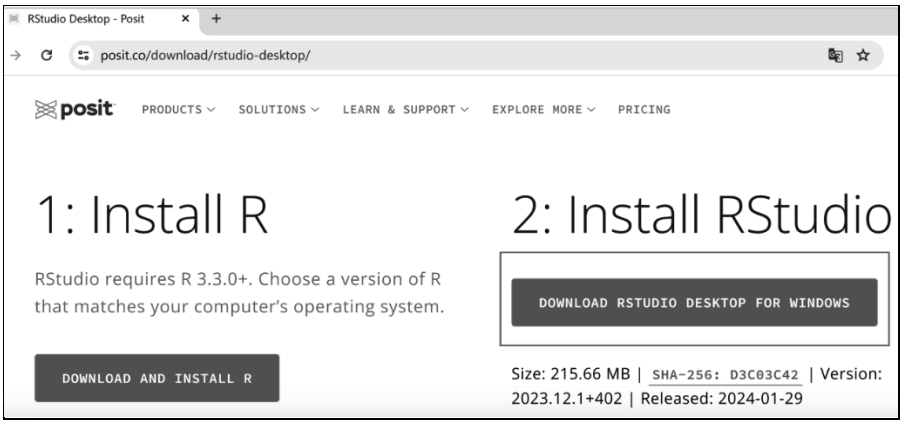

图 1-6

双击下载的 RStudio 软件安装包以启动 RStudio 安装程序，如图 1-7 所示。按照默认设置，逐步单击“下一步”按钮即可。

图 1-7

推荐读者直接使用功能更强大、体验更好的 RStudio 来学习 R 语言和编写脚本。

1.2.3 RStudio 操作

当我们完成安装并第一次打开 RStudio 时，依次单击界面左上角菜单栏中的“File”→“New File”→“R Script”菜单选项，即可看见如图 1-8 所示的界面。

图 1-8

这一步操作将新建一个名为“Untitled1”的 R 代码文件（后缀名默认为.R）。现在可以在代码编写区域内编写代码。编写完成后，按快捷键 Ctrl + S 即可保存文件，也可以依次单击菜单栏中的“File”→“Save”菜单选项进行保存。接着会跳出“Save File”对话框，在对话框中可将“Untitled1”文件重命名，然后单击“Save”按钮保存文件。

在代码编写区域输入的代码，可通过单击“Run”按钮来运行光标所在行的代码，每单击一次按钮便运行一行，也可通过按快捷键 Ctrl+Enter 运行。界面左下方是 Console 区，这个区域用来执行代码，执行结果也会显示在这里。右上方的区域中包含 4 个模块，其中“Environment”模块用于记录当前变量的数值，我们可以通过它清楚地查看每个变量当前的赋值。右下方区域包括“Plots”，用于显示绘图结果。例如，在代码编写区输入如下代码：

```
#准备一个向量
cvd19 = c(83534,2640626,585493)

#显示条形图
barplot(cvd19)
```

界面如图 1-9 所示。

图 1-9

由于网速较慢，有时可能会安装失败。此时可以通过将包的安装切换至中国镜像网站来解决：依次单击菜单栏中的“Tools”→“Global Options...”菜单选项，接着依次单击“Packages”→“Change...”选项选中一个中国镜像，如图 1-10 所示。以后安装包的时候将通过这个镜像网站进行安装。

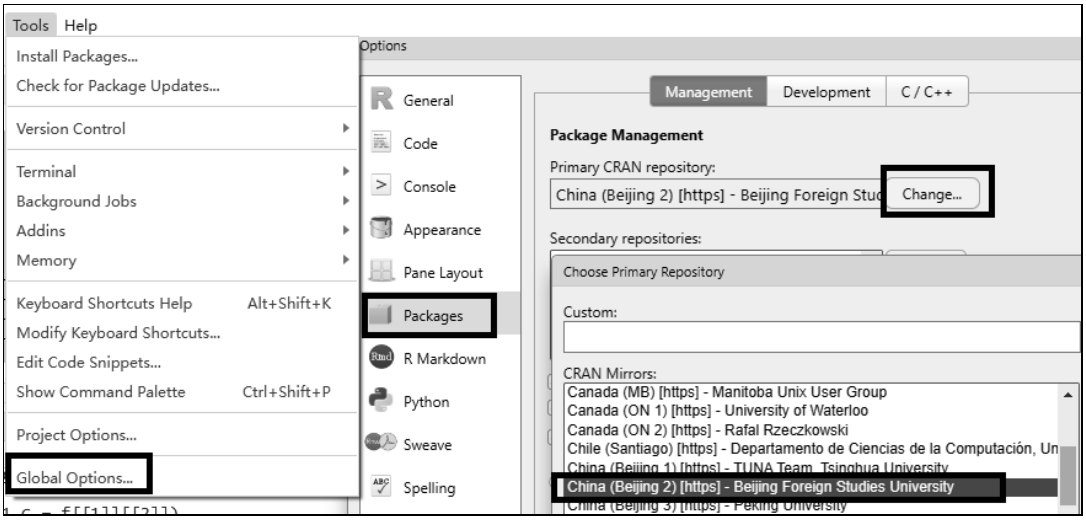

图 1-10

1.3 R 语言包

本节主要介绍什么是 R 语言包（简称 R 包），以及如何安装这些包。

1.3.1 什么是 R 包

R 语言的一个显著特点是它拥有众多的第三方扩展包，这些扩展包涵盖各行各业的数据分析内容。R 包是 R 函数、实例数据和预编译代码的集合，包括 R 程序、注释文档、实例、测试数据等。如果把 R 语言比作沃土，那么 R 包就是其上的鲜花。开源共享的开发者社区提供了很多功能丰富的 R 包，方便用户充分利用 R 语言完成各项工作。R 包的原理是将函数和数据等打包成一个库，用户在安装和加载 R 包后，可以直接调用其中的函数和数据，从而加快编程和分析的速度，提升编程效率和数据处理能力。

一般来说，一个包负责解决某个具体问题。例如，graphics 包由一些基本绘图函数构成，为 R 语言提供基本绘图功能。只有在包被载入时，它的内容才能被访问。一些常用的且基本的程序包（如 base、stats 等）已经被收录到标准安装文件中，R 语言安装好之后即可使用。这些包提供了很多默认函数和数据集，用户可以直接使用。

但是，当我们需要进行其他操作，使用别的包时，就必须下载并安装这些包。存储库（repository）是包所在的位置，因此可以从存储库中安装 R 包。R 包中最受欢迎的 3 个存储库是：

- CRAN：官方存储库，由全球 R 社区维护的 FTP 和 Web 服务器网络。它由 R 基金会协调，包在发布前需通过若干测试，以确保遵循 CRAN 策略。
- Bioconductor：一个专题库，专注于生物信息学的开源软件。作为 R 语言的综合档案网络，Bioconductor 有自己的提交和审核流程，其社区非常活跃，每年举行多次会议。
- GitHub：虽然这不是 R 语言特有的，但 GitHub 可能是开源项目中最受欢迎的存储库。它因开源的无限空间、与 Git 的集成、版本控制软件以及与其他人共享和协作的便利性而备受欢迎。

1.3.2 R 包的安装

根据 R 包的安装源不同，有以下 3 种方法安装 R 包。

1. CRAN 网站（https://cran.r-project.org/）

CRAN 网站提供多种镜像支持，可以选择离自己最近的镜像网站来减少网络负载。从 CRAN 安装 R 包，主要使用 install.packages 函数在线安装。例如，要安装 ggplot2 包，可使用如下命令：

```
install.packages("ggplot2")
```

2. Bioconductor（https://bioconductor.org）

Bioconductor 是一个专注于生物学的 R 包平台，包含各种基因组数据分析和注释的工具。从 Bioconductor 安装 R 包，主要通过 BiocManager 包来完成。例如，要安装 DESeq2 包，可使用如下命令：

```
install.packages("BiocManager")
library(BiocManager)
BiocManager::install("DESeq2")
```

3. GitHub（https://github.com）

这是一个开源的社区平台，很多开发者会把自己开发的 R 包发布在 GitHub 上，而不是挂载到 CRAN 上。此外，有些人还会把 GitHub 当作服务器，将自己网页的源码托管在上面，并解析到个人域名上。从 GitHub 安装 R 包，主要通过 devtools 包来完成。例如，要安装 survminer 包，可使用如下命令：

```
install.packages("devtools")
library(devtools)
devtools::install_github("kassambara/survminer")
```

这 3 种方法都可用来在线安装 R 包。一般来说，一个 R 包（如 ggplot2、Seurat 等）可能会依赖于数十个其他 R 包，因此需要确保网络连接良好。R 包安装完成后，可以使用 library()函数载入相应的 R 包，例如“library(ggplot2)”。注意，每次重新打开 RStudio 时都需要载入 R 包。安装 R 包时，圆括号中的 R 包的名称必须加引号（单双引号皆可），而载入 R 包时，包名可不加引号。

1.4　初识 R 语言的注意事项

R 语言是一种区分字母大小写的解释型语言，可以在命令行中输入一条命令，也可以一次性执行写在脚本文件中的一组命令。R 语言支持多种数据类型，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）。R 语言中的多数功能是由程序内置函数和用户自定义函数提供的，一次交互式会话期间的所有数据对象会被保存在内存中。一些基本函数默认是直接可用的，而其他高级函数则包含在按需加载的程序包中。

R 语言通常用“<-”进行赋值，它将右侧表达式的值赋给左侧的变量。例如，“x<-3”表示将 3 赋给变量 x。在 RStudio 中，使用快捷键 Alt+ -时会自动在其前后添加空格。在 R 语言中，单行注释用“#”。

初学者易犯的错误如下：

（1）使用了错误的字母大小写，R 语言是区分字母大小写的。例如，help()、Help()和 HELP()表示 3 个不同的函数，但只有第一个是正确的。

（2）忘记使用必要的引号。例如，install.packages("gclus")能够正常执行，而 install.packages(gclus)将会报错。

（3）在函数调用时忘记使用圆括号。例如，要使用 help()而非 help。即使函数不需要参数，仍需加上“()”。

（4）在 Windows 上，路径名中使用了“\”。R 语言将反斜杠视为转义字符。例如，代码 setwd("c:\nhanes")是错误的，而 setwd("c:/nhanes")则是正确的。这里的 setwd()函数用于设置 R 语言的工作目录。

（5）使用了尚未载入包中的函数。例如，函数 order.clusters()包含在包 gclus 中，如果还没有载

入这个包（通过 library()函数载入）就使用它，将会报错。

在 R 语言中，工作环境（working environment）是指当前正在执行的 R 代码的上下文。所有对象（如数据、变量、函数等）都存在于工作环境中，并可以在代码中进行访问和操作。在一个 R 会话结束时，我们可以将当前工作空间保存到一个镜像中，并在下次启动 R 语言时自动载入，保存的文件格式为.RData。例如，使用 save.image("myfile")可以将工作空间保存到文件 myfile 中，load("myfile")可以将保存的工作空间载入当前会话中。

R 语言的工作目录是 R 语言用于读取数据和文件以及保存结果的文件夹，即我们录入 R 语言的数据和编写的 R 代码需要保存在文件夹中，以便下次直接读取和调用。我们可以使用函数 getwd()来查看默认的工作目录（文件夹）。

R 语言默认的工作目录路径很深，会增加操作复杂度，因此，我们一般会设置一个便于自己工作的工作目录。

R 语言是统计编程的首选语言，集统计分析与图形显示于一体。在医学科研领域具有丰富的使用场景和巨大的潜力。本书将手把手教读者学会使用 R 语言进行数据挖掘，撰写并发表 SCI 文章！