第一章　关于 R 的基本知识

1.1　R 是什么

我国外语界大部分的统计教程都是基于 SPSS 的，本书则介绍另一种选择：R。R 是一种统计计算与制图系统，SPSS 能做的它都能做；同时它还有方便而强大的制图功能和用作基于命令行的高级计算器。但它能做的远不止于此，更像是一把统计与计算领域的高级瑞士军刀。

R 是 20 世纪 90 年代初问世的开源（open-source）软件，对非商业使用完全免费，因此在学术界和教育界备受青睐。R 是一个开放平台，用户可以自由修改、补充已有功能或者添加新功能，而且可以像为智能手机开发 APP 一样为 R 制作各种包（package），所有用户都可以自由下载和使用这些包。这种社区式的架构吸引了心理学、医学、生命科学、计算语言学、自然语言处理、人工智能等众多领域的专业人士为 R 开发了数不胜数的包，而且时常发布更新，R 的功能因此不断得到拓展。自诞生以来，经无数用户和计算机高手的贡献，R 在很多领域迅速普及，成为国际上最流行的统计计算与制图工具之一。

1.2　R 与 SPSS 的主要区别

R 和 SPSS 在统计计算功能方面不相上下，主要区别在于其他方面：

表 1.1　R 与 SPSS 的主要区别

	R	SPSS
费用	对个人完全免费	昂贵
操作界面	命令行	图形界面
语言	英文	多语种

（待续）

（续表）

	R	SPSS
开放性	开源	封闭
自编程	允许，是主要手段	允许，是辅助手段
扩展性	无限扩展	无法扩展

R 与 SPSS 最直观的区别在于界面。SPSS 采用窗口式图形界面和鼠标操作，各种统计方法被分门别类地组织到菜单中，层次非常清晰，界面设计得很精致。但据编者了解，很多研究生用 SPSS 学了统计之后，记忆中往往只留下窗口、菜单、选项、按钮等，却常常忘记操作背后的原理。教师讲解起来也相当费力，由于操作总是离不开窗口，所有 SPSS 教材都包含大量截图，一旦离开书本就只能面对面、手把手地教；如果师生所用的 SPSS 版本不同，个别操作有所改变，就更麻烦。各种 SPSS 教程的作者都对截图和在图上做各类标注之烦琐深有体会。虽然图形、窗口和鼠标操作确实能降低学习门槛，但对深入学习来说反而构成干扰。而 R 则是通过命令行来工作，采用函数加参数的形式，统计运算的诸多细节（相当于 SPSS 中的各种“选项”）是以参数的形式写到命令中的，用户的全部注意力都放在统计任务本身，而不需要考虑打开哪个窗口、在哪里打钩；这意味着教师讲解会变得更简单，只要列出几行命令就可以了，不需要截图，更不用手把手、面对面地讲，显然可以极大地简化教学、提高效率。更重要的是，命令行操作方式能促使用户更多地注意统计原理而不是操作方法。根据编者的教学实践，学生一开始就从 R 入手学统计，后续的学习并不困难。

当然，用惯了 Windows 图形界面的人一开始会对 R 的使用觉得不适应。为了改善这种体验，多数 R 用户会使用一些类似图形界面的辅助性环境。这类环境有好几种，本书推荐用 RStudio，它将用 R 工作时所涉及的各类内容同时显示在主界面上，使人一目了然。这种环境在所有 R 环境中用得最为普遍。虽然 RStudio 只是 R 的一个外壳，但它能极大地提高使用 R 的工作效率。本书就结合 RStudio 介绍 R 的用法。R 和 RStudio 都是英文的界面，用户需要掌握统计的基本英文术语，并记忆一些函数的名称和用法。在这一点上 R 对用户的要

求高于 SPSS，不过这会促使用户更深入地理解统计，其实是有好处的。R 中有非常详尽的帮助文档，用户随时可以查阅任何命令的使用方法；网上也有大量资料可供参考，包括各种专门网站、教程、讨论组等。

与商业软件相比，R 有无与伦比的扩展性，这是因为 R 有大量实用的包可用，好比无数种工具的集合；多数包能将复杂的工作简单化。很多领域都能找到多种可用的包。仅以应用语言学为例，就有 corpus 和 koRpus 用于文本处理、词性标注、可读性分析、词汇多样性计算等；NLP 包提供了一些自然语言处理的基本工具；lsa 包用于潜在语义分析（latent semantic analysis）；tm 包用于文本挖掘（text mining），提供语料处理、元信息（metadata）标注、文本预处理、tf 计算、idf 计算等；stylo 包可使用 Burrows' Delta 及类似算法做文本聚类，而且有图形界面；translateR 提供了调用 Google 和 Microsoft 的自动翻译 API 的方法（需要用户拥有相应 API 的账号）；seewave 用于语音分析与合成，可以轻松制作 2D、3D 甚至动画音频频谱图等；Rcrawler 是一种网络爬虫工具，用于从网站上自动采集语料。另外还有一些专用的脚本工具，如搭配构式分析方法创始人之一 Stefan Th. Gries 教授编写的 coll.analysis.r，专用于搭配构式分析计算。显然，如此丰富的扩展功能在 SPSS 中是不具备的。

R 的巨大优势之一是数据可视化（制图）能力非常强大。很多学术性文章或著作都需要方便的制图工具，手绘的效果显然不够理想；而 R 往往只用一条命令就可以基于数据制出高质量的图，非常专业。而且其制图命令都有大量选项（参数），用于微调制图的细节。本书第五章对此会有集中描述。

在 R 中可以将多个命令按某种方式组合成为程序脚本（script），快速实现复杂的任务处理，而且脚本常可以在相似的任务中重复使用，所以能节省大量的时间精力。由于统计计算与数据是分开的，只要编写好一个脚本，点一下鼠标或者写一句命令，在适当位置加载数据表，即可一次性完成整个统计任务，而不需要每次都按固定顺序打开某些窗口，点击选取某些选项；如果数据表本身有所变化，使用脚本的优点就更为明显，只要对脚本做少量修改即可。当然，这需要研究者花时间学一些编程方法，但这是非常值得的。由于 R 编程超出了本教程的范围，本书不做介绍。其实 SPSS 也允许用户编写命令，但使用命令完成任务的用户似乎不多，可能是因为多数用户养成了对图形界

面的依赖。

很多人学 SPSS 时花大量时间去“寻找”合适的版本。由于安装包非常庞大，无论下载还是安装都非常费力。而 R 和 RStudio 则对个人用户完全免费，安装包加在一起远远小于 SPSS，并且少了很多麻烦。

对普通 Windows 用户来说，R 的主要“缺点”包括：

- 操作基于命令行，一开始不习惯；
- 计算结果的输出形式是文字而不是 SPSS 那样的表格，而且全都是英文；
- 包大多是用户编写的，质量有高有低，需要甄别。

目前推广 R 的主要障碍似乎在于命令行界面，但这并非真正的困难，反而是一种优势，因为其减轻了关于窗口、选项、顺序等的记忆负担；计算结果的形式差异也只是表面的；包的质量差异同样不构成问题，因为实现同样目的的命令和方法非常多样，可以选择最优方案，建议采用多数人使用的包即可。不习惯命令行的用户可能担心自己会忘记操作方法，但其实与任何统计软件一样，R 也需要经常使用，随着实践的增多，这些“困难”都不在话下。当然 R 确实有一个缺点，即处理海量数据的效率不如商业软件，但这对外语界的科研内容来说可以忽略。

总体上看，SPSS 是把统计方法做成了现成的工具，而 R 更像是一大堆零部件组成的统计软件。前者像是做工精细的传统玩具，后者更像是形状各异的乐高积木。编者认为后者更有助于激发想象力和创造力。

1.3　R 的学习资源

R 的相关学习资料极其丰富，国内外出版的关于 R 的文献已经有很多，读者可参考本书的相关文献推荐部分。

还有很多专门的 R 教程网站，以及邮件列表（mailing list）等学习资源：

- 网上书籍 *Summary and Analysis of Extension Program Evaluation in R*，网址是 http://rcompanion.org；
- 专题网站 R Tutorial，网址是 http://www.r-tutor.com；
- 邮件列表 R-help，是 R 项目组提供的，非常活跃，每天都有很多人提出

疑问，许多专家做出解答。可向 r-help@r-project.org 发邮件申请加入，具体见 https://stat.ethz.ch/mailman/listinfo/r-help 中的说明。

R 本身就包含大量帮助文档，任何时候要想查看某个命令的用法，可利用 Console（关于 Console 和各种命令的用法，见后面的章节）在想查阅的命令前加问号，如“?hist”，回车即可在 Help 一栏看到帮助信息。

1.4 R 的引注

R 和各种包都凝结了贡献者的智慧和时间精力。如果在研究中用了 R，应将相关知识资源列入参考文献，也就是注明出处，这是 R 项目组对用户的明确要求。另外一些包的作者也要求使用者注明出处。要想知道怎样引注 R，可在 Console 中输入 `citation()` 并回车。

1.5 本书的形式约定

本书与所有 R 教程一样，包含大量代码（code）。计算机代码与普通文字不同，必须遵循一些形式约定。考虑到外语界多数读者对计算机代码比较陌生，这里对 R 代码的一般约定做集中说明：

- 代码中的符号都必须是半角，除非是引号内的字符；
- 除引号内的文字之外，空格可有可无，数量也可多可少，其设置与安排主要考虑代码内容布局的美观性和易读性；
- 括号（包括方括号、圆括号、花括号等）和引号必须成对出现；
- 引号只能是直引号（' 或 "），多数情况下单引号和双引号可以互换；
- 引号可以嵌套，但相邻层次的引号必须有区别。如果引号内的文字中也包含本层次的引号，则要在其前面加反斜线（\"）；
- 凡“#”号后面直到行尾的文字只是用于提示信息（注释），不会被作为命令执行；
- 同一行中可以有多个命令，但要用分号隔开。

本书为了区分代码和普通文字，用虚线方框表示 R 中输入的命令[1]及其输出结果，其中：

- 行首的“>”号是命令提示符，输入命令后回车即可执行；如果一行命令还没有结束，后续各行的提示符会变为加号（+），直到命令完全结束；
- 行首没有提示符的文字均是 R 的命令的执行结果或 R 的提示信息。

RStudio 只是 R 的外壳，所以多数情况下操作方式其实与用哪种界面无关；个别涉及 RStudio 的具体操作会伴有文字说明。

1　为便于读者辨认和学习，虚线方框内和正文中出现的命令均采用等宽字体和半角符号的形式。