江苏“十四五”普通高等教育本科规划教材
“十三五”江苏省高等学校重点教材

R语言在生物统计中的应用

（第二版）

主编　杨泽峰

南京大学出版社

江苏“十四五”普通高等教育本科规划教材
“十三五”江苏省高等学校重点教材
（编号：2020-2-236）

R语言在生物统计中的应用

（第二版）

主　编　杨泽峰
副主编　徐　扬　鲁　月　陈茹佳
编　委　陈洪瑜　陈茹佳　崔彦茹　方慧敏
黄晓敏　李鹏程　李钱峰　鲁　月
乔　琴　陶天云　徐　扬　徐辰武
杨泽峰　张恩盈　周　勇　左示敏

南京大学出版社

图书在版编目(CIP)数据

R 语言在生物统计中的应用 / 杨泽峰主编. -- 2 版.
南京 : 南京大学出版社, 2025. 10. -- ISBN 978 - 7 - 305
- 29270 - 5

Ⅰ. Q - 332

中国国家版本馆 CIP 数据核字第 2025RV8491 号

出版发行　南京大学出版社
社　　址　南京市汉口路 22 号　　邮　　编　210093

书　　名　R 语言在生物统计中的应用
R YUYAN ZAI SHENGWU TONGJI ZHONG DE YINGYONG
主　　编　杨泽峰
责任编辑　吴　华　　编辑热线　025 - 83596997

照　　排　南京开卷文化传媒有限公司
印　　刷　南京鸿图印务有限公司
开　　本　787 mm×1092 mm　1/16 开　印张 17.5　字数 426 千
版　　次　2025 年 10 月第 2 版　2025 年 10 月第 1 次印刷
ISBN 978 - 7 - 305 - 29270 - 5
定　　价　49.80 元

网　　址:http://www.njupco.com
官方微博:http://weibo.com/njupco
微信服务号:NJUYUNSHU
销售咨询热线:(025)83594756

前　言

自《R 语言在生物统计中的应用》第一版付梓问世，承蒙广大读者的厚爱与支持，这本书在助力生物统计学爱好者与科研工作者学习和运用 R 语言的征程中，发挥了积极的作用。在与读者的互动交流中，我们收到了诸多宝贵的反馈和建议，这些犹如熠熠星光，为第二版的修订之路照亮方向。与此同时，生物统计学与 R 语言生态系统正以前所未有的速度蓬勃发展、革新演进，这一形势也激励着我们精心打磨并推出第二版，力求满足读者在全新学术环境下的学习与工作需求。

在第二版的写作中，我们秉持与时俱进的理念，积极融入前沿知识与技术。一方面，我们延续第一版的写作风格，系统地阐述了 R 语言在农学和生物学数据分析中的常用方法。从基础的描述性统计分析，到图表的精心制作；从常用概率分布分析，到假设测验、方差分析；从相关和回归分析，再到聚类分析、主成分分析以及非参数测验等，我们都进行了详细讲解，并对每一项输出结果从生物学意义的角度进行深入解释和推断，帮助读者更好地理解数据背后的生物学内涵。另一方面，我们对原有的内容进行了全面且细致的梳理和优化。不仅更新了部分案例和数据，确保内容紧跟时代步伐，还着重强化了数据可视化的相关内容，让数据以更加直观、形象的方式呈现，使其更契合当前生物统计学的研究热点与实际应用场景。尤为重要的是，我们着重引入了支持向量机、决策树、随机森林和朴素贝叶斯等机器学习内容。通过一系列典型案例分析，读者能够深入洞悉机器学习算法在生物统计中的应用原理与实践技巧，熟练掌握运用 R 语言实现这些算法的方法，进而显著提升在生物大数据分析领域的专业能力。

本书的适用范围广泛，无论是生物统计学的初学者，渴望搭建 R 语言与生物统计知识的基础框架，还是具有一定经验的科研人员，寻求新的研究思路与分析方法，本书都具有极高的参考价值。

本书涉及的所有 R 语言代码（含详尽注释）和数据可以在扬州大学农学院网站下载，下载网址为：https://nxy.yzu.edu.cn/rcpy/bkjy/cbjc.htm。本书与智慧树课程运行服务平台合作，建有课程学习板块，涵盖教学视频、教案、演示文稿、课后作业、章节测试等内

容，读者可以在智慧树网站注册并参加该课程学习。

尽管在修订过程中，我们全力以赴、精益求精，但受限于学识与精力，书中或许仍存在一些不足之处。我们衷心期盼读者能够继续不吝赐教，提出宝贵的意见和建议，助力我们不断完善本书。让我们携手并肩，深入探索 R 语言在生物统计领域的无限可能，为推动生物统计学的发展贡献力量。

再次向一直关注和支持本书的广大读者致以最诚挚的谢意！

编　者

2025 年 8 月

目　录

第 1 章　R 语言基础 …… 1

1.1　R 语言的功能和特点 …… 1
1.2　R 的获取与安装 …… 3
1.3　R 的菜单操作 …… 4
1.4　工作空间 …… 5
1.5　程序包 …… 6
1.6　RStudio 的安装和使用 …… 9

第 2 章　基本数据管理 …… 15

2.1　创建数据集 …… 15
2.2　数据的读取和存储 …… 25
2.3　R 的运算符 …… 31
2.4　R 常用函数及其应用 …… 33
2.5　编辑数据集 …… 39
2.6　R 语言编程简介 …… 47

第 3 章　描述性统计 …… 54

3.1　利用函数求解描述性统计数 …… 54
3.2　描述性统计数的相关软件包 …… 57
3.3　分组描述性统计 …… 62
3.4　频数分布分析 …… 65

第 4 章　假设测验 …… 73

4.1　F 测验 …… 73
4.2　t 测验 …… 75
4.3　卡方测验 …… 81

第 5 章　方差分析 …… 86

5.1　方差分析常用函数和包简介 …… 86
5.2　单因素试验资料的方差分析 …… 89
5.3　两向分组资料的方差分析 …… 93
5.4　二因素完全随机化试验资料的方差分析 …… 95
5.5　二因素随机区组试验资料的方差分析 …… 98
5.6　系统分组试验资料的方差分析 …… 100
5.7　品种区域试验资料的方差分析 …… 102
5.8　裂区试验资料的方差分析 …… 106
5.9　协方差分析 …… 109

第 6 章　相关与回归分析 …… 116

6.1　线性相关分析 …… 116
6.2　一元线性回归分析 …… 127
6.3　多元线性回归分析 …… 131
6.4　非线性回归 …… 135
6.5　二分类变量 Logistic 回归 …… 138

第 7 章　聚类分析和判别分析 …… 146

7.1　聚类分析 …… 146
7.2　判别分析 …… 161

第 8 章　主成分分析和因子分析 …… 171

8.1　主成分分析 …… 171
8.2　因子分析 …… 177

第 9 章　非参数测验 …… 191

9.1　二项分布检验 …… 191
9.2　Kolmogorov-Smirnov 检验 …… 194
9.3　两个独立样本的测验 …… 196
9.4　多个独立样本的测验 …… 198
9.5　两个配对样本的测验 …… 199
9.6　多个配对样本的测验 …… 200

第 10 章　机器学习 …… 204

10.1　机器学习算法简介 …… 204
10.2　支持向量机 …… 206

10.3 决策树 …… 210
10.4 随机森林 …… 217
10.5 朴素贝叶斯 …… 223

第 11 章 绘图与数据可视化 …… 227

11.1 基础图形系统 …… 227
11.2 ggplot2 图形系统 …… 237
11.3 ggplot2 拓展包图形系统 …… 261

主要参考文献 …… 272

第1章

R语言基础

R语言是一款功能极为强大的编程语言，集统计计算与图形展示能力于一身，在数据分析、数据挖掘以及统计绘图等领域得到了广泛应用。作为开源软件，它不仅配备了丰富多样的统计分析工具包，还具备直观便捷的数据处理和图形生成能力，这使其在科学研究和数据分析实践中广受赞誉。

R语言最初是在1997年由新西兰奥克兰大学的Ross Ihaka、Robert Gentleman和其他志愿人员共同开发的。其设计初衷旨在打造一个与S语言相似但功能更为强大的统计分析环境。R语言是在S语言的基础上发展而来的，而S语言本身则受到了20世纪60年代贝尔实验室所开发的统计软件的影响。自问世以来，R语言迅速发展成一个功能强大的统计计算平台，尤其在数据分析和图形展示方面表现卓越。如今，R语言由一个核心团队负责维护，全球众多用户也积极贡献力量，不断为其增添新的功能和软件包。

1.1 R语言的功能和特点

1.1.1 R语言的功能

R语言的主要功能可大致归纳为以下几个方面：数据存储与处理、数组运算、统计分析、统计制图以及编程功能。

(1) 数据存储与处理：R语言提供了诸如向量、矩阵和数据框等多种数据结构，这让数据的存储与处理变得极为灵活。这些数据结构能够高效应对大量数据集，足以胜任各类复杂的数据分析任务。在数据处理环节，R语言具备强大的数据导入与导出功能，能够处理来自不同源头的数据，例如文本文件、数据库，甚至是从网页抓取的数据。R语言中还有众多用于数据清洗的函数，可协助用户处理缺失值、异常值以及进行数据转换等问题。

(2) 数组运算工具：R语言在数组运算方面表现十分卓越，尤其在向量和矩阵运算上优势显著。这些功能为开展复杂的数学计算提供了极大便利，特别是在统计学和线性代数的应用场景中。R语言支持广泛的数学运算操作，涵盖代数运算、逻辑运算以及特殊的数学函数等，这使其在解决统计计算问题时展现出强大的能力。

(3) 统计分析功能：R语言实现了诸多经典及现代统计方法，例如描述性统计分析、假设检验、回归分析等，这些工具助力研究人员在不同研究领域开展精准的数据分析。此

外，R 语言还具备广义线性回归、非线性回归、混合模型等高级统计方法的实现，进一步拓展了其在统计分析领域的应用范畴。这些高级功能尤其适用于生物统计学、社会科学和市场研究等领域。

(4) 统计制图功能：R 语言拥有一流的可视化软件包，能够生成从简单散点图到复杂多维度图表等多种高质量的统计图表。这些图形功能极大地提升了数据的视觉呈现能力，有助于研究者更好地阐释和展示他们的数据分析结果。R 语言支持动态和交互式图形的创建，这些功能在数据探索和呈现过程中尤为关键，能够实时展示数据分析结果，提升用户体验。

(5) 编程功能：R 语言的语法简洁明了，易于学习与使用。它支持分支和循环结构，允许用户编写自定义函数，进而实现高度定制化的数据处理和分析流程。R 语言还支持面向对象编程，这为复杂项目和大型分析任务提供了更优的代码组织和管理方式。

此外，R 语言拥有一个活跃的社区以及丰富的资源，这对于初学者和专业人士而言都是巨大的财富。借助这些资源，用户能够有效提升自身的数据分析技能，解决各类统计问题。

1.1.2 R 语言的特点

R 语言作为一门专注于统计计算和数据分析的编程语言，具备一系列特性，使其在数据科学领域脱颖而出。这些特性不仅体现在功能性上，还体现在开放性、灵活性以及广泛的应用场景方面。

(1) 自由且开源的特性：R 语言属于自由软件，用户不仅能够免费使用，还可以访问其源代码。这对于希望深入学习或进行功能自定义的用户而言极为有益。作为开源软件，R 语言鼓励全球范围内的用户和开发者参与到其测试、开发与改进工作中。这种开放的协作模式极大地加快了新功能的实现以及 bug 的修复速度，同时也保障了软件能够持续更新和优化。

(2) 跨平台兼容性：R 语言支持多种操作系统平台，涵盖 Windows、Mac OS 和 Linux。这意味着用户能够依据个人偏好选择最适合自己的操作系统开展数据分析工作，无需担忧兼容性问题。跨平台特性使 R 语言在全球范围内得以广泛应用，无论是学术研究还是商业分析领域，它都占据着重要地位。

(3) 面向对象编程：R 语言支持面向对象的编程方式，这提升了代码的模块性与可重用性。用户能够定义自己的类和方法，以满足特定的分析需求。面向对象编程特性让复杂项目的管理变得更为简便，便于维护和扩展。

(4) 丰富的资源库：R 语言拥有极为庞大的资源库，即综合典藏网(CRAN)。这里存储着大量第三方包，涵盖从统计分析到机器学习，再到可视化等多个领域的功能。综合典藏网提供的各类包极大地拓展了 R 语言的功能，使其能够应用于各种复杂的数据分析场景。

(5) 社区支持与文档：R 语言拥有一个十分活跃的社区，用户可以在社区论坛上提问并分享经验，这为初学者和专业人士解决了大量难题。R 语言的文档极为详尽，再加上社区的支持，让处于从入门到精通各个阶段的用户都能获得有效的帮助和指导。

R 语言的这些特性彰显了其作为统计编程语言的独特优势。在数据分析和统计建模领域，R 语言营造了高效且灵活的环境，让数据科学家能够更加专注于数据本身，而无需过多关注工具的使用。这种聚焦于数据的语言设计，加之强大的社区支持和资源库，确保了 R 语言在数据科学领域持续保持热度且无可替代。

1.2　R 的获取与安装

若要下载 R 语言，可访问其官方网站（https://www.r-project.org/）。在该网站首页，您能够找到最新版本 R 语言的下载链接。点击“Download R”按钮，便可进入下载页面。

除官方网站外，用户也可以借助 CRAN（Comprehensive R Archive Network）镜像站点进行下载。CRAN 作为 R 语言的软件仓库，在全球范围内提供 R 语言软件包及更新服务。目前，全球 CRAN 镜像站数量超过一百个。在中国，为提升下载速度，您可以选用清华大学、中科大等高校提供的 CRAN 镜像站点。这些网站提供了适用于 Linux、Mac OS X 和 Windows 系统的安装版本，用户只需在该网站选择相应版本的 R 语言进行下载，随后即可完成安装。

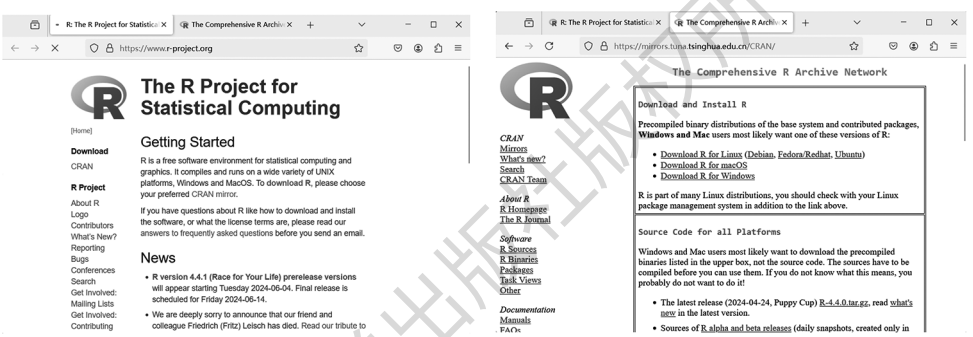

图 1－1　R 语言的官方网站和 CRAN 下载界面

R 的安装在 Windows 环境下与一般的软件一致，在此不赘述。在 Windows 系统中启动 R 之后，其操作界面如图 1－2 所示。

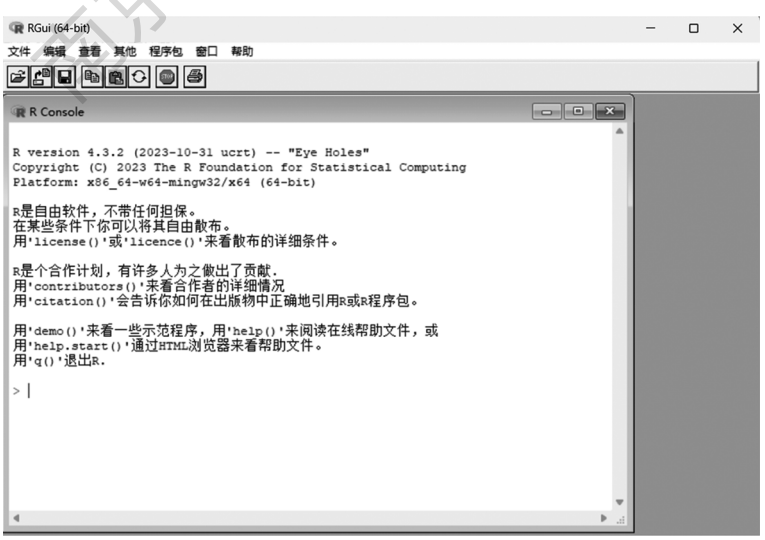

图 1－2　R 的运行界面

1.3 R 的菜单操作

R 主窗口标题栏下是菜单栏，主要有【文件】【编辑】【查看】【其他】【程序包】【窗口】【帮助】等主菜单构成。

(1)【文件】菜单

R 的【文件】菜单主要是有关 R 文件调入、保存、转换及打印等功能命令。

【运行 R 脚本文件】：调入已保存的 R 程序并输出结果；

【新建程序脚本】：建立新的 R 程序；

【打开程序脚本】：打开已保存的 R 程序；

【显示文件内容】：显示前次保存文件路径下的文件内容；

【加载工作空间】：调入已保存的工作空间，包括所有用户定义的对象(向量、矩阵、函数、数据框、列表)；

【保存工作空间】：保存运行中的工作空间，包括所有用户定义的对象(向量、矩阵、函数、数据框、列表)；

【加载历史】：调入运行记录；

【保存历史】：保存运行记录；

【改变工作目录】：改变 R 用来读取文件和保存结果的默认目录；

【打印】：打印当前的内容；

【保存到文件】：以文本文件格式储存记录；

【退出】：结束 R 的工作，并退出 R。

(2)【编辑】菜单

【编辑】菜单的基本功能为文本编辑，可以对文本进行清空、复制、粘贴和数据编辑等操作；

【复制】：复制被标记后存在剪贴板上的文本；

【粘贴】：粘贴存在剪贴板上的文本；

【仅贴命令行】：仅粘贴剪贴板上文本中的命令运行内容；

【复制并粘贴】：复制的同时粘贴文本内容；

【全选】：选择程序编辑、输出、LOG 窗口的所有内容；

【清空控制台】：清空程序窗口中的所有内容；

【数据编辑器】：对当前工作空间的数据对象进行编排；

【GUI 选项】：对图形用户界面进行设置。

(3)【查看】菜单

用于指定是否显示工具栏和状态栏。

【工具栏】：选中该选项则在主菜单栏的下方显示工具栏；

【状态栏】：选中该选项则在操作面板的下方显示状态栏。

(4)【程序包】菜单

用于加载、安装和更新程序包的窗口。

【加载程序包】:加载已经安装的程序包;

【设定 CRAN 镜像】:设定用于下载程序包的默认镜像网站;

【选择软件库】:选择镜像中具体的软件库;

【安装程序包】:安装需要运行的程序包;

【更新程序包】:对已安装的程序包进行更新;

【从本地 zip 文件安装程序包】:从已下载的程序包选择进行安装。

(5)【帮助】菜单

R 的帮助系统非常强大,可以通过多种途径寻求帮助。【帮助】菜单中有手册、网站链接、搜索等多种求助形式,学会如何使用这些帮助文档有助于将来的编程工作。R 的内置帮助系统提供了当前已安装包中所有函数的细节、参考文本以及使用示例。

图 1-3　R 的【文件】菜单

1.4　工作空间

R 语言的工作空间,是 R 运行期间专门用于存储和管理对象的特殊环境。它宛如一个“工作台”,用户能在此创建、修改、存储并操作各类数据对象,诸如向量、矩阵、数据框以及自定义函数等。深入理解工作空间的概念及其管理方法,有助于我们高效地组织与处理数据,规避命名冲突,还能在必要时迅速恢复或移除特定的数据对象。

当启动 R 语言时,系统默认会生成一个名为“Global Environment”的全局工作环境。这是多数初学者开启工作的起点,可将其视作我们的“主工作台”。在此,我们能够便捷地查看、调用和修改已创建的对象。随着分析任务渐趋复杂,我们或许需要更多定制化的

工作空间，以维持项目的结构化与有序性。在一次 R 会话结束时，当前工作空间可保存其映像。下次启动 R 时，该工作空间（包括历史代码）会自动重新加载，我们可通过上下键浏览历史代码。各类命令可在 R 命令行中以交互式方式输入，利用上下方向键查看已输入命令的历史记录，进而挑选一条之前输入过的命令并进行适当修改，最后按回车键重新执行该命令。

当前的工作目录（working directory）是 R 读取文件和保存结果的默认路径。表 1－1 罗列了 R 常用的用于管理工作空间的函数，例如，可运用函数 getwd()查看当前的工作目录，或使用函数 setwd()设定当前的工作目录。若需读入一个不在当前工作目录下的文件，则要在调用语句中写明完整路径，目录名和文件名需用引号括起来。

表 1－1　常用的 R 语言工作空间管理函数

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

1.5　程序包

R 语言本身具备大量可直接使用的功能，然而，其最具吸引力的部分功能是通过下载和安装可选模块来实现的。截至 2025 年 2 月，有超过 22 000 个由用户贡献的模块，这些模块被称为程序包（package），可从 http://cran.r-project.org/web/packages 进行下载。这些程序包能够助力用户完成各类任务，例如数据清洗、数据分析以及数据可视化等。

程序包是 R 语言的一种扩展机制，它包含了一组函数、数据集、文档以及其他资源。程序包有助于用户拓展 R 语言的功能，从而实现更为复杂的任务。R 语言社区已经开发出许多高质量的程序包，其覆盖范围广泛，涉及统计、生物信息学、机器学习等多个领域。

1. 安装程序包

在 R 语言中，可运用 install.packages()函数来安装程序包。例如，若要安装名为 ggplot2 的程序包，可运行以下代码。

代码 1－1　安装 ggplot2 程序包

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

2. 加载程序包

安装完程序包后，需要使用 library()函数将其加载到 R 的工作空间中。例如，要加载 ggplot2 程序包，可以运行以下代码。

代码 1－2　调用 ggplot2 程序包

```
> library(ggplot2)
```

3. 更新程序包

如果需要更新已安装的程序包，可以使用 update.packages()函数。例如，要更新所有已安装的程序包，可以运行以下代码。

代码 1－3　更新程序包

```
> update.packages("ggplot2")
> update.packages()
```

在上述代码中，update.packages("ggplot2")表示对程序包 ggplot2 进行更新；而 update.packages()则表示对所有已经安装的程序包进行更新。

4. 查看已安装的程序包

可以使用 installed.packages()函数查看已安装的程序包列表。例如，要查看所有已安装的程序包，可以运行以下代码。

代码 1－4　查看已经安装的程序包

```
> installed.packages()
```

执行上述代码后，将返回一个数据框，包含了已安装包的名称、版本、依赖关系等详细信息。

5. 卸载程序包

如果需要卸载某个程序包，可以使用 remove.packages()函数。例如，要卸载名为 easyanova 的程序包，可以运行以下代码。

代码 1－5　卸载已经安装的程序包

```
> remove.packages("easyanova")
```

上述代码表示卸载已经安装的 easyanova 程序包。

6. 程序包的使用

在加载了程序包后，就可以使用其中包含的函数和数据集了。例如，使用 ggplot2 程序包绘制一个简单的散点图。

代码 1－6　使用已经安装的程序包

```
> library(ggplot2)
> data(mpg)
> ggplot(data = mpg, aes(x = displ, y = hwy)) + geom_point()
```

上述代码的第一行运用 library()函数加载 ggplot2 程序包。在 R 语言里，加载程序包是为了能够使用该程序包所提供的各类功能。ggplot2 作为一个强大的数据可视化程序包，提供了丰富的绘图函数和工具。第二行借助 data()函数指定绘图所需的数据。这里指定的绘图数据集为 mpg。mpg 是 R 语言的内置数据集，其中涵盖了不同汽车的多方面信息，诸如车型、发动机排量、气缸数量、车辆重量、加速度等，还包含了对应的每加仑英里数。这些数据为后续的绘图和分析提供了丰富的素材。第三行代码先使用 ggplot()函数创建了一个绘图对象，并将 mpg 数据集指定为该绘图的数据来源。接着，利用 aes()函数进行美学映射，明确将 x 轴变量设定为发动机排量，y 轴变量设定为每加仑英里数。最后，通过 geom_point()函数添加散点图这一几何对象，从而完成散点图的构建。当执行上述代码后，会生成如图 1－4 所示的散点图，该图直观展示了发动机排量与每加仑英里数之间的关系。

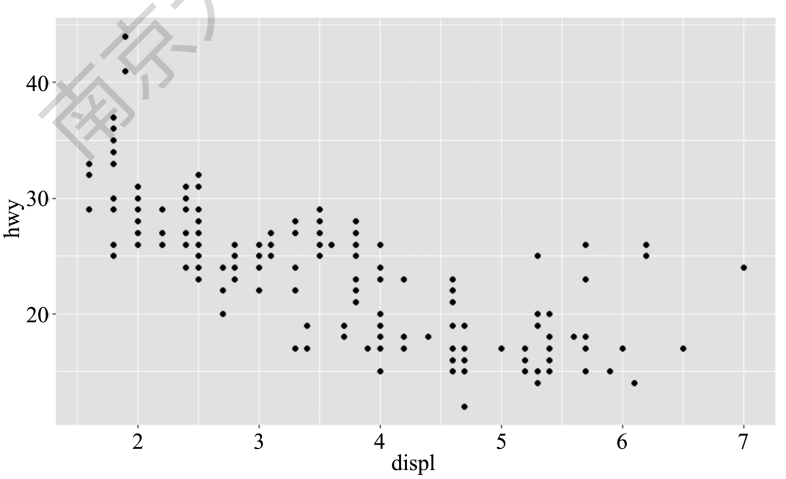

图 1－4　代码 1－6 生成的散点图

除了 CRAN 这一核心软件包仓库外，R 语言还拥有其他丰富的软件包仓库资源。以 Bioconductor(网址为：http://bioconductor.org/，如图 1－5 所示)为例，Bioconductor 仓

库专注于提供与基因组大数据分析紧密相关的工具。与从 CRAN 安装程序包不同，安装 Bioconductor 中的软件包需要借助特定的命令。当使用的 R 版本高于 3.5.0 时，可运用 BiocManager::install()命令来安装 Bioconductor 仓库中的软件包。

代码 1－7　安装 Bioconductor 仓库中的“GenomicFeatures”软件包

```
> if (! requireNamespace("BiocManager", quietly = TRUE))
      install.packages("BiocManager")
> BiocManager::install("GenomicFeatures")
```

此外，Bioconductor 网站的【Learn】板块中还提供了常见基因组数据分析相关课程和软件包使用方法的教学资料（如图 1－5），用户可以根据需要进行学习。

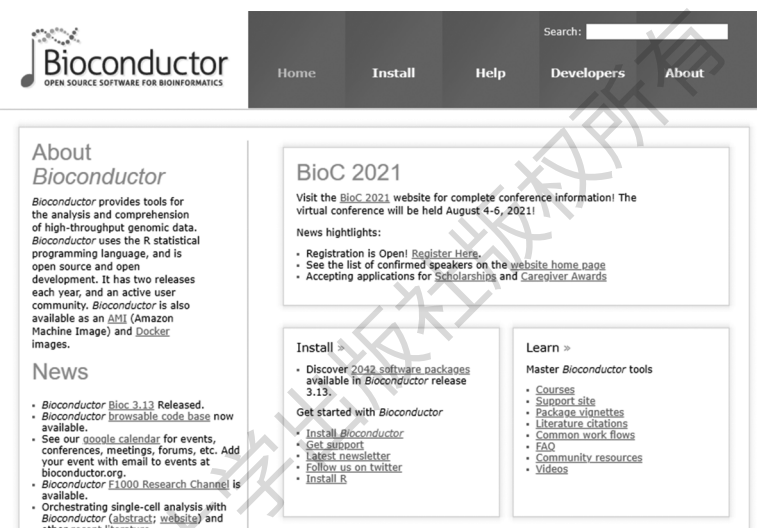

图 1－5　Bioconductor 网站首页

1.6　RStudio 的安装和使用

1.6.1　RStudio 简介

RStudio 是 R 语言的集成开发环境（Integrated Development Environment，IDE），它将强大的编程工具集成在一个直观且易于学习的界面中。作为一款集成开发环境，RStudio 并非只是一个简单的代码编辑器，它整合了代码编写、执行、调试和可视化等多种功能，极大地提升了使用 R 语言的效率和体验。RStudio 具有以下特点：

（1）界面友好：RStudio 提供了清晰、直观的用户界面，无论是新手还是经验丰富的开发者，都能轻松使用 R 语言。用户可在多个窗口分别查看代码、图表和结果，且这些窗口可按需调整。

（2）编辑便捷：RStudio 支持语法高亮、代码自动完成和智能提示功能，大幅提高了编

码效率。例如，当输入一个函数名时，RStudio 会提示函数的参数和用法，减少了查询文档的次数。

(3) 执行高效：在 RStudio 中，用户可直接在 IDE 内运行 R 代码，并实时查看结果。执行代码时，输出结果会显示在专门的窗口，便于用户查看和分析。

(4) 调试强大：RStudio 包含功能强大的代码调试工具，如断点、步进和变量检查等。这些工具助力开发者轻松定位问题和修复错误，在编写复杂程序时尤为有用。

(5) 扩展定制：RStudio 支持丰富的扩展包和定制选项，用户可根据自身需求安装扩展包来增添额外功能。RStudio 不仅有桌面版本，还有服务器版本，适用于不同的使用场景和需求。

(6) 绘图简便：RStudio 内置多种数据可视化工具，可直接生成高质量图表。这对数据分析师和研究人员而言，是非常有价值的特性。

(7) 兼容跨平台：RStudio 支持 Windows、Mac OS 和 Linux 等多个操作系统，意味着在不同平台上都能拥有良好的兼容性和一致的使用体验。

与 R 一样，RStudio 也是一个开源项目，其代码库可从 GitHub 获取(https://github.com/rstudio/rstudio)。RStudio 构建于许多其他开源项目之上，其大部分代码使用C++和 Java 编写，这两种语言并非使用 GWT(Google Web Toolkit)，GWT 主要用于将 Java 代码编译成 JavaScript，与 RStudio 使用C++和 Java 编写核心代码并无直接关联。RStudio 相关项目的具体信息可在 RStudio 的“About”对话框中查阅。

1.6.2 RStudio 的安装

在安装 RStudio 之前，需要先在当前操作系统中安装 R。用户可以从 RStudio 的官方网站(https://www.rstudio.com/)下载适配操作系统的 RStudio 软件包。常见的 RStudio 版本有桌面版和服务器版，桌面版适合单用户使用，而服务器版可通过服务器的 IP 地址进行远程连接。与桌面版相比，服务器版具有更强的灵活性，能够非常方便地完成 R 项目的部署和调试。RStudio 的安装过程与常规软件相同，在此不再详细说明。此外，也可从 RStudio 的代码库(https://github.com/rstudio/rstudio)进行安装，安装步骤在源代码附带的安装文件中。

更新 RStudio 软件也十分简便。若要查看是否有可用更新，点击“Help”下的“Check for Updates”命令，便会打开一个包含更新信息的对话框。如果有更新，用户可以先停止当前运行的程序，安装新版本，然后重新启动。

1.6.3 RStudio 的操作界面

对于桌面版本，RStudio 的启动方式与大多数其他应用程序相同。而启动服务器版本时，需要知晓服务器的 URL(Uniform Resource Locator，统一资源定位符)。RStudio Server 默认开启的端口是 8787，因此，只需在浏览器中输入服务器 URL:8787，并通过用户账户进行身份验证，即可使用 RStudio Server。在使用服务器版本时，每个用户账户仅能开启一个会话。当启动新会话时，旧的会话会断开连接，系统会发出提示。

RStudio 桌面版本的运行界面如图 1-6 所示，整体分为四个窗口，从左至右依次为程

序编辑窗口、工作空间与历史信息窗口、程序运行与输出窗口(控制台)、绘图和函数包帮助窗口。

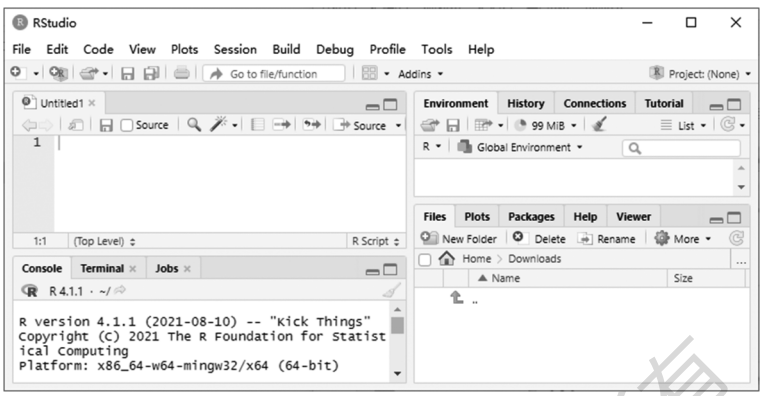

图 1－6　RStudio 的工作界面

1. 程序编辑窗口

在程序编辑窗口中,用户可以编辑、调试和储存代码,代码需保存在 R 脚本文件(后缀为.R)里。打开 RStudio 后,系统会自动生成一个标题为“Untitled1”的编辑窗口。调用【File】【New File】【R script】命令(或 Ctrl＋Shift＋N 快捷键)可以新建程序编辑窗口;调用【File】【Open File】命令,在弹出的文件选择窗口中,可选择并打开已保存的代码文件。若在刚启动时看不到该窗口,可点击调用【File】【New File】【R Script】命令将程序编辑窗口打开。

RStudio 为用户编写程序提供便捷的辅助功能,包括括号、引号的自动补齐,换行自动缩进以及函数输入辅助功能。用户可以简单地输入函数的前几位字母,再按 Tab 键,会出现所有已安装的程序包中以该字符开头的函数及其简介以供选择。完成函数输入后,还可以使用 Tab 键查看函数的各项参数说明,并且还可以通过 Ctrl＋F 快捷键实现快速查找和替换字符。完成代码编写后,通过点击保存图标或 Ctrl＋S 以保存 R 脚本,脚本默认保存在当前工作目录中。重新启动 RStudio 后,便可以打开对应的 R 脚本以继续之前的工作(或重复之前的操作)。

在运行代码时,将光标移动到目标语句的位置,点击【Run】按钮(绿色箭头),或者使用 Ctrl＋Enter 键即可运行当前的一句语句(注:运行一条完整的语句,如果语句中间有换行,会自动执行所包含的行),同时光标自动跳到下一条语句,用户可以逐条执行语句。如果需要执行整个文件中的代码,可以点击【Source】按钮,或使用 Ctrl＋Shift＋Enter 快捷键执行。

2. 工作空间和历史记录窗口

该窗口包含【Environment】(工作空间)、【History】(历史记录)、【Connections】(链接)和【Tutorial】(教程)4 个选项卡。

【Environment】选项卡:记录了当前使用的数据集、变量和自定义函数的信息,方便用户查看当前数据的状况,并且可以通过 Import Dataset 工具快速导入.xlsx、.csv 等格式的数据;

【History】选项卡：查看控制台内代码执行的历史记录；

【Connections】选项卡：轻松连接到各种数据源，如 Access、Excel、SQLServer 数据库等，方便对数据进行提取和操作；

【Tutorial】选项卡：包含了 RStudio 常用功能的教程，而且每个知识点的介绍后面都配有相应的练习题目，用户可以自行选择内容进行学习。

(3) 程序运行与输出窗口

该窗口包含【Console】(控制台)、【Terminal】(终端)、【Jobs】(任务)3 个选项卡。

【Console】选项卡：用于执行代码并显示执行结果，也可以直接在控制台中输入代码并点击回车键执行，使用 Ctrl+L 快捷键可清除控制台中的内容；

【Terminal】选项卡：提供从 RStudio IDE 内访问系统 shell 的接口，可以在其中执行 shell 的常规命令行操作，包括高级源代码控制操作、执行长时间运行的作业、远程登录和系统管理等操作；

【Jobs】选项卡：查看后台运行的任务。

(4) 绘图和帮助窗口

该窗口包含【Files】(文件)、【Plots】(图形)、【Packages】(包)、【Help】(帮助)和【Viewer】(浏览)等选项卡。

【Files】选项卡：显示当前工作路径下的文件，让用户了解所在的工作路径，便于文件读写；

【Plots】选项卡：显示输出的图形；

【Viewer】选项卡：浏览储存在本地的网页文件；

【Help】选项卡：查询函数的帮助文档；

【Packages】选项卡：安装 R 包和查阅每个包的介绍。

在【Packages】选项卡中单击【Install】图标，将会弹出如图 1-7 所示的【Install Packages】对话框。在【Packages】框中输入需要安装的包的名称，如“ggplot2”，在弹出的下拉列表中选择相应的包版本等信息(若有)，再单击【Install】按钮，即可快速安装所需要的包。此外，在【Packages】选项卡(如图 1-8)中选中所需的包，即可快速加载；若在该选项卡中选中相应的包，单击【Update】按钮，可对选中的包进行更新。

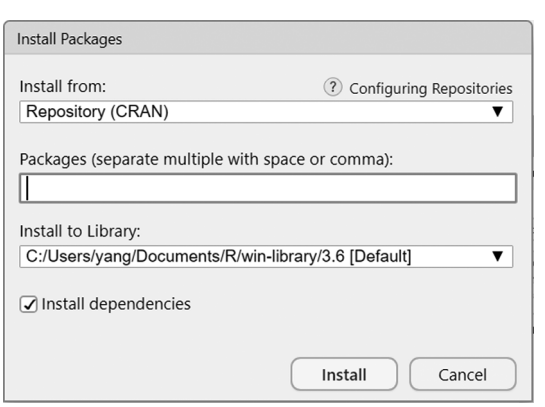

图 1-7 【Install Packages】对话框

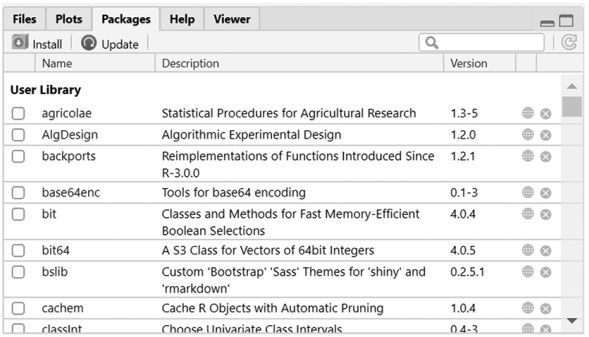

图 1－8　【Packages】选项卡

1.6.4　RStudio 中的项目管理

项目(project)是 RStudio 中提供的一项非常实用的功能，可以将不同的工作划分成若干个模块来进行管理，极大地提高工作的系统性和便捷性。如果用户进行比较复杂的数据分析工作，例如有多个表格文件作为数据源，然后在 R 中用不同的方法分析导出多个图表，这时候用户希望这些文件都集中在一起，可以使用 R project 来管理它们。

R 语言中的项目与工作目录一一对应，通过依次点击【File】【New Project】，在弹出的窗口中选取项目所在的目录，并单击【Create Project】可以创建新项目。在 RStudio 中每创建一个项目，将创建一个新文件夹并将其分配为工作目录，这样运行过程中所生成的所有文件将被分配到同一个目录中。项目中包含的所有文件可以在绘图和帮助窗口中的【Files】选项卡查看(如图 1－9)。同时，在工作目录中会生成一个后缀为.Rproj 的项目文件，这个文件包含了各种项目选项，并且能作为快捷方式进行快速启动，直接打开项目。当用户重新打开一个项目时，RStudio 会记得上次使用该项目时打开了哪些文件，并恢复工作环境，对于提高工作效率有很大帮助。

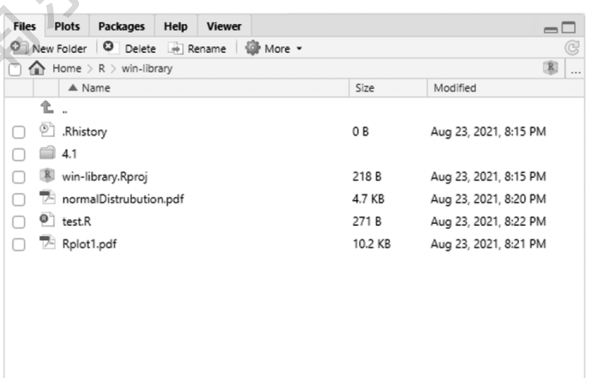

图 1－9　在【Files】选项卡中查看项目目录中的文件

习 题

1-1 R 的主要特点是什么？

1-2 到 CRAN 社区(http://cran.r-project.org/)下载并安装 R 的最新版本(中文),并尝试 R 的启动与退出。

1-3 到 RStudio 官方网站(https://www.rstudio.com/)下载并安装 RStudio 软件,在 RStudio 中完成以下操作：

(1) 查看当前的工作目录；

(2) 在计算机中选取合适的文件夹(如 H:/R_language/Rstudy),创建一个名为“第 1 章”的新项目；

(3) 新建一个程序编辑窗口,在窗口中输入以下代码并运行：

```
print ("Hello world !")
```

查看运行结果,将代码保存为“习题 1-3.R”；

(4) 在【Packages】选项卡中搜索并安装以下软件包：“Hmisc”、“pastecs”、“psych”；

(5) 在【Help】选项卡中查看函数“sum”的帮助文档。

第2章

基本数据管理

数据管理是对不同类型的数据进行收集、整理、储存、加工和检索的过程，其目的在于从大量原始数据中提取有价值的信息。R 语言作为一种强大的统计编程语言，因其丰富的功能和灵活的数据处理能力，已成为数据科学家和分析师的首选工具之一。本章将深入探讨 R 语言在数据管理中的应用，帮助读者掌握如何高效地处理和分析数据。

R 语言内置了多种数据结构，包括向量、矩阵、数组、数据框和列表等，这些结构为处理各种类型的数据提供了极大的便利。它们不仅支持快速的数据操作，还能够应对复杂的数据组织需求，例如表格数据的处理和多维数据的管理。这些数据结构的多样性和灵活性使得 R 语言能够适应从简单到复杂的各种数据处理任务。

除了强大的数据结构，R 语言在数据的读取和存储方面也表现出色。它能够直接处理各种常见的文件格式，如 CSV、Excel 和文本文件，这使得数据的导入和导出变得高效且直观。这种强大的文件处理能力使得 R 语言在处理实际数据时更加便捷和高效。

此外，R 语言丰富的运算符和函数库使得数据计算和处理变得简单易行。从基本的算术运算到高级的统计分析，R 提供了全面的工具支持。例如，通过使用 dplyr 包，用户可以轻松实现数据的筛选、排序、组合和重塑等操作，极大地提高了数据管理的效率。

2.1 创建数据集

在数据分析的起始阶段，创建一个符合个人需求且包含研究信息的数据集是至关重要的。在 R 语言中，这一过程主要涉及两个关键步骤：首先，需要选择合适的数据结构来存储数据；其次，将数据输入或导入到选定的数据结构中。

通常情况下，数据集呈现为一个矩形数组的形式，其中行代表观察值，列代表变量。值得注意的是，不同行业对于数据集中的行和列有着不同的称呼方式，例如观察值和变量、记录和字段、示例和属性等，这些术语在本质上都指向相同的概念，即数据集的基本构成单位。

R 语言能够处理多种类型的数据，包括数值型、字符型、逻辑型(TRUE/FALSE)、复数型(虚数)以及因子型。每种数据类型都有其特定的用途和适用场景，为数据分析提供了丰富的表达方式。为了满足多样化的数据存储需求，R 提供了多种用于存储数据的对象类型，如标量、向量、矩阵、数组、数据框和列表。这些对象类型在存储数据的类型、创建

方式、结构复杂度，以及用于定位和访问其中个别元素的标记等方面存在差异。选择合适的数据结构对于高效地组织和分析数据至关重要。

2.1.1 向量

向量(vector)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。单个向量中的元素必须拥有相同的数据类型(数值型、字符型或逻辑型)，而不能混合不同类型的数据。在代码 2-1 中，我们通过函数 c()分别创建一个数值型向量 v1，字符型向量 v2 和逻辑型向量 v3。

代码 2-1 创建向量

```
> v1 <- c(1, 2, 3, 10, -21, -9)
> v2 <- c("A", "B", "C")
> v3 <- c(TRUE, FALSE, TRUE, TRUE)
```

通过在方括号[]中指定元素的位置(下标)，可以访问向量中的特定元素。例如，a[2]表示访问向量 a 中的第 2 个元素；而 a[c(2, 4)]则用于同时访问向量 a 中的第 2 个和第 4 个元素。

代码 2-2 访问向量中的元素

```
> a <- c(1,3,5,7,9,11,13,15,17,19)
> a[2]
[1] 3
> a[3:6]
[1]  5  7  9  11
> a[c(2,4,6)]
[1]  3  7  11
```

在上例中，3:6 表示生成一个从 3 到 6 的数值序列，因此 a[3:6]等价于 a[c(3, 4, 5, 6)]，即提取向量 a 中第 3 到第 6 个元素。

R 语言中的 seq()函数是一个非常实用的工具，用于生成等差数列。其一般格式如下：

```
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),
    length.out = NULL, along.with = NULL, ...)
```

from：指定数列起始值；

to：指定数列结束值；

by：用于指定数列中相邻元素之间的步长；

length.out：指定所生成数列的长度；

along.with：也指定所生成数列的长度，通常与另一个向量结合使用，以生成一个与该向量长度相同的序列。

需要注意的是，在 seq()函数中，by、length.out 和 along.with 这三个参数只能选择其中一个使用。这要求用户根据具体需求灵活选择适当的参数，以达到预期的序列生成效果。

代码 2－3　等差序列在创建向量中的应用

```
> seq(0, 1, length.out = 11)
  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> seq(1, 9, by = 2)
[1] 1 3 5 7 9
> seq(1, 9, by = pi)
[1] 1.000000 4.141593 7.283185
```

上例中，第 1 个语句表示数列的起始值为 0，结束值为 1，等差序列中包含 11 个数值；第 2 个语句表示数列的起始值为 1，结束值为 9，步长为 2；第 3 个语句表示数列的起始值为 1，结束值为 9，步长为 π 值(3.1415926)。

2.1.2　矩阵

矩阵(matrix)是 R 语言中的一种二维数组，其所有元素必须具有相同的数据类型(数值型、字符型或逻辑型)。R 通过函数 matrix()创建矩阵，其一般格式如下：

```
mymatrix < - matrix(vector, nrow = number, ncol = number,
          byrow = logical_value, dimnames = list (vector_rnames, vector_cnames))
```

vector：向量，包含了矩阵的元素；

nrow、ncol：数值，用以指定矩阵的行数和列数；

byrow：逻辑值，用于设置矩阵按行填充(TRUE)还是按列填充(FALSE)，默认情况下按列填充；

dimnames：包含两个字符型向量的列表，分别记录行名和列名。

在下面的示例中，第一部分利用 matrix()函数创建一个 4 行 5 列的矩阵，包含从 1 到 20 的自然数，使用按列填充的方式进行填充；第二部分创建了一个 3 行 3 列的包含行列名称标签的矩阵，并按行进行填充。代码和运行结果如下：

代码 2－4　创建矩阵

```
> x <- matrix (1:20, nrow = 4, ncol = 5, byrow = FALSE)
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> data <- c(1, 2, 3, 11, 12 ,13, 21, 22, 23)
> rowname <- c("R1", "R2", "R3")
> colname <- c("C1", "C2", "C3")
> y <- matrix (data, nrow = 3, ncol = 3, byrow = TRUE, dimnames = list (rowname, colname))
> y
```

```
    C1  C2  C3
R1   1   2   3
R2  11  12  13
R3  21  22  23
```

在 R 语言中，矩阵元素的提取非常灵活，可以通过下标和方括号来选择矩阵中的特定行、列或元素。具体来说，X[i,]表示提取矩阵 X 中的第 i 行，X[,j]表示提取第 j 列，而 X[i,j]则表示提取第 i 行第 j 列的元素。当需要选择多行或多列时，下标 i 和 j 可以是数值型向量。在下面的示例中，我们创建一个 5×6 的矩阵，然后分别使用下标提取矩阵中的不同元素，代码和运行结果如下：

代码 2－5　使用矩阵下标提取矩阵中的元素

```
> a1 <- matrix (1 : 30, nrow = 5, ncol = 6)
> a1
     [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,]   1     6     11    16    21    26
[2,]   2     7     12    17    22    27
[3,]   3     8     13    18    23    28
[4,]   4     9     14    19    24    29
[5,]   5    10     15    20    25    30
> a1[3,]
[1]  3  8  13  18  23  28
> a1[,2]
[1]  6  7  8  9  10
> a1[3, c(2,3)]
[1]  8  13
> a1[, c(2,3)]
      [,1][,2]
[1,]     6    11
[2,]     7    12
[3,]     8    13
[4,]     9    14
[5,]    10    15
```

在上述代码中，a1[3, c(2, 3)]表示提取矩阵 a1 中第 3 行的第 2 和第 3 列的元素，而 a1[, c(2, 3)]表示提取矩阵 a1 中第 2 列和第 3 列的所有元素。

矩阵运算是数据分析和统计建模中的重要工具。R 提供了丰富的矩阵运算功能，包括基本的算术运算、矩阵乘法、转置、求逆等。具体而言，当两个矩阵具有相同的维度时，它们可以进行加法运算。矩阵乘法则需要满足特定的条件，即左侧矩阵的列数必须等于右侧矩阵的行数。此外，R 语言中的 t()函数能够对矩阵进行转置操作。对于方阵，即行数和列数相等的矩阵，可以使用 solve()函数来求其逆矩阵，但需注意，只有可逆矩阵才能

进行此操作。这些运算使得矩阵的处理和分析变得简单而高效，为复杂的数学计算和数据处理提供了强大的支持。

代码 2－6　矩阵的运算

```
> mat1 <- matrix(1:6, nrow = 2, ncol = 3)
> mat2 <- matrix(1:6, nrow = 2, ncol = 3)
> mat_sum <- mat1 + mat2  #矩阵相加
> mat_sum
     [,1] [,2] [,3]
[1,]    2    6   10
[2,]    4    8   12
> mat_transpose <- t(mat1)  #矩阵转置
> mat_transpose
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> mat_prod <- mat1 %*% t(mat2)  #矩阵乘法
> mat_prod
     [,1] [,2]
[1,]   35   44
[2,]   44   56
> mat3 <- matrix(1:4, nrow = 2, ncol = 2)
> mat_inverse <- solve(mat3)  #求逆矩阵
> mat_inverse
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
```


在上述示例中，我们首先构建了两个矩阵，分别是 mat1 和 mat2，它们均为 2 行 3 列，其元素依次为 1 至 6，并对这两个矩阵执行了逐元素相加的操作，从而得到了新的矩阵 mat_sum；随后运用 t()函数对 mat1 进行了转置处理；接着进行了矩阵乘法运算，将 mat1 与 mat2 的转置矩阵相乘；最后创建了一个 2 行 2 列的矩阵 mat3，其元素为 1 到 4，并对该矩阵求解了其逆矩阵。

2.1.3　数组

数组(array)是矩阵的扩展形式，它允许维度超过二维。在 R 语言中，数组可以通过 array()函数创建，其一般格式如下：

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

vector:数值型向量,包含了数组中的数据;

dimensions:数值型向量,指定了数组在各个维度上的大小;

dimnames:列表,包含各维度名称标签,此项为可选项。

接下来,我们使用 array()函数创建一个具有三个维度的数组。其中,dim1、dim2 和 dim3 这三个维度分别具有 2、3 和 4 个水平,数组中的元素为 1 到 24 的自然数。

代码 2-7　使用 array()函数创建数组

```
> dim1 <- c("X1","X2")
> dim2 <- c("Y1","Y2","Y3")
> dim3 <- c("Z1","Z2","Z3","Z4")
> xyz <- array(1:24, c(2,3,4), dimnames = list(dim1, dim2, dim3))
> xyz
, , Z1
     Y1  Y2  Y3
X1    1   3   5
X2    2   4   6
, , Z2
     Y1  Y2  Y3
X1    7   9  11
X2    8  10  12
, , Z3
     Y1  Y2  Y3
X1   13  15  17
X2   14  16  18
, , Z4
     Y1  Y2  Y3
X1   19  21  23
X2   20  22  24
```

数组是矩阵的自然延伸,与矩阵类似,数组中的数据也必须具有相同的数据类型。提取数组中的元素可以参考矩阵的方式,通过下标进行提取。

代码 2-8　提取数组中的元素

```
> xyz[2,2,2]
[1] 10
> xyz[2,2,]
Z1  Z2  Z3  Z4
 4  10  16  22
> xyz[,2,2]
X1  X2
 9  10
```

在上述例子中，我们展示了如何提取一个三维数组的单个元素，以及在指定两个维度的值后，如何获取第三个维度的所有元素。

2.1.4　数据框

在使用 Excel、SAS 和 SPSS 等软件处理数据时，我们通常会在不同的列中存储不同类型的数据。在 R 语言中，这种需求可以通过创建数据框来实现。数据框是 R 语言中一种非常灵活的数据结构，它类似于表格，每一列可以包含不同类型的值，如数值型、字符型或逻辑型等。数据框可以通过函数 data.frame()创建，其一般格式如下：

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

col1，col2，col3，...：向量，可为任何类型（如字符型、数值型或逻辑型）。

在创建数据框时，每一列的数据类型必须是唯一的，但可以将不同数据类型的列组合在一起。由于数据框的结构与分析人员通常设想的数据集形态较为接近，因此在讨论数据框时，我们会交替使用“列”和“变量”这两个术语。

在下面的例子中，我们将创建一个包含 3 个变量的数据框 rice_data，其中 1 个变量为数值型，2 个变量为字符型。以下是代码和运行结果：

代码 2－9　创建数据框

```
> height <- c(120, 115, 130, 125, 118, 122)
> genotype <- c("Aa", "aa", "aa", "AA", "Aa", "Aa")
> disease_resistance <- c("S", "R", "R", "S", "S", "S")
> rice_data <- data.frame(height, genotype, disease_resistance)
> print(rice_data)
      height  genotype  disease_resistance
1      120     Aa        S
2      115     aa        R
3      130     aa        R
4      125     AA        S
5      118     Aa        S
6      122     Aa        S
```

与矩阵类似，数据框中的元素、行和列也可以通过下标和方括号来提取。如果在方括号中只包含一个向量，则默认提取相应的列，此时向量可以是数值型向量，也可以是包含列名称的字符型向量。此外，还可以通过符号 $ 来选取一个给定数据框中的某个特定变量。

代码 2－10　提取数据框中的元素和变量

```
> rice_data [2, 2]
[1] "aa"
> rice_data [2 : 3]
  genotype disease_resistance
```

```
1           Aa                  S
2           aa                  R
3           aa                  R
4           AA                  S
5           Aa                  S
6           Aa                  S
> rice_data[c("height", "genotype")]
  height genotype
1     120       Aa
2     115       aa
3     130       aa
4     125       AA
5     118       Aa
6     122       Aa
> rice_data$disease_resistance
[1] "S" "R" "R" "S" "S" "S"
```

当频繁使用 rice_data 数据框中的某个变量时，在每个变量名前都键入一次 rice_data$ 可能会显得比较麻烦，在 R 中可以联合使用函数 attach()和 detach()来简化代码。函数 attach()可将数据框添加到 R 的搜索路径中。当 R 遇到一个变量名以后，将检查搜索路径中的数据框，以定位到这个变量。这里通过一个简单的例子展示 attach()和 detach()函数的用法。

代码 2－11　使用 attach()和 detach()函数添加和解除数据索引

```
> attach(rice_data)
The following object is masked _by_ .GlobalEnv:
    disease_resistance
> height
[1] 120 115 130 125 118 122
> detach(rice_data)
```

如上例所示，在建立了 rice_data 数据框之后，用 attach()函数将数据框 rice_data 添加到 R 的搜索路径中，可以直接使用该数据框中的变量名。在该路径不再使用时，使用函数 detach()可以将数据框从搜索路径中移除。

2.1.5　因子

因子(factor)是一类特殊的数据格式，用于对数据进行分类，并将分类信息作为水平(level)存储的数据对象，也称为类别变量。根据因子中各水平的排列是否有固定的先后顺序，又进一步分为有序型因子和无序型因子。

因子在 R 中非常重要，因为它从数据结构层面影响着后续的分析方式、结果的展示及结果的统计学意义。在数据的初步处理阶段，我们就需要根据实际情况和分析目的来

决定数据中哪些变量应设置为因子，以及如何对因子进行考查。例如，在两因素完全随机试验数据中，应包含试验因素 A 和试验因素 B 两个因子；而在两因素随机区组试验数据中，应包含试验因素 A、试验因素 B 和区组三个因子。

使用 factor()函数可以创建因子，其基本用法如下：

```
factor(x = character(), levels =, labels =, exclude = NA, ordered = TRUE)
```

x：字符型向量，如果 x 不是字符型向量，可以使用 as.character(x)把 x 转换为字符型向量；

levels：字符型向量，设置因子包含的水平，默认值是向量 x 中的所有唯一值；

labels：字符型向量，设置因子水平的标签，相当于对因子水平重命名；

exclude：字符型向量，设置向量 x 中需要排除的字符；

ordered：逻辑值，TRUE 表示创建有序型因子。

factor()函数以整数向量的形式存储因子的水平值，整数的取值范围是 1 到 k(其中 k 是向量中唯一值的个数)。同时，一个由字符串(原始值)组成的内部向量会映射到这些整数上。例如，对于字符型向量 genotype <- c("Aa", "aa", "aa", "AA", "Aa", "Aa")，执行 genotype <- factor(genotype)后，该向量会被存储为(2, 1, 1, 3, 2, 2)，并在内部将其关联为 1 = aa、2 = Aa、3 = AA(赋值顺序根据字母排序而定，小写字母在前)。在对向量 genotype 进行分析时，它会被视为名义型变量，并自动选择适合这一测量尺度的统计方法。若要生成有序型因子，需要为 factor()函数指定参数 ordered＝TRUE。因子水平的顺序默认按字母顺序排序，但在某些情况下，我们可能希望指定水平的先后顺序。此时，可以通过设置 levels 参数来覆盖默认排序。例如：genotype <- factor(genotype, ordered = TRUE, levels = c("AA", "aa", "Aa"))，各水平的赋值将为 1＝AA、2＝aa、3＝Aa。这样在进行数据统计和绘图时，因子的各水平将根据用户指定的顺序排列。

在创建因子时，需要注意确保指定的水平与数据中的真实值相匹配。因为任何在数据中出现但未在参数中列举的值都将被设为缺失值。接下来，我们将创建一个包含普通向量和因子型向量的数据框，并使用 str()函数和 summary()函数来分析其具体信息。其中，str()函数可以提供 R 中某个对象的结构信息，而 summary()函数则根据变量的类型返回不同的结果：对于数值型变量，返回其最小值、最大值、平均值、中位数和各四分位数；而对于因子型变量，则返回其各水平的频数。

代码 2－12　创建含有因子型向量的数据框并使用 str()和 summary()函数分析具体信息

```
> height <- c(120, 115, 130, 125, 118, 122)
> genotype <- c("Aa", "aa", "aa", "AA", "Aa", "Aa")
> disease_resistance <- c("S", "R", "R", "S", "S", "S")
> disease_resistance <- factor(disease_resistance)
> genotype <- factor (genotype, order = TRUE)
> rice_data <- data.frame(height, genotype, disease_resistance)
```

```
> str(rice_data)
'data.frame':6 obs. of  3 variables:
$ height              : num  120 115 130 125 118 122
$ genotype            : Ord.factor w / 3 levels "aa"<"Aa"<"AA": 2 1 1 3 2 2
$ disease_resistance: Factor w / 2 levels "R","S": 2 1 1 2 2 2
> summary(rice_data)
     height        genotype disease_resistance
Min.   :115.0    aa：2    R：2
1st Qu.:118.5    Aa：3    S：4
Median :121.0    AA：1
Mean   :121.7
3rd Qu.:124.2
Max.   :130.0
```

在上例中，首先以向量的形式输入了数据。然后，通过 factor()函数将向量 disease_resistance 和 genotype 分别指定为一个无序型因子和一个有序型因子。最后，将三个向量合并为一个数据框。使用 str()函数检查数据框的结构，显示数据框包含 6 个观测和 3 个变量，其中变量 height 为数值型。genotype 为有序因子，水平为"aa"、"Aa"和"AA"，内部编码分别为 1、2 和 3；disease_resistance 为无序因子，水平为"R"和"S"，内部编码分别为 1 和 2。使用 summary()函数生成数据框的统计摘要，结果显示了连续型变量 height 的最小值、最大值、平均值、中位数和各四分位数，以及因子型变量 disease_resistance 和 genotype 各水平的频数值。

2.1.6 列表

列表(list)是 R 的数据类型中最为复杂的一种。一般来说，列表就是一些对象(或成分，component)的有序集合。列表允许用户整合若干(可能无关的)对象到单个对象名下。例如，某个列表中可能是若干向量、矩阵、数据框，甚至其他列表的组合。在 R 中，使用函数 list()创建列表，其一般格式如下：

```
mylist <- list(object1, object2, object3, ...)
```

object1、object2、object3：列表中所要包含的对象。

下面我们通过一个简单的实例创建一个列表，列表中包含 4 种类型的对象：a 为一个字符串，b 为一个数值型向量，c 为一个矩阵，d 为一个因子型向量。

代码 2-13 建立一个包含 4 种对象类型的列表

```
> a <- "This is a list"
> b <- c(1:10)
> c <- matrix(21:40, nrow = 5, byrow = FALSE)
> d <- c("A", "A", "B", "B")
> d <- factor(d)
```

```
> mylist <- list(a, b, c, d)
> mylist
[[1]]
[1] "This is a list"

[[2]]
 [1]  1  2  3  4  5  6  7  8  9  10

 [[3]]
     [,1] [,2] [,3] [,4]
[1,]   21   26   31   36
[2,]   22   27   32   37
[3,]   23   28   33   38
[4,]   24   29   34   39
[5,]   25   30   35   40

[[4]]
[1] A A B B
Levels: A B
```

用户在列表中可以组合任意多的对象。另外，可以通过在双重方括号中指明代表某个成分的数字或名称来访问列表中的元素。在上例中，我们可以使用 mylist[[3]]表示矩阵 c。

列表是 R 语言中一种重要的数据结构。首先，列表可用于组织不同类型的数据，形成一个复合数据结构。例如，一个学生的信息可以包含姓名、年龄、成绩等多个字段，这些字段可以存储在一个列表中。其次，许多 R 函数的运行结果都以列表的形式返回，这使得函数能够同时返回多个不同类型的结果，分析人员可以根据需要选择性地提取列表中的特定元素。

2.2　数据的读取和存储

2.2.1　数据的读取

在 R 语言中，数据导入是数据分析、可视化及模型构建过程中的第一步，也是至关重要的一步。R 提供了多种内置函数和包来支持从各种数据源读取数据，包括文本文件（如 CSV、TXT）、Excel 文件、数据库以及统计软件产生的数据文件等。

（1）read.table()函数

read.table()函数是一个强大的工具，主要功能是从指定的文本文件中读取数据，这

些数据按照行和列的形式组织，每行代表一个观测值，每列代表一个变量。函数读取数据后，会将这些数据转换为一个数据框(data frame)，其中包含了文件中的所有数据，并且可以根据需要进行进一步的数据处理和分析。其语法格式和主要参数为：

```
read.table(file , header = FALSE, sep = "", quote = "\"",
                dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
                row.names, col.names, as.is = ! stringsAsFactors, tryLogical = TRUE,
                na.strings = "NA", colClasses = NA, nrows = - 1,
                skip = 0, check.names = TRUE, fill = ! blank.lines.skip,
                strip.white = FALSE, blank.lines.skip = TRUE,
                comment.char = "#",
                allowEscapes = FALSE, flush = FALSE,
                stringsAsFactors = FALSE,
                fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

file：字符型向量，表示要读取的文件名(包含路径)或可访问的连接(URL)。

header：逻辑值，指示文件的第一行是否包含列名。如果为 TRUE，则第一行会被用来作为数据框的列名。默认值为 FALSE。

sep：字符型向量，用于指定字段之间的分隔符。默认值为"(空格、制表符、换行符、回车符等都可以作为字段分隔符)。可以指定任何单个字符或字符串作为分隔符，如","(逗号分隔值，CSV 文件)、"\ t"(制表符分隔值，TSV 文件)等。

quote：控制文件中引号使用的字符(通常是单引号或双引号)。如果设置为""(默认值)，则不使用引号来标识字段。如果设置为其他字符(如"或'或')，则这些字符将被用来标识可能包含分隔符、换行符或引号本身的字段。

dec：字符型向量，指定小数点字符。默认值是.(点号)。如果数据文件使用不同的小数点字符(如逗号)，则需要设置这个参数。

numerals：字符串向量，用于指定如何处理可能导致舍入损失的数字。选项包括“alow、loss”(允许损失)、“warn、loss”(警告损失)和“no.loss”(不允许损失)。

row.names：一个整数向量，指定哪一列或哪些列应该被用作行名。也可以是一个字符向量，给出行名的显式值。如果省略，则会自动生成行名。

col.names：字符向量，如果 header＝FALSE，则可以提供列名。

as.is：逻辑向量或逻辑值，用于指定哪些列应该按原始格式读取(即不将字符向量转换为因子)。如果设置为 TRUE，则所有列都会被按原始格式读取。

na.strings：字符向量，指定哪些字符串应该被视为缺失值(NA)。默认值是"NA"(不区分大小写)。

colClasses：字符向量，用于指定每列的类(如"numeric"、"character"、"factor"等)。这可以帮助 R 更准确地读取和存储数据。

nrows：整数，指定要读取的行数。这对于跳过文件末尾的部分数据很有用。

skip：整数，指定在读取数据之前要跳过的行数。

strip.white：逻辑值，指定在读取之前是否应该删除字段两端的空格。默认值为

FALSE。

blank.lines.skip：逻辑值，指定是否应该跳过空行。默认值为 TRUE。

comment.char：字符型向量，指定注释字符（如 #），该字符开头的行将被视为注释并跳过。

allowEscapes：逻辑值，指定是否应该允许转义字符（如\n 表示换行）。这通常用于读取包含特殊字符的字符串。

flush：逻辑值，如果 file 是一个连接（而不是文件名），并且希望每次读取后都刷新该连接，则设置为 TRUE。

fill：逻辑值，如果文件中的某些行比其他行包含更少的字段，则设置为 TRUE 以填充缺失的字段。默认值为 FALSE。

stringsAsFactors：逻辑值，在 R 4.0.0 之前的版本中，这个参数默认为 TRUE，意味着字符向量会被自动转换为因子。在 R 4.0.0 及以后的版本中，默认值为 FALSE。需要注意 read.table()函数本身没有 stringsAsFactors 参数；这个参数是 read.csv()和其他读取函数的参数，但 read.table()的行为可以通过 colClasses 或 as.is 参数来控制。

fileEncoding：字符串，指定文件的编码格式。默认值为空字符串，表示使用系统默认编码。可以指定其他编码格式，如"UTF-8"或"latin1"。

text：字符串，表示直接从给定的文本字符串读取数据，而不是从文件中读取。如果提供 text 参数，则 file 参数将被忽略。

skipNul：逻辑值，表示是否跳过文件中的空字符。默认值为 FALSE。

例 2-1　打开数据文件 Example2_1.txt，并将其中的数据读入到 mydata 的数据框中。Example2_1.txt 文件中，数值之间使用制表符分开。

代码 2-14　利用 read.table()函数导入数据

```
> mydata <- read.table("E:/RinBio/Chapt2/Example2_1.txt",sep ="\t",header = TRUE)
> mydata
     Name  Chinese  Mathematics  English  Physics  Chemistry
1 Zhangsan     88          83       93       78         84
2     Lisi     91          99       91       76         91
3   Wangwu     90          79       74       75         80
4  Zhaoliu     95          98       77       87         78
5    Liuqi     94          75       75       78         86
6   Chenba     80          89       71       97         77
```

在上例中，利用 read.table()函数将 Example2_1.txt 文件中的数据导入到了 mydata 数据框中。

（2）read.csv()函数

read.csv()是最常见的读取数据函数之一，用于处理以逗号作为字段分隔符的文本文件，而 read.csv2()则是为处理使用分号作为字段分隔符的 CSV 文件设计的。read.csv()

的基本语法格式为：

```
read.csv(file , header = TRUE, sep = ",", quote = "\"",
             dec = ".", fill = TRUE, comment.char = "", ...)
```

read.csv()函数的参数与 read.table()函数的释义相近，在此不做赘述。

例 2－2　打开数据文件 Example2_2.csv，并将其中的数据读入到 mydata 的数据框中。

代码 2－15　利用 read.csv()函数导入数据

```
> mydata <- read.csv("E:/RinBio/Chapt2/Example2_2.csv", header = TRUE)
> mydata
      Name  Chinese  Mathematics  English  Physics  Chemistry
1 Zhangsan       88           83       93       78         84
2     Lisi       91           99       91       76         91
3   Wangwu       90           79       74       75         80
4  Zhaoliu       95           98       77       87         78
5    Liuqi       94           75       75       78         86
6   Chenba       80           89       71       97         77
```

在上例中，利用 read.csv()函数将 Example2_2.csv 文件中的数据导入到了 mydata 数据框中。

若使用 read.table()函数导入 CSV 数据文件，则需用 sep 参数指定分隔的符号为逗号(“,”)。由此可见，在读取数据文件时，read.table()函数更为通用。

值得注意的是，R 没有内置的读取 Excel 文件的函数，但 readxl 包提供了 read_excel()函数，用于读取.xlsx 和.xls 文件。此外，openxlsx 包中的 read.xlsx()函数也可用于读取 Excel 数据文件。

2.2.2　数据的存储

在分析过程中，我们常常需要将软件的计算结果或经过编辑的数据保存到文件中，以便于在退出 R 后查阅这些数据。使用 write.table()函数可以很方便地将表格型数据写入指定文件，write.table()的一般用法如下：

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = "",
              eol = "\n", na = "NA", dec = ".", row.names = TRUE,
              col.names = TRUE, qmethod = c("escape", "double"),
              fileEncoding = "")
```

x：要写入文件的对象，通常是数据框(data frame)或矩阵(matrix)。

file：字符串，指定输出文件的路径和文件名。如果为空字符串(默认值)，则输出到控制台。

append：逻辑值，表示是否将内容追加到已存在的文件中。默认值为 FALSE，即如果文件已存在，会覆盖原文件。

quote：逻辑值，TRUE 表示输出的字符型和因子型向量会带有双引号。

sep：字符串，指定字段分隔符。默认值为"（空格）。可以设置为其他字符，如逗号（,）或制表符（\t）。

eol：字符串，指定每行的结束符。默认值为"\n"（换行符）。

na：字符串，指定如何表示缺失值 NA。

dec：字符串，指定小数点的字符。默认值为"."。

row.names：逻辑值，表示是否写入行名。

col.names：逻辑值，表示是否写入列名。

qmethod：字符串，指定如何处理引号。取值为"escape"（默认值）或"double"，如果为"escape"，则在引号内再加一个引号；如果为"double"，则用双引号括起来。

fileEncoding：字符串，指定文件的编码格式。默认值为空字符串，表示使用系统默认编码。可以指定其他编码格式，如"UTF - 8"或"latin1"。

代码 2 - 16　利用 write.table()函数将数据框数据保存到 Code2_1.txt 文件中

```
> df <- data.frame(ID = 1 : 5,
+                  Name = c("Alice", "Bob", "Charlie", "David", "Eve"),
+                  Age = c(24, 27, 22, 32, 29))
> print(df)
  ID       Name Age
1  1       Alice  24
2  2       Bob    27
3  3       Charlie  22
4  4       David  32
5  5       Eve    29
> write.table(df, file = "E: / RinBio / Chapt2 / Code2_1.txt", sep = ",")
```

在该代码中，首先利用 data.frame()函数构建了一个数据框 df，并用 write.table()函数将数据框 df 中的内容保存到了 Code2_1.txt，数值之间的分隔采用的是逗号（“,”）。

除 write.table()函数外，也可以使用 write.csv()函数，将数据输出为.csv 格式的文件。

2.2.3　内置数据集

R 语言中的内置数据集是 R 语言安装时自动包含的数据集，这些数据集覆盖了多个领域，如生物学、经济学、地理学等，非常适合用于数据分析、数据可视化、统计建模和机器学习的练习。以下是一些常见的 R 语言内置数据集的简介：

（1）iris（鸢尾花数据集）：常用于机器学习和统计分类的示例数据集；包含 5 个变量（花萼长度、花萼宽度、花瓣长度、花瓣宽度、品种名称）、150 条记录；常用于分类算法的教学和测试。

（2）mtcars（汽车性能数据集）：记录了关于不同型号汽车的性能数据；包含 32 辆汽车的性能参数，如马力、重量、加速度等；适合用于回归分析、聚类分析等。

(3) ChickWeight(小鸡体重数据集):记录了不同饲料和时间点下小鸡的体重数据;包含了小鸡在不同饮食条件下体重变化的数据;适合用于时间序列分析、生长模型等。

(4) AirPassengers(航空旅客数据集):包括航空公司 1949 年至 1960 年的每月国际航空旅客数量;包含每月的旅客数量数据;常用于时间序列分析和预测。

(5) swiss(瑞士人口数据集):瑞士不同省份的生育率和社会经济指标;适合用于回归分析、相关性分析等。

(6) faithful(喷泉喷发时间数据集):记录了 Old Faithful 喷泉的喷发时间和间隔时间;包含喷泉每次喷发的等待时间和持续时间;适合用于时间间隔数据的分析。

(7) trees(树木生长数据集):包含了 3 种不同类型的树木的直径、高度和体积数据;提供了树木的生长数据;适合用于回归分析、数据可视化等。

(8) quakes(地震数据集):包含 1964 年至 1975 年新西兰地震的相关信息;包括地震的震级、位置等数据;适合用于地理数据分析、地震学研究等。

(9) economics(美国宏观经济数据集):包含一些美国的经济指标数据;提供了如失业率、工资等宏观经济数据;适合用于宏观经济分析、时间序列分析等。

(10) USArrests(美国各州暴力犯罪数据集):美国各州的暴力犯罪数据;包含了每个州的人口、暴力犯罪率等指标;适合用于犯罪数据分析、地理数据分析等。

使用 data()命令可以查看全部的数据集信息,而使用 help()命令可以查阅每个数据集的具体信息,包括数据的含义、数据集格式、每个变量的属性以及数据来源等。例如,在 RStudio 中通过 help("iris")命令查看数据集 iris,会在绘图和帮助窗口中显示以下信息(如图 2-1)。

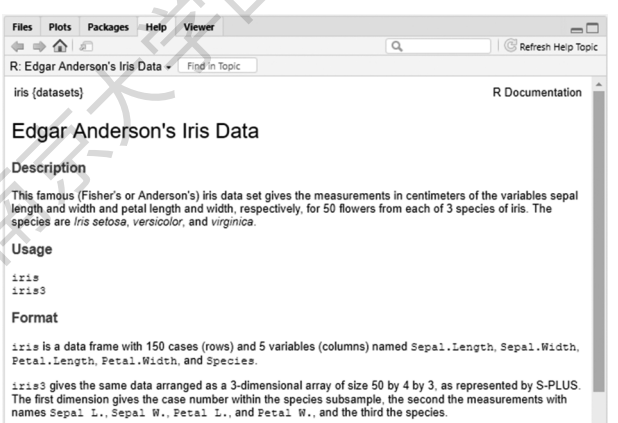

图 2-1　内置数据集 iris 的帮助信息

帮助信息显示,iris 数据集中包含了三个鸢尾花品种 *setosa*、*versicolor* 和 *virginica* 的花瓣长、宽以及花萼长、宽的数据,单位为厘米,数据集格式为数据框,包含 5 个变量、150 条记录。而 iris3 数据集格式为数组,包含 3 个维度,每个维度的元素数目分别为 50、4、3。

2.3　R 的运算符

运算符是一些符号，告诉编译器执行特定的数学或逻辑操作。R 语言有丰富的内置运算符，并提供了算术运算符、关系运算符、逻辑运算符等。

1. 算术运算符

算术运算符是指可以在程序中实现如加、减、乘、除等数学运算的运算符号。算术运算符的表示方法及其含义如表 2－1 所示。

表 2－1　R 的算术运算符

运算符	含义	示例
+	加	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v + t) [1] 10.0　8.5　10.0
-	减	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v - t) [1] - 6.0　2.5　2.0
*	乘	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v *t) [1] 16.0 16.5 24.0
/	除	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v / t) [1] 0.250000 1.833333 1.500000
%%	求余	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v %% t) [1] 2.0 2.5 2.0
%/%	整除	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v %/% t) [1] 0 1 1
^	乘幂	v <- c(2, 5.5, 6) t <- c(8, 3, 4) print(v ^ t) [1]　256.000　166.375 1296.000

2. 比较运算符

比较运算符用于建立两个变量之间的一种关系，并要求 R 确定这种关系是否成立，如果成立，输出的结果是 TRUE，若不成立，则输出 FALSE。

表 2-2　R 的比较运算符

运算符	含义	示例
>	大于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v > t) [1] FALSE TRUE FALSE FALSE
<	小于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v < t) [1] TRUE FALSE TRUE FALSE
==	等于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v == t) [1] FALSE FALSE FALSE TRUE
<=	小于等于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v <= t) [1] TRUE FALSE TRUE TRUE
>=	大于等于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v >= t) [1] FALSE TRUE FALSE TRUE
!=	不等于	v <- c(2, 5.5, 6, 9) t <- c(8, 2.5, 14, 9) print(v != t) [1] TRUE TRUE TRUE FALSE

3. 逻辑运算符

在 R 语言中，逻辑运算符用于执行逻辑操作，例如比较、条件判断和逻辑组合。对于数字，非零数值的逻辑值为 TRUE，而零的逻辑值为 FALSE。当对向量进行逻辑运算时，会将第一个向量的每个元素与第二个向量的相应元素进行比较，返回一个包含逻辑值的向量。逻辑运算符在数据分析、条件判断和流程控制中非常有用，能够帮助用户根据条件筛选数据、执行分支操作或进行复杂的逻辑判断。

表 2-3　R 的逻辑运算符

运算符	含义	示例
&	元素逻辑与运算符。它结合第一向量的每个元素与第二向量的相应元素，如果这两个元素都为 TRUE 则输出 TRUE。	v <- c(3, 1, TRUE, 2 + 3i) t <- c(4, 1, FALSE, 2 + 3i) print(v & t) [1]　TRUE TRUE FALSE TRUE
|	元素逻辑或运算符。它结合第一向量的每个元素与第二向量的相应元素，如果两个元素中有一个为 TRUE 则输出 TRUE。	v <- c(3, 0, TRUE, 2 + 2i) t <- c(4, 0, FALSE, 2 + 3i) print(v | t) [1]　TRUE FALSE TRUE TRUE
!	逻辑非运算符。取向量的每个元素，并给出了相反逻辑值。	v <- c(3, 0, TRUE, 2 + 2i) print(! v) [1]　FALSE TRUE FALSE FALSE
&&	逻辑与运算符。取两个向量的第一元素，仅当两个都为 TRUE 时结果为 TRUE。	v <- c(3, 0, TRUE, 2 + 2i) t <- c(1, 3, TRUE, 2 + 3i) print(v && t) [1]　TRUE
||	逻辑或运算符。取两个向量的第一元素，并且如果有一个为 TRUE 时结果为 TRUE。	v <- c(0, 0, TRUE, 2 + 2i) t <- c(0, 3, TRUE, 2 + 3i) print(v || t) [1]　FALSE

2.4　R 常用函数及其应用

在 R 语言中，函数是组织在一起的一组用以执行特定任务的语句。R 语言提供了大量的内置函数，同时用户也可以创建自己的函数。在 R 中，函数是一个对象，这意味着 R 语言解释器可以将控制权传递给函数，并提供函数完成任务所需的参数。接下来，我们将介绍一些基础函数，包括数学函数、统计函数、概率函数、字符处理函数以及其他一些实用函数。

1. 数学函数

R 中的数学函数与我们通常理解的数学函数概念一致，即通过一定的计算，使得一个集合中的每一个元素在另一个集合中都有唯一确定的元素与之对应。在 R 中，当对一个向量应用数学函数时，返回值也是一个向量，且返回向量中的每个元素值与原向量中的对应元素具有一一对应的关系。R 中常用的数学函数及其含义如表 2-4 所示。

表 2-4　R 常用的数学函数

函数	含义	示例
abs(x)	绝对值	> t <- c(-3, 5) > abs(t) [1] 3 5
sqrt(x)	平方根	> t <- c(2, 9) > sqrt(t) [1] 1.414214 3.000000
ceiling(x)	不小于 x 的最小整数	> t <- c(3.245, 5.978) > ceiling(t) [1] 4 6
floor(x)	不大于 x 的最大整数	> t <- c(3.245, 5.978) > floor(t) [1] 3 5
trunc(x)	向 0 的方向截取的 x 中的整数部分	> t <- c(3.245, 5.978) > trunc(t) [1] 3 5
round(x, digits = n)	将 x 舍入为指定位的小数	> t <- c(3.245, 5.978) > round(t,1) [1] 3.2 6.0
signif(x, digits = n)	将 x 舍入为指定的有效数字位数	> t <- c(3.245, 5.978) > signif(t,3) [1] 3.24 5.98
cos(x) sin(x) tan(x)	余弦 正弦 正切	> t <- c(1.5708, 3.1416) > sin(t) [1] 1.00000e + 00 - 7.34641e - 06
acos(x) asin(x) atan(x)	反余弦 反正弦 反正切	> t <- c(0, 1) > asin(t) [1] 0.000000 1.570796
cosh(x) sinh(x) tanh(x)	双曲余弦 双曲正弦 双曲正切	> t <- c(1.5708, 3.1416) > sinh(t) [1] 2.301308 11.548825
acosh(x) asinh(x) atanh(x)	反双曲余弦 反双曲正弦 反双曲正切	> t <- c(0,1) > asinh(t) [1] 0.0000000 0.8813736

续　表

函数	含义	示例
log(x,base = n) log(x) log10(x)	对 x 取以 n 为底的对数 对 x 取以自然对数 e 为底的对数 对 x 取以 10 为底的对数	> t <- c(8, 10) > log(t, 2) [1] 3.000000 3.321928 > log(t) [1] 2.079442 2.302585 > log10(t) [1] 0.90309 1.00000
exp(x)	指数函数 exp(2.3026)返回值为 10	> t <- c(2, 2.3026) > exp(t) [1] 7.389056 10.000149

2. 统计函数

与数学函数不同,统计函数将向量中的元素作为一个整体,返回反映其特征的统计数。常用统计函数的名称、含义和使用方法见表 2－5。

表 2－5　R 常用的统计函数

函数	含义	示例
mean(x)	平均数	> t <- c(3, 4, 5, 5, 7) > mean(t) [1] 4.8
median(x)	中位数	> t <- c(3, 4, 5, 5, 7) > median(t) [1] 5
sd(x)	标准差	> t <- c(3, 4, 5, 5, 7) > sd(t) [1] 1.48324
var(x)	方差	> t <- c(3, 4, 5, 5, 7) > var(t) [1] 2.2
mad(x)	绝对中位差	> t <- c(3, 4, 5, 5, 7) > mad(t) [1] 1.4826
quantile(x,probs)	求分位数,其中 x 为待求分位数的数值型向量,probs 为一个由[0,1]之间的概率值组成的数值向量	> t <- c(1∶20) > quantile(t, 0.05) 5 % 1.95
range(x)	求值域	> t <- c(3, 4, 5, 5, 7) > range(t) [1] 3 7

续 表

函数	含义	示例
sum(x)	求和	> t <- c(3, 4, 5, 5, 7) > sum(t) [1] 24
diff(x, lag = n)	滞后差分,lag 用以指定滞后几项	> t <- c(3, 4, 5, 5, 7) > diff(t) [1] 1 1 0 2
min(x)	求最小值	> t <- c(3, 4, 5, 5, 7) > min(t) [1] 3
max(x)	求最大值	> t <- c(3, 4, 5, 5, 7) > max(t) [1] 7

3. 概率函数

概率函数通常用来生成特征已知的模拟数据,以及在用户编写的统计函数中计算概率值。在 R 中,概率函数的一般格式为:

```
[dpqr]distribution_abbreviation()
```

其中第一个字母表示该函数所指分布的某一方面,d 表示密度函数,p 表示分布函数,q 表示分位数函数,r 表示生成随机数。distribution_abbreviation 表示需要计算的概率分布的缩写名称,R 中常见的概率分布及缩写如表 2-6 所示。

表 2-6 R 中的概率分布名称及缩写

分布名称	缩写	分布名称	缩写
Beta 分布	beta	Logistic 分布	logis
二项分布	binom	多项分布	multinom
柯西分布	cauchy	负二项分布	nbinom
(非中心)卡方分布	chisq	正态分布	norm
指数分布	exp	泊松分布	pois
F 分布	f	Wilcoxon 符号秩分布	signrank
Gamma 分布	gamma	t 分布	t
几何分布	geom	均匀分布	unif
超几何分布	hyper	Weibull 分布	weibull
对数正态分布	lnorm	Wilcoxon 秩和分布	wilcox

下面我们以正态分布为例,应用一些概率函数计算相关的统计数,并根据正态分布规律生成一组随机数,代码和运行结果如下:

代码 2－17　应用概率函数计算正态分布统计数及生成随机数

```
> pnorm(- 1.96)
[1] 0.0249979
> qnorm(0.025)
[1] - 1.959964
> qnorm(0.9, mean = 10, sd = 10)
[1] 22.81552
> rnorm (20, mean = 10, sd = 10)
  [1]  17.819916      8.082111    18.295216    34.754601    15.959769    11.813063
  [7]  3.243254     - 15.915453   10.119754    13.972396    1.578250     23.492389
  [13] 9.574523      10.989098    21.298849    13.610136    10.753961    17.986536
  [19] 8.178387      19.020259
```

上例中，第一行代码表示计算标准正态分布(均值为 0，标准差为 1)中，随机变量小于或等于－1.96 的概率；第二行代码表示计算标准正态分布中，使得随机变量小于或等于该值的概率为 0.025 的分位点的值；第三行代码表示计算均值为 10，标准差为 10 的正态分布中，使得随机变量小于或等于该值的概率为 0.9 的分位数；第四行代码表示生成 20 个均值为 10，标准差为 10 的正态分布随机数。

4. 字符处理函数

在 R 语言中，字符处理函数是一组用于操作和处理字符串的工具，它们在文本数据的分析和处理中发挥着重要作用。这些函数可以用于执行各种常见的文本操作，如字符串拼接、分割、替换、查找、大小写转换等。这些字符处理函数为用户提供了强大的文本处理能力，使得在 R 中处理文本数据变得简单而高效。一些最常用的字符处理函数见表 2－7。

表 2－7　R 常见的字符处理函数

函数	含义	示例
nchar(x)	计算 x 中的字符数量。	> t <- c("we", "have a", "family") > nchar(t) [1] 2 6 6 > nchar(t[2]) [1] 6
substr(x, start, stop)	提取或替换一个字符向量中的字符串。	[1] "cdef" > t <- "abcdefghigklmn" > substr(t, 3, 6) [1] "cdef"
grep(pattern, x, ignore.case = FALSE, fixed = FALSE)	在 x 中搜索某种模式。若 fixed = FALSE，则 pattern 为一个正则表达式；若 fixed=TRUE，则 pattern 为一个文本字符串。返回值为匹配的下标。	> t <- c("a", "A", "b") > grep("A", t, fixed = TRUE) [1] 2

续 表

函数	含义	示例
sub(pattern, replacement, x, ignore. case = FALSE, fixed=FALSE)	在 x 中搜索 pattern,并以文本 replacement 将其替换。若 fixed = FALSE, 则 pattern 为一个正则表达式;若 fixed = TRUE,则 pattern 为一个文本字符串。	> t <- "abcdefghigklmn" > sub("cdef", "2222", t, fixed = TRUE) [1] "ab2222ghigklmn"
strsplit(x, split, fixed = FALSE)	在 split 处分割字符向量 x 中的元素。若 fixed = FALSE,则 pattern 为一个正则表达式;若 fixed = TRUE,则 pattern 为一个文本字符串。	> t <- "abcd" > strsplit(t, "") [[1]] [1] "a" "b" "c" "d" > strsplit(t, "b") [[1]] [1] "a" "cd"
paste(..., sep ="")	连接字符串,分隔符为 sep。	> paste("x", 1:3, sep ="") [1] "x1" "x2" "x3" > paste("Today is", date()) [1] "Today is Mon Mar 09 20:35:42 2020"
toupper(x)	大写转换。	> t <- "abcedfg" > toupper(t) [1] "ABCEDFG"
tolower(x)	小写转换。	> t <- "ABCDEFG" > tolower(t) [1] "abcdefg"

5. 其他实用函数

表 2-8 中的函数对于数据管理和处理同样非常实用,只是它们无法清楚地划入其他分类中。

表 2-8 其他实用函数

函数	含义	示例
length(x)	对象 x 的长度。	> t <- c(1:5) > length(t) [1] 5
seq(from, to, by)	生成一个序列。	> t <- seq(1, 10, 2) > t [1] 1 3 5 7 9
rep(x, n)	将 x 重复 n 次。	> t <- c(1:3) > rep(t, 2) [1] 1 2 3 1 2 3

续　表

函数	含义	示例
cut(x, n)	将连续型变量 x 分割为有着 n 个水平的因子。使用选项 ordered_result = TRUE 以创建一个有序型因子。	
pretty(x, n)	创建美观的分割点。通过选取 n+1 个等间距的取整值，将一个连续型变量 x 分割为 n 个区间。绘图中常用。	

下面我们通过一个简单的例子展示 cut()和 pretty()函数的使用。

代码 2-18　使用 cut()和 pretty()函数分割序列

```
> t <- seq(1, 7, 0.2)
> cut(t,4)
  [1] (0.994,2.5] (0.994,2.5] (0.994,2.5] (0.994,2.5] (0.994,2.5] (0.994,2.5]
  [7] (0.994,2.5] (0.994,2.5]     (2.5,4]     (2.5,4]     (2.5,4]     (2.5,4]
 [13]    (2.5,4]     (2.5,4]     (2.5,4]     (2.5,4]     (4,5.5]     (4,5.5]
 [19]    (4,5.5]     (4,5.5]     (4,5.5]     (4,5.5]     (4,5.5]  (5.5,7.01]
 [25] (5.5,7.01]  (5.5,7.01]  (5.5,7.01]  (5.5,7.01]  (5.5,7.01]  (5.5,7.01]
 [31] (5.5,7.01]
 Levels: (0.994,2.5] (2.5,4] (4,5.5] (5.5,7.01]
> pretty(t,4)
[1] 0 2 4 6 8
```

在代码 2-18 中，我们首先使用 seq()函数生成了一个从 1 到 7 的等差数列 t，步长为 0.2。接着，cut(t, 4)将这个数列分割成 4 个区间，每个区间对应一个因子水平。结果是一个因子向量，其中每个元素都被分配到一个区间中，这些区间是根据数据的范围自动划分的。而 pretty(t, 4)则生成了一组分割点，将 t 分割为 4 个区间，这些分割点通常是整数或简单的分数，便于阅读和理解。

2.5　编辑数据集

在 R 语言中，编辑数据集是数据分析和处理中的一个重要步骤。R 提供了多种工具和函数，用于对数据集进行各种编辑操作，包括数据的筛选、排序、修改、添加和删除等。这些操作可以帮助用户清理数据、调整数据结构，以及为后续的分析和建模做好准备。

2.5.1　变量的重编码

变量的重编码是指根据一个变量和/或其他变量的现有值创建新值的过程，在 R 语

言中，常用的函数包括 within()、transform()、mutate()、transmute()等。以 within()和 transform()函数为例，其一般格式如下：

```
within(data, {expr1; expr2})
```

data：待修改的原始数据；

expr1、expr2：表达式，表示需要进行的修改。

```
transform(data, expr1, expr2)
```

data：待修改的原始数据；

expr1、expr2：表达式，表示需要进行的修改。

下面我们通过一个简单的例子说明如何使用 within()和 transform()函数为数据框添加变量，我们创建一个包含 2 个变量和 4 条记录的数据框，我们将两个变量的平均数和它们的和通过计算添加为第 3、4 个变量。代码和运行结果如下：

代码 2-19　在数据框中使用 within()和 transform()函数添加变量

```
> mydata <- data.frame(x1 = c(1, 3, 5, 7), x2 = c(8, 8, 9, 9))
> mydata <- within(mydata, {meanx =(x1 + x2) / 2})
> mydata <- transform(mydata, sumx = x1 + x2)
> mydata
  x1 x2 meanx sumx
1  1  8   4.5    9
2  3  8   5.5   11
3  5  9   7.0   14
4  7  9   8.0   16
```

上例中，我们首先创建了一个名为 mydata 的数据框，包含两列 x1 和 x2，分别存储了两组数值。然后用 within()函数计算了 x1 和 x2 两列的平均值，并将结果存储为新列 meanx；用 transform()函数计算了 x1 和 x2 的和，并将结果存储为新列 sumx。最后，打印出修改后的数据框 mydata，可以看到新增的两列 meanx 和 sumx，分别存储了 x1 和 x2 的平均值和和值。

在 R 中修改变量名，可以使用函数 names()，或使用 reshape 包中的函数 rename()，其一般格式如下：

```
names(x)
names(x) <- value
```

x：一个 R 对象，通常是向量、列表或数据框。对于向量和列表，names()返回或设置每个元素的名称；对于数据框，names()返回或设置列名。

value：一个字符向量，其长度最多与 x 的长度相同，或者为 NULL。如果 value 是一个字符向量，它将被用作 x 的名称。如果 value 为 NULL，则会删除 x 的名称属性。

```
rename(.data, ...)
```

.data：一个数据框；

...：对于 rename()，使用 new_name = old_name 来重命名选定的变量。

两个函数的一般用法见代码 2 - 20。

代码 2 - 20　修改变量名

```
> mydata <- data.frame(x1 = c(1, 3, 5, 7), x2 = c(8, 8, 9, 9))
> names(mydata)[1:2]<- c("Name1", "Name2")
> mydata
    Name1 Name2
1     1     8
2     3     8
3     5     9
4     7     9
> library(reshape)
> rename(mydata, c(Name1 = "A", Name2 = "B"))
  A B
1 1 8
2 3 8
3 5 9
4 7 9
```

上例中，首先使用 data.frame()函数创建了一个包含两列的数据框 mydata，这两列分别存储了两组数值。接着，通过 names(mydata)[1:2] <- c("Name1", "Name2")将数据框 mydata 的前两列的列名分别重命名为 Name1 和 Name2。最后，加载 reshape 包并使用 rename()函数将数据框 mydata 的列名 Name1 和 Name2 分别重命名为 A 和 B。

2.5.2　缺失值处理

在 R 语言中，缺失值通常用 NA(Not Available)表示，它是一种特殊的值，用于标记数据中的缺失或不可用部分。而不可能出现的值(如用 0 做除数的结果)用符号 NaN(Not a Number)表示。缺失值在数据分析中非常常见，可能由于数据收集不完整、数据录入错误或数据丢失等原因产生。正确处理缺失值对于确保数据分析的准确性和可靠性至关重要。正确处理缺失值可以避免数据分析中的偏差和错误。例如，在计算均值、方差等统计量时，如果不处理缺失值，可能会得到不准确的结果。此外，在进行机器学习或统计建模时，缺失值的存在可能会影响模型的性能和预测准确性。因此，了解并掌握处理缺失值的方法对于数据科学家和分析师来说是非常重要的技能。

在 R 语言中，is.na()函数用于检测向量或数据框中的缺失值(NA)，并返回一个逻辑向量或逻辑矩阵，其元素值为 TRUE 或 FALSE，分别表示对应位置是否存在缺失值。当数据中包含缺失值时，涉及这些数据的算术运算和函数计算通常也会返回缺失值 NA。为了方便处理缺失值，许多 R 函数包含 na.rm 参数，指定 na.rm＝TRUE 则允许在计算时自动忽略缺失值 NA，仅对非缺失值进行操作。此外，na.omit()函数可以用来删除数据框中包含缺失值 NA 的行。这在需要清理数据以去除不完整的记录时非常有用。例如，

na.omit(mydata)会返回一个删除了所有包含缺失值行的新数据框。

代码 2－21　缺失值判定和处理

```
> x <- c(1, 2, 3, NA)
> is.na(x)
[1] FALSE FALSE FALSE  TRUE
> mean(x)
[1] NA
> mean(x, na.rm = TRUE)
[1] 2
```

在上例中，向量 x 中包含缺失值 NA，通过 is.na()函数可以查看缺失值的具体位置，使用 mean()函数计算向量 x 的算术平均数时，由于缺失值的存在，结果显示为 NA；设置 na.rm＝TRUE 后，则函数会依据变量的其他非缺失值计算结果。

2.5.3　排序

在 R 语言中，对向量、数组或数据框进行排序通常使用 order()函数，其用法一般如下：

```
order(x, na.last = TRUE, decreasing = FALSE)
```

x：待排序的向量，多个向量排序使用逗号隔开；

na.last：逻辑值，TRUE 表示把缺失值 NA 放在最后；

decreasing：逻辑值，TRUE 表示按照降序排列，默认值为 FALSE。

下面以 R 内置数据集 PlantGrowth 为例进行排序，该数据集为植物在三种不同的处理条件下的重量数据，每种处理包括 10 个个体。数据集包含两个变量：数值型变量 weight 和因子型变量 group，在本例中我们首先对 group 变量按照升序排列(按照字母顺序)，随后对 weight 变量进行降序排列(按照数值大小)，代码和运行结果如下。

代码 2－22　对数据集 PlantGrowth 进行排序

```
> attach(PlantGrowth)
> orderdata <- PlantGrowth[order(group, - weight), ]
> orderdata
   weight group
4    6.11  ctrl
2    5.58  ctrl
9    5.33  ctrl
3    5.18  ctrl
7    5.17  ctrl
10   5.14  ctrl
6    4.61  ctrl
```

```
8    4.53  ctrl
5    4.50  ctrl
1    4.17  ctrl
17   6.03  trt1
15   5.87  trt1
18   4.89  trt1
11   4.81  trt1
20   4.69  trt1
13   4.41  trt1
19   4.32  trt1
12   4.17  trt1
16   3.83  trt1
14   3.59  trt1
21   6.31  trt2
28   6.15  trt2
29   5.80  trt2
23   5.54  trt2
24   5.50  trt2
25   5.37  trt2
26   5.29  trt2
30   5.26  trt2
22   5.12  trt2
27   4.92  trt2
```

在上例中，我们通过在 weight 变量名称前添加负号，来实现对该变量的降序排列。在多个变量排序时，主要变量写在前面，次要变量在后，可同时根据多个变量依次进行排序。

2.5.4　数据集的合并

在 R 中拥有相同行数或列数的矩阵或数据框，可以简单地使用函数 cbind()或 rbind()合并。两个函数的参数为要合并的数组或数据框。其中，rbind()函数用于将多个数据框或向量按行合并，要求合并的数据在列数上必须一致，即每个数据框或向量的列数相同。合并后的数据框将包含所有输入数据的行、列名保持一致。而 cbind()函数则用于按列合并数据，要求合并的数据在行数上必须一致，即每个数据框或向量的行数相同。合并后的数据框将包含所有输入数据的列、行名保持一致。这两个函数在数据整理和分析中非常实用，能够方便地将分散的数据整合到一起，为后续的数据处理和分析提供便利。

在下面的例子中，我们创建一个 4 行、2 列的数据框，并分别将数据框与两个向量进行行合并和列合并：

代码 2－23　使用 rbind()和 cbind()函数合并数据

```
> mydata <- data.frame(x1 = c(1, 3, 5, 7), x2 = c(8, 8, 9, 9))
> y1 <- c(11,12)
> mydata <- rbind(mydata, y1)
>mydata
 x1  x2
1  1  8
2  3  8
3  5  9
4  7  9
5  11  12
> x3 <- c("a", "b", "c", "d", "e")
> mydata <- cbind(mydata, x3)
> mydata
  x1  x2  x3
1  1   8   a
2  3   8   b
3  5   9   c
4  7   9   d
5 11  12   e
```

在上例中，首先创建了一个包含两列 x1 和 x2 的数据框 mydata，接着使用 rbind()函数将一个包含两个元素的向量 y1 作为新的一行添加到 mydata 中。由于 y1 的长度为 2，与 mydata 的列数一致，因此合并成功，新行的值分别为 11 和 12。随后，创建了一个字符向量 x3，其长度与 mydata 的行数一致。使用 cbind()函数将 x3 作为新的一列添加到 mydata 中。最终，mydata 变成了一个包含三列（x1、x2 和 x3）的数据框，其中 x3 列包含了字符数据。运行结果是一个扩展后的数据框，包含了原始数据和新添加的行与列。

对于复杂一些的数据集整合，需要使用 merge()函数，merge()函数可以根据一个或多个共同的列将两个数据框合并在一起，形成一个新的数据框。merge()函数的基本用法如下：

```
merge (x, y, by = intersect(names(x), names(y)),
      by.x = by, by.y = by, all = FALSE, all.x = all, all.y = all,
      sort = TRUE, suffixes = c(".x", ".y"), no.dups = TRUE,
      incomparables = NULL, ...)
```

x，y：需要合并的数据框。

by、by.x、by.y：指定合并的列名或列位置。默认情况下，merge()会自动寻找两个数据框中同名的列作为合并键。如果 x 和 y 中合并的列名不同，可以使用 by.x、by.y 这两个参数分别指定。

all、all.x、all.y：逻辑值，TRUE 表示 x 和 y 列向量合并之后输出所有的行。

sort：逻辑值，TRUE 表示按照 by 指定的向量进行排序。

suffixes：一个长度为 2 的字符向量，用于指定当合并的列名在两个数据框中不同时，如何为这些列添加后缀以区分。默认值为 c(".x", ".y")。

no.dups：逻辑值，TRUE 表示不允许重复的列名称。

incomparables：向量，表示 by 中哪些单元不进行合并。

在下面的例子中，我们使用不同的参数对 2 个数据框进行合并，2 个数据框均包含 3 个变量和 5 条记录。第一种情况下，我们希望依据变量 id 进行合并，并保留全部的记录；第二种情况下，我们希望依据 id 和 group 两个变量进行合并，且只保留这两个数据框的交集。代码和运行结果如下。

代码 2－24　使用 merge()函数合并数据框

```
data1 <- data.frame(id = c(1, 2, 3, 4, 5), group = c("a", "a", "b", "b", "b"),
length = c(4.7, 3.6, 2.4, 3.9, 4.5))
data2 <- data.frame(id = c(3, 4, 5, 6, 7), group = c("b", "b", "c", "c", "c"),
width = c(0.6, 0.8, 0.5, 0.5, 0.3))
merge1 <- merge (data1, data2, by ="id", all = TRUE)
> merge1
  id group.x length group.y width
1  1        a      4.7     <NA>    NA
2  2        a      3.6     <NA>    NA
3  3        b      2.4        b   0.6
4  4        b      3.9        b   0.8
5  5        b      4.5        c   0.5
6  6     <NA>      NA         c   0.5
7  7     <NA>      NA         c   0.3
> merge2 <- merge (data1, data2, by = c("id", "group"))
> merge2
  id group length width
1  3       b      2.4   0.6
2  4       b      3.9   0.8
```

在第一步中我们通过设置 by ="id"来选择合并所依据的变量，并通过 all＝TRUE 来保留所有的行。由于在数据框 data1 中不存在 width 变量，而数据框 data2 中不存在 length 变量，因而合并结果中相应的位置显示缺失值 NA。同时，2 个数据框中名称相同的变量 group 在合并后的数据框中分别添加了默认后缀，显示为 group.x 和 group.y。而在第二步操作中，合并方式：by = c("id", "group")表示根据 id 和 group 两列进行合并。由于没有指定 all 参数，只有 id 和 group 在两个数据框中都匹配的行才会被保留。

2.5.5　提取数据集的子集

在数据分析中，我们常常需要从一个很大的数据集中选择有限数量的变量创建新的

数据集，需要根据一定的条件对行或列进行筛选，此时可以使用 subset()函数，其基本用法如下：

```
subset(x, subset, select, drop = FALSE, ...)
```

x：用于提取的数据集，可以是数据框、向量、矩阵。

subset：逻辑表述，表示选择行的条件，可以使用列名或列的位置索引。

select：向量，选取要显示的列向量，如果不指定 select 参数，则默认选择所有列。

drop：逻辑值，默认值为 FALSE，表示结果保持与原对象相同的数据结构，如果设置为 TRUE，则在可能的情况下将结果简化为向量或矩阵。

例如，我们在 R 内置数据集 iris 中筛选品种名称为“versicolor”，且花瓣长度大于 4.8 的个体。提取时需要设置两个条件，即 Petal.Length 变量的值大于 4.8，且 Species 变量的值等于“versicolor”，两个条件之间使用“&”符号连接，代码和运行结果如下：

代码 2－25　使用 subset()函数提取 iris 数据集的子集

```
> subdata <- subset(iris, Petal.Length > 4.8 & Species == "versicolor",
                    select = Sepal.Length: Species)
> subdata
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
53           6.9         3.1          4.9         1.5 versicolor
73           6.3         2.5          4.9         1.5 versicolor
78           6.7         3.0          5.0         1.7 versicolor
84           6.0         2.7          5.1         1.6 versicolor
```

在上例中，Petal.Length > 4.8 & Species == "versicolor"用于指定提取的条件，表示只提取花瓣长度大于 4.8 且品种为 versicolor 的行。select＝Sepal.Length:Species 指定了要选择的列，表示选择从 Sepal.Length 到 Species 的所有列。运行结果是一个新的数据框 subdata，其中只包含满足条件的行和指定的列。从结果可以看出，只有那些花瓣长度大于 4.8 且品种为 versicolor 的样本被提取出来。

2.5.6　随机抽样

在生物统计学的实际应用中，有时需要对变量进行随机抽样。例如建立模型时，抽取一份数据用于构建预测模型，另一份用于验证模型的有效性。在 R 语言中，运用函数 sample()可以从数据集中抽取大小为 n 的一个随机样本，基本用法如下：

```
sample(x, size, replace = FALSE, prob = NULL)
```

x：要从中抽样的向量或数据框。如果 x 是一个向量，则直接从向量中抽样；如果 x 是一个数据框，则从数据框的行中抽样。

size：数值，表示需要抽取的样本数量。如果 size 大于 x 的长度且 replace＝FALSE，则会报错。

replace：逻辑值，表示是否进行有放回抽样。默认值为 FALSE，表示无放回抽样。如

果 replace＝TRUE，则每次抽取后将元素放回，允许同一个元素被多次抽取。

prob：一个与 x 长度相同的数值向量，表示每个元素被抽取的概率。如果未指定，则默认为均匀分布，即每个元素被抽取的概率相同。

在下面的例子中，我们利用 sample()函数在数据集 iris 中采取随机无放回的方式抽取 8 个样本，代码和运行结果如下：

代码 2－26　使用 sample()函数对数据集 iris 随机抽样

```
> sampledata <- iris[sample(1:nrow(iris), 8, replace = FALSE), ]
> sampledata
     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
83            5.8         2.7          3.9         1.2 versicolor
92            6.1         3.0          4.6         1.4 versicolor
142           6.9         3.1          5.1         2.3  virginica
87            6.7         3.1          4.7         1.5 versicolor
61            5.0         2.0          3.5         1.0 versicolor
16            5.7         4.4          1.5         0.4     setosa
2             4.9         3.0          1.4         0.2     setosa
144           6.8         3.2          5.9         2.3  virginica
```

在上例中，nrow(iris)表示数据集 iris 的行数，因此，备选向量为 iris 数据集第一行到最后一行；抽取的样本数量为 8；replace＝FALSE：表示进行无放回抽样，即每个样本只能被抽取一次。sample(1:nrow(iris), 8, replace＝FALSE)这个表达式返回一个随机的行号序列，用于从 iris 数据集中抽取对应的行，进一步使用返回的随机行号序列从 iris 数据集中抽取对应的行，形成一个新的数据框 sampledata。

2.6　R 语言编程简介

在本章前面的内容中，我们介绍了 R 中的各种常用函数。尽管这些函数在许多情况下都非常有用，但在某些特定的数据分析任务中，仅靠单一函数可能无法满足复杂的需求，或者反复调用函数会使代码显得冗长且繁琐。在这种情况下，编写自定义程序就显得尤为必要。R 语言在数据分析领域的编程应用极为广泛，它不仅能够帮助我们高效地处理复杂的数据问题，还能使代码更加简洁和易于维护。在本章中，我们将简要介绍一些核心的编程技巧，包括如何编写自定义函数、掌握基本的流程控制结构，以及如何进行程序调试。

2.6.1　编写自定义函数

在使用 R 语言进行数据处理时，我们经常会遇到需要重复执行的操作。为了提高代码的可读性和可维护性，我们可以通过编写自定义函数来封装这些重复的操作。自定义函数不仅可以使代码更加简洁，还能让我们在需要时轻松地复用代码。在 R 语言中，自

定义函数的基本语法结构如下：

```
myfunction <- function(arg1, arg2,...){statements
                                    return(object)
          }
```

myfunction：自定义函数名称；

arg1，arg2：参数名称；

statements：函数执行的语句；

return(object)：返回结果。

编写好的自定义函数和普通函数的调用方式一致，都是通过函数名(参数)的格式进行调用。在代码 2－27 中，我们通过编写一段自定义函数用以求取两样本平均数比较 t 检验中的 t 值，其计算公式为：

$$t=\frac{(\bar{y}_1-\bar{y}_2)-(\mu_1-\mu_2)}{s_{\bar{y}1-\bar{y}2}}$$，其中 $s_{\bar{y}1-\bar{y}2}=\sqrt{\bar{s}^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$。

代码 2－27　通过自定义函数计算两样本平均数比较的 t 值

```
> n = 15
> a = rnorm(n); a
  [1]  - 0.84540493    0.29156637   - 1.26964297   - 0.79225151   - 0.84748603   - 0.88469883
  [7]  - 1.72883583  - 0.31979692   - 0.53901396    0.53728268     1.38120225     0.04084851
  [13] - 1.44349041    0.73076999   - 0.13182878
> b =rexp(n); b
  [1]  0.3787410  0.5837818  0.3182901  0.2598803  0.3460923  0.3219301  0.3472058
  [8]  2.2087519  0.2507193  0.2767760  2.2946466  2.1676922  0.1095069  0.4857460
[15] 1.4154224
> t1 <- function(x,y){
  s =((n - 1) *var(x)+(n - 1) *var(y))/(2 *n - 2)
    t =(mean(x)- mean(y))/(s *sqrt(2 / n))
    return(t)
}
> t1(a,b)
[1] - 4.6105
```

在上例中，我们首先利用正态分布函数与指数分布函数分别随机生成两个长度为 15 的向量 a 和 b，并利用自己定义的函数计算两个样本的 t 值，结果显示两个样本比较的 t 值为－4.6105。统计数的具体含义将在本书后面的章节中进行介绍。

对于编写好的自定义函数，可以通过以下函数来查看函数的结构以及参数设置(注：只适用于自定义函数，对 R 自带函数不适用)：

body()：查看函数的内部代码。

formals()：查看函数调用的参数列表。

environment()：查看函数的变量所在环境。

force()：主要是在参数缺失或者函数未传入参数时使得函数进行报错，从而提醒调用者。

invisible(x)：查看函数运行过程中不可见的变量的值。

2.6.2　流程控制

在 R 语言中，程序的执行通常是从上到下依次进行的，每条语句只执行一次。然而，通过使用流程控制结构，我们可以实现更复杂的逻辑，例如在满足特定条件时执行循环或进行选择性执行。这些流程控制结构通常由语句、条件、表达式和序列等组成。具体来说：

语句(Statement)：可以是一条单独的 R 命令，或者是由多条命令组成的复合语句，后者通常用花括号{}括起来，语句之间用分号“;”分隔。

条件(Condition)：是一个逻辑表达式，其结果为 TRUE 或 FALSE。

表达式(Expression)：是用于计算数值或字符串的语句。

序列(Sequence)：是一组有序的数值或字符串。

R 语言中的流程控制主要分为两大类，第一类是循环结构，即当满足特定条件时，会重复执行某段代码。常见的循环结构包括 for 循环和 while 循环。第二类是条件结构：只有在满足特定条件时，才会执行某条或某组语句。常见的条件结构有 if-else、ifelse 和 switch 语句。

(1) for 结构

for 循环会重复执行一段代码，直到循环变量的值遍历完指定的序列为止。其基本语法为：

```
for(i in sequence) {statement}
```

i：循环变量，用于存储序列中的当前值；

sequence：一个序列，i 会在每次循环中依次取该序列中的值；

statement：一个或一组语句，会在每次循环中针对序列中的当前值 i 执行，当序列中的所有元素都被遍历后，循环结束。

(2) while 结构

while 结构重复地执行一个语句，直到指定的条件不再为真为止，其基本语法为：

```
while(condition) {statement}
```

condition：一个逻辑条件，用于判断是否继续执行循环；

statement：当条件为真时执行的语句。

在循环过程中，若要输出每次循环的结果，可使用函数 cat()或 print()，cat()函数的基本格式为：

```
cat(expr1, expr2, ...)
```

expr1、expr2：可以是字符串或表达式，表示要输出的内容。如果是表达式，则输出其计算结果。

下面我们分别使用 for 和 while 两种循环结构，求取 1—100 内所有自然数的和，代码及运行结果如下：

代码 2-28　使用 for 和 while 结构对 1—100 内所有自然数求和

```
> sum = 0
> for(i in 1 : 100){sum = sum + i}
> print(sum)
[1] 5050
> sum = 0; i = 1;
> while (i <= 100) {sum = sum + i; i = i + 1}
> print(sum)
[1] 5050
```

需要注意的是，在使用 for 循环时，一旦循环变量遍历完序列中的所有值，循环会自动停止。然而，在使用 while 循环时，必须确保循环条件最终会变为 FALSE，否则循环将无限进行下去。在前面的例子中，我们通过设置计数变量 i 并在每次循环时将其值加 1，以此来控制循环的次数。当 i 的值达到 101 时，循环条件不再满足，循环随即终止。

(3) if-else 结构

if-else 结构允许在某个给定条件为真时执行一组语句，而在条件为假时执行另一组语句。其基本语法为：

```
if (condition) {statement1} else {statement2}
```

condition：一个逻辑条件，用于判断真假；

statement1：当 condition 为 TRUE 时执行的语句；

statement2：当 condition 为 FALSE 时执行的语句。

(4) ifelse 结构

ifelse 是 if-else 结构的一个紧凑且向量化的版本，其基本语法如下：

```
ifelse(condition, statement1, statment2)
```

condition：一个逻辑条件，可以是一个向量。对于向量中的每个元素，如果条件为 TRUE，则返回 statement1 的结果；如果条件为 FALSE，则返回 statement2 的结果。

statement1：当条件为 TRUE 时返回的值。

statement2：当条件为 FALSE 时返回的值。

ifelse 函数特别适合在需要对整个向量进行条件判断时使用，而 if-else 结构则更适合用于单个条件判断的场景。下面我们通过一个简单的例子，利用 ifelse 结构判断向量中元素的正负，并输出其绝对值。

代码 2－29　使用 ifelse 结构计算绝对值

```
> x <- rnorm(10)
> x
[1] -1.59043804 -0.06856828 0.33171178 -1.27610565 0.07581569 -0.54443898 -0.48063221
-0.70625277 -1.45662212 -0.42339751
> ifelse(x > 0, x, -x)
[1] 1.59043804 0.06856828 0.33171178 1.27610565 0.07581569 0.54443898 0.48063221
0.70625277 1.45662212 0.42339751
```

上例中，我们通过正态分布函数 rnorm()生成一个长度为 10 的向量，其值有正有负，使用 ifelse()函数对每个元素进行判断，若值为负则输出其相反数。

(5) switch 结构

switch 结构根据一个表达式的值选择列表中的返回值，其基本语法如下：

```
switch(expression, list)
```

expression：表达式，其值可以是一个整数值或一个字符串。

list：包含返回值的列表，根据 expression 的值来决定输出列表中的哪一个元素。

switch 函数的行为取决于 expression 的值。如果 expression 的结果是一个整数，且值在 1 到 length(list)之间，switch 函数将返回列表中相应位置的元素；如果 expression 的结果是一个字符串，switch 函数将返回列表中以该字符串为名称的元素对应的值；如果 expression 的结果超出了列表的范围，switch 函数将返回 NULL；如果 expression 的结果与列表中的任何元素都不匹配，但列表包含一个未命名的元素，则 switch 函数将返回该未命名元素的值。

通过 switch 结构，可以在一个自定义函数中根据不同的参数执行不同的功能。下面通过一个例子来展示 switch 结构的使用方法。

代码 2－30　switch 结构的使用

```
> myfunction <- function(x,type){
   switch(type,
            mean = mean(x),
            median = median(x),
            sum = sum(x),
            max = max(x),
            min = min(x),
            sqrt = sqrt(x))
  }
> x <- c(18,25,7,19,31,24,10
> myfunction(x,"mean")
[1] 19.14286
> myfunction(x,"median")
```

```
[1] 19
> myfunction(x,"sum")
[1] 134
> myfunction(x,"max")
[1] 31
> myfunction(x,"min")
[1] 7
> myfunction(x,"sqrt")
[1] 4.242641 5.000000 2.645751 4.358899 5.567764 4.898979 3.162278
```

在上例中，通过 switch 结构，myfunction 函数能够根据传入的 type 参数选择执行不同的统计操作。这种设计使得函数具有很高的灵活性，能够根据用户的需求动态调整其行为。在上述示例中，myfunction 成功地计算了均值、中位数、总和、最大值、最小值和平方根，展示了 switch 结构的强大功能。

2.6.3 程序的注释和调试

注释是编程中不可或缺的一部分，它能够帮助开发者更好地理解代码的功能和逻辑。为编写的程序添加注释，可以很大程度地提高代码的可读性。在 R 语言中，注释的符号为 #，# 符号后面的字符会被编译器自动忽略，不会对代码的执行产生任何影响。注释的内容可以根据个人的使用习惯和需求进行设定。若有多行注释内容，需要在每一行前分别添加 # 符号。

例如，我们在编写了代码 2-30 中的程序后，可以这样添加注释，便于理解程序的内容。

```
> myfunction <- function(x,type){   #定义函数,参数 1 为向量,参数 2 为计算类型
+   switch(type,                     #根据参数 2 的值选择操作
+          mean = mean(x),           #求算术平均数
+          median = median(x),       #求中位数
+          sum = sum(x),             #求和
+          max = max(x),             #最大值
+          min = min(x),             #最小值
+          sqrt = sqrt(x))           #平方根
+ }                                  #注意 x 必须为数值型向量
```

在编写程序的过程中，难免会出现错误。通过在程序中加入调试语句，可以有效地帮助我们发现问题。最常见的调试方法是对程序中的重要变量进行赋值和输出。例如，可以在循环或条件分支代码中加入显示函数，在运行过程中即时输出变量的值以供查看。在确认程序运行正常后，可以将这行代码进行注释。此外，在编写函数的过程中，可以使用 browser()命令对函数进行调试。将 browser()命令插入到需要检测的行，在函数执行到这里时会暂停，并显示一个提示符。此时可以在提示符后输入交互式命令检查函数运行情况，如查看中间变量的赋值是否正确。

调试过程中常用的命令包括：

输入 n:逐行运行程序,并提示下一行将运行的语句;

输入 c:跳到下一个中断点;

输入 Q:退出调试模式。

除了 browser(),R 语言还提供了其他一些调试工具,例如 trace()、setBreakpoint()、traceback()和 recover()等,在本章中不做详细介绍。

习 题

2-1　R 中向量、矩阵、数组和数据框分别有什么特点?

2-2　用函数 rep()构造一个向量 x,它由 3 个 3,4 个 2,5 个 1 构成。

2-3　使用 R 的循环结构输出 1 至 100 之间的能够被 3 或 5 整除的数,并求和。

2-4　利用 R 自带数据集中的 iris 数据集,完成以下操作:

(1) 为数据集添加一个变量,其名称为"ID",值为从 1 到数据集行数的自然数;

(2) 提取数据集中 Species 值为 setosa、Sepal.Length 大于 5 的子集;

(3) 对上述子集按照 Petal.Length 值从小到大排序;

(4) 提取数据集中 Species 值为 setosa、Sepal.Width 大于 3 的子集;

(5) 根据变量 ID 合并上述两个子集,并保留所有的行。

2-5　对某班级同学的成绩进行统计,得下表

学生成绩统计表

学号	数学成绩	语文成绩	英语成绩
1802020101	93	71	64
1802020102	77	94	83
1802020103	91	88	65
1802020104	88	84	96
1802020105	92	79	77
1802020106	97	79	64
1802020107	92	90	96
1802020108	60	82	65
1802020109	76	80	83
1802020110	60	87	66

(1) 使用 R 自带函数 sum()和 mean()为数据添加“总成绩”和“平均成绩”两个变量;

(2) 使用条件结构对学生成绩评定等次:三门课成绩均大于 80 分为优秀,三门课成绩均大于 60 分但至少有一门小于 80 分为合格,有一门小于 60 分则为不合格,并将判断情况作为“等次”变量添加在数据中。

第3章

描述性统计

描述性统计(descriptive statistics),是指运用图表和数学方法,对统计数据进行整理、分析,描述分布状态、数字特征和随机变量之间关系的方法。描述性统计主要包括数据的频数分布分析、集中趋势分析、离散程度分析、分布分析以及绘制一些基本的统计图形。

数据的频数分布分析指的是在分组的基础上,把样本的所有观察值按组归并排列,形成总体中各个单位在各组间的分布。数据的集中趋势分析,主要是计算反映变数集中趋势的统计数,包括算术平均数、几何平均数、调和平均数、中位数和众数等。数据的离散程度分析,主要是计算反映变数离散特性的统计数,包括标准差、方差、变异系数和分位数等。数据的分布分析,主要是求取变数分布特性的统计数,包括峰度系数和偏度系数。基本统计图形,主要是通过绘制直方图、箱线图、散点图等图形,直观展示数据的分布和特征。描述性统计为我们提供了一种系统的方法,用于总结和描述数据集的主要特征,是数据分析的基础步骤。

3.1 利用函数求解描述性统计数

3.1.1 利用R的基础函数

在描述性统计的计算方面,R提供了多种选择,其中最直接的方式是使用R的基本函数来计算描述性统计量。这些基本函数包括:mean()、sd()、var()、min()、max()、median()、sum()、range()、quantile(),分别用于计算算术平均数、标准差、方差、最小值、最大值、中位数、求和、取值范围和分位数。需要注意的是,R基础安装的函数中没有计算偏度系数和峰度系数的函数。

例3-1 考查106个“岱字棉”原种单株的纤维长度(单位:毫米),得结果如表3-1,试利用R的基础函数对该数据求取描述性统计数。

表 3－1　106 个“岱字棉”原种单株的纤维长度

27.25	27.64	27.82	27.92	28.04	28.22	28.22	28.37	28.44	28.46
28.55	28.57	28.61	28.64	28.68	28.69	28.73	28.79	28.82	28.89
28.91	28.94	28.96	29.06	29.06	29.15	29.21	29.24	29.24	29.26
29.29	29.32	29.33	29.33	29.38	29.39	29.41	29.43	29.45	29.47
29.48	29.53	29.58	29.59	29.66	29.67	29.67	29.69	29.72	29.74
29.86	29.86	29.88	29.89	29.91	29.94	29.97	29.97	29.99	29.99
30.00	30.08	30.12	30.14	30.16	30.19	30.22	30.25	30.27	30.27
30.33	30.38	30.41	30.45	30.47	30.47	30.48	30.52	30.52	30.57
30.58	30.61	30.62	30.66	30.74	30.75	30.75	30.78	30.85	30.89
30.92	30.96	30.97	31.03	31.15	31.16	31.32	31.36	31.44	31.50
31.58	31.69	31.71	31.92	32.24	32.38				

观察例 3－1 中的数据，可以发现其包含一个变量（纤维长度）和 106 个观察值，对于这类数据，可以将所有观察值录入在 Excel 表格的一列中。根据题意，以“length”作为变量名，将其保存为 Example3_1.csv（如图 3－1）。

	A		A		A		A		A		A
1	length	19	30.08	37	29.06	55	31.15	73	29.97	91	29.45
2	27.25	20	30.38	38	29.33	56	32.24	74	30.22	92	29.72
3	28.55	21	30.61	39	29.59	57	28.22	75	30.48	93	29.99
4	28.91	22	30.96	40	29.89	58	28.69	76	30.75	94	30.27
5	29.29	23	31.69	41	30.14	59	29.15	77	31.32	95	30.52
6	29.48	24	27.82	42	30.45	60	29.39	78	28.37	96	30.85
7	29.86	25	28.61	43	30.66	61	29.67	79	28.79	97	31.44
8	30	26	28.96	44	31.03	62	29.94	80	29.24	98	28.46
9	30.33	27	29.33	45	31.92	63	30.19	81	29.43	99	28.89
10	30.58	28	29.58	46	28.04	64	30.47	82	29.69	100	29.26
11	30.92	29	29.88	47	28.68	65	30.75	83	29.97	101	29.47
12	31.58	30	30.12	48	29.06	66	31.16	84	30.25	102	29.74
13	27.64	31	30.41	49	29.38	67	32.38	85	30.52	103	29.99
14	28.57	32	30.62	50	29.66	68	28.22	86	30.78	104	30.27
15	28.94	33	30.97	51	29.91	69	28.73	87	31.36	105	30.57
16	29.32	34	31.71	52	30.16	70	29.21	88	28.44	106	30.89
17	29.53	35	27.92	53	30.47	71	29.41	89	28.82	107	31.5
18	29.86	36	28.64	54	30.74	72	29.67	90	29.24	108	

图 3－1　例 3－1 数据录入格式

读入文件并计算各描述性统计数，代码和运行结果如下：

代码 3－1　读取例 3－1 的数据并利用基础函数求取描述性统计数

```
> example3_1 <- read.table("E:/RinBio/Chapt3/Example3_1.csv",sep =",",header = TRUE)
> mean(example3_1$length)
[1] 29.855
> sd(example3_1$length)
```

```
[1] 1.038006
> var(example3_1$length)
[1] 1.077456
> min(example3_1$length)
[1] 27.25
> max(example3_1$length)
[1] 32.38
> median(example3_1$length)
[1] 29.885
> sum(example3_1$length)
[1] 3164.63
> range(example3_1$length)
[1] 27.25 32.38
> quantile(example3_1$length)
    0%       25%      50%       75%      100%
27.2500   29.2175   29.8850     30.5575   32.3800
```

这段代码通过 R 语言的基本函数对数据集 example3_1 中的 length 变量进行了描述性统计分析。结果显示，“岱字棉”的纤维长度（单位：毫米）的平均数为 29.855，标准差为 1.038，方差为 1.077。最小值为 27.25，最大值为 32.38，中位数为 29.885。总和为3164.63，取值范围从 27.25 到 32.38。分位数分析显示，第一四分位数为 29.2175，第三四分位数为 30.5575。这些统计量为数据的分布特征提供了基本的描述。

3.1.2 利用 summary()函数来获取描述性统计数

在 R 语言中，summary()函数是一个非常实用的工具，用于快速获取数据集的描述性统计信息。对于数值型变量，summary()函数能够返回其最小值、最大值、平均值、中位数、第一四分位数和第三四分位数等统计量，而对于因子型变量，则返回各水平的频数。通过调用 summary()函数，用户可以迅速了解数据的基本特征，为后续的数据分析和建模提供重要的参考。在这里我们使用 summary()函数对例 3－1 的数据求取描述性统计数，代码和运行结果如下：

代码 3－2　利用 summary()函数求取例 3－1 数据的描述性统计数

```
> example3_1 <- read.table("E:/RinBio/Chapt3/Example3_1.csv", header = TRUE, sep =",")
> summary(example3_1$length)
   Min. 1st Qu.   Median   Mean   3rd Qu.   Max.
  27.25   29.22   29.89   29.86   30.56   32.38
```

在上述代码中，使用 read.table()函数从指定路径读取数据文件 Example3_1.csv，header＝TRUE 表示文件的第一行包含列名，sep =","表示数据字段之间以逗号分隔。本例中使用的数据为数值型向量，summary()函数的返回结果包括了最小值、最大值、平均数和四分位数，计算结果与代码 3－1 中通过 quantile()和 mean()函数求取的结果相同。

3.2　描述性统计数的相关软件包

3.2.1　Hmisc 包

Hmisc 是一个在 R 语言中广泛应用于数据分析和统计建模的程序包。它提供了丰富的函数和工具，用于数据处理、描述性统计、缺失值处理、变量转换、相关性分析等。Hmisc 包中的 describe()函数通过计算并返回数据集中每个变量的统计摘要，帮助用户快速了解数据的基本特征和分布形态。这些统计摘要包括变量的类型、非缺失值数量、唯一值数量、最小值、最大值、中位数、均值、标准差、四分位数等。describe()函数的语法结构为：

```
describe(x, ..., type = NULL, labels = FALSE, quantiles = c(0.25, 0.5, 0.75),
              na.rm = FALSE, use = "everything", full = TRUE, trim = 0,
              digits = max(3, getOption("digits")), format.args = list())

describe (x, descript, exclude.missing = TRUE, digits = 4,
         listunique = 0, listnchar = 12,
         weights = NULL, normwt = FALSE, minlength = NULL, ...)
```

x：要进行描述性统计分析的对象，通常是一个数据框(data frame)或向量(vector)。

...：其他参数，这些参数可以传递给函数内部的其他函数，用于控制输出的详细程度和格式。

type：指定要描述的变量类型。例如，可以设置为"numeric"只描述数值型变量，或"factor"只描述因子型变量。如果省略，则描述所有类型的变量。

labels：如果为 TRUE，则显示变量的标签(如果有的话)。这些标签通常是变量名或用户自定义的标签。

quantiles：指定要计算的分位数。默认为 c(0.25, 0.5, 0.75)，即第一四分位数、中位数和第三四分位数。也可以设置为其他自定义的分位数。

na.rm：如果为 TRUE，则在计算统计量时忽略 NA 值。

use：指定在计算统计量时如何处理 NA 值。可选值包括"everything"(使用所有数据，包括 NA 值，但 NA 值不会影响统计量的计算，前提是 na.rm = TRUE)、"all.obs"(使用所有数据，包括 NA 值，NA 值会影响统计量的计算)、"complete.obs"(仅使用没有 NA 值的观测)等。

full：如果为 TRUE，则提供完整的描述性统计信息。如果为 FALSE，则可能只提供部分统计信息。

trim：用于计算修剪均值(trimmed mean)时去除的尾部比例。默认为 0，即不计算修剪均值。如果设置为非零值，则计算去除一定比例最小和最大观测值后的均值。

digits：指定输出中数字的有效位数。默认为 max(3, getOption("digits"))，即至少为 3 位数字，或者根据全局设置确定。

format.args：一个列表，用于控制输出数字的格式。例如，可以设置小数点的位数、千分位分隔符等。

descript：可选参数，用于为描述性统计结果提供一个标题或描述性文字。

exclude.missing：是否排除缺失值（NA）在统计计算中。如果设置为 TRUE，则在计算统计量时会忽略缺失值。

listunique：指定在输出中列出的唯一值的数量。如果设置为 0，则不会列出任何唯一值；如果设置为一个正整数 n，则会列出每个变量的前 n 个唯一值。

listnchar：指定在输出中列出的字符变量的最大字符数。如果某个字符变量的长度超过这个值则会被截断。

weights：用于加权统计的权重向量。如果设置为 NULL，则不使用加权统计。`weights = c(1, 2, 3)`，表示每个观测值的权重分别为 1、2、3。

normwt：是否将权重向量归一化，使其总和为 1。

minlength：用于字符变量的最小长度。如果设置为一个正整数 n，则字符变量的长度会被限制为至少 n 个字符。

例 3-2 许多害虫的发生都和气象条件有一定的关系。山东临沂测定 1964—1973 年（共 10 年）间 7 月下旬的温雨系数（雨量 mm/平均温度℃，x）和大豆第二代造桥虫发生量（每百株大豆上的虫数，y）的关系如表 3-2，试求取 x 和 y 变量的描述性统计数。

表 3-2 温雨系数和造桥虫虫口密度

温雨系数（x）	虫口密度（y）	温雨系数（x）	虫口密度（y）
1.58	180	2.41	175
9.98	28	11.01	40
9.42	25	1.85	160
1.25	117	6.04	120
0.30	165	5.92	80

例 3-2 的数据中包含 2 个变量，每个变量具有 10 个观察值。对于这种类型的数据，录入时每个变量的数据位于一列，每个观测值位于一行，变量名称 x 和 y 录入第一行，并且需要注意两个变量的观测值之间的一一对应关系。录入的数据保存为 Example3_2.csv（如图 3-2）。

	A	B
1	x	y
2	1.58	180
3	9.98	28
4	9.42	25
5	1.25	117
6	0.3	165
7	2.41	175
8	11.01	40
9	1.85	160
10	6.04	120
11	5.92	80

图 3-2 例 3-2 数据录入格式

使用 read.table() 函数读入文件后，利用 Hmisc 包中的 describe() 函数计算描述性统计数，代码和运行结果如下：

代码 3－3　通过 Hmisc 包中的 describe()函数计算例 3－2 数据的描述性统计数

```
> library(Hmisc)
>   example3_2 <- read.table("E:/RinBio/Chapt3/Example3_2.csv", header = TRUE, sep =",")
> describe(example3_2)
example3_2

 2  Variables      10  Observations
---------------------------------------------------------------
x
      n      missing   distinct  Info  Mean  pMedian  Gmd      .05      .10      .25
      10     0         10        1     4.976 5.335    4.728    0.7275   1.1550   1.6475
       .50    .75       .90       .95
      4.1650 8.5750    10.0830   10.5465

Value       0.30  1.25  1.58  1.85  2.41  5.92  6.04  9.42  9.98  11.01
Frequency   1     1     1     1     1     1     1     1     1     1
Proportion  0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1

For the frequency table, variable is rounded to the nearest 0
---------------------------------------------------------------
y
      n      missing  distinct    Info    Mean  pMedian  Gmd     .05    .10    .25
      10     0        10          1       109   104      73.16   26.35  27.70  50.00
       .50    .75      .90        .95
     118.50 163.75   175.50     177.75

Value       25  28  40  80  117  120  160  165  175  180
Frequency   1   1   1   1   1    1    1    1    1    1
Proportion  0.1 0.1 0.1 0.1 0.1  0.1  0.1  0.1  0.1  0.1

For the frequency table, variable is rounded to the nearest 0
---------------------------------------------------------------
```

运行结果显示，数据集 example3_2 包含 2 个变量（x 和 y）和 10 个观测值，并分别显示了温雨系数（x）和虫口密度（y）两个变量的观测数（n）、缺失值数目（missing）、唯一值数目（distinct）、信息量（Info，1 表示变量具有完全的信息量）、平均值（Mean）、中位数（pMedian）、基尼均差（Gmd）、分位数、五个最大观测值和五个最小观测值，以及每个观察值的频数和频率。这些统计信息为用户提供了数据的基本特征和分布形态的全面了解，有助于进一步的数据分析和建模。

3.2.2 pastecs 包

pastecs 是 R 语言中一个非常实用的包，主要用于计算和展示数据的基本统计描述。它能够对各种数据结构（如向量、数据框等）进行处理。在计算描述性统计数方面，除了常见的均值、中位数、众数外，还可以计算分位数、极差、变异系数等。在输出格式上，它会以一种整齐有序的方式呈现统计结果，方便用户查看和比较不同变量的统计信息。在数据探索性分析场景下使用频率较高。研究人员可以用它来初步了解数据的整体情况，从而为数据清洗、建模等后续步骤提供有力的依据。

stat.desc()函数是 pastecs 包中的一个核心函数，用于计算数据框（data frame）或时间序列（time series）中各个字段（列）的描述性统计信息。该函数提供了全面的描述性统计数，包括样本数、缺失值数、最小值、最大值、均值、中位数、数据范围（极差）、总和、标准误差、置信区间、方差、标准差、变异系数、偏度系数、峰度系数等以及正态性检验的结果（当设置相应参数时）。其一般使用格式为：

```
stat.desc(x, basic = TRUE, desc = TRUE, norm = FALSE, p = 0.95)
```

x：数据框或时间序列。

basic：逻辑值，TRUE 表示输出结果将包含数据的基本信息，如样本容量、空值数目、缺失值数目、最小值、最大值、值域及总和。

desc：逻辑值，TRUE 表示输出结果将包含描述性统计量，如中位数、平均数、平均数的标准误、置信区间、方差、标准差以及变异系数。

norm：逻辑值，TRUE 表示计算正态分布统计量，包括偏度系数和峰度系数、它们的标准误差以及 Shapiro-Wilk 正态检验结果。

p＝0.95：输出结果中置信区间的置信度为 95%。

使用 pastecs 包中的 stat.desc()函数计算例 3－2 中数据的描述性统计数，代码和运行结果如下：

代码 3－4　通过 pastecs 包中的 stat.desc()函数计算例 3－2 数据的描述性统计数

```
> library(pastecs)
>   example3_2 <- read.table("E:/RinBio/Chapt3/Example3_2.csv", header = TRUE, sep =",")
> stat.desc(example3_2,norm = TRUE)
                     x           y
nbr.val     10.0000000  10.0000000
nbr.null     0.0000000   0.0000000
nbr.na       0.0000000   0.0000000
min          0.3000000  25.0000000
max         11.0100000 180.0000000
range       10.7100000 155.0000000
sum         49.7600000 1090.0000000
median       4.1650000  118.5000000
```

```
mean           4.9760000          109.0000000
SE.mean        1.2774047           19.5896798
CI.mean.0.95     2.8896901           44.3149345
var           16.3176267         3837.5555556
std.dev        4.0395082           61.9480069
coef.var       0.8117983            0.5683303
skewness       0.2946624           - 0.2193784
skew.2SE       0.2144425           - 0.1596541
kurtosis       - 1.7653653         - 1.7966651
kurt.2SE       - 0.6615578         - 0.6732872
normtest.W     0.8775107            0.8768165
normtest.p     0.1221640            0.1199319
```

运行结果依次输出了温雨系数(x)和虫口密度(y)的样本容量(nbr.val)、缺失值数(nbr.null 和 nbr.na)、最小值(min)、最大值(max)、值域(range,即最大值与最小值之差)、总和(sum)、中位数(median)、平均数(mean)、平均数标准误(SE.mean)、平均数的 95%置信区间(CI.mean.0.95)、方差(var)、标准差(std.dev)、变异系数(coef.var)、偏度系数(skewness)、偏度系数标准误差(skew.2SE)、峰度系数(kurtosis)、峰度系数标准误差(kurt.2SE)、Shapiro-Wilk 正态性检验的 W 统计量和 p 值(normtest.W 和 normtest.p)。

3.2.3　psych 包

psych 包是 R 语言中的一个专门用于心理研究和数据分析的包,它提供了多种数据分析工具,包括数据可视化、因子分析、可靠性分析、相关性分析、主成分分析、聚类分析等。这些功能使得 psych 包在心理学、社会科学以及其他需要复杂数据分析的领域中具有广泛的应用。

psych 包中的 describe()函数可以计算数据集的基本描述性统计数,如均值、标准差、中位数、四分位数、最小值、最大值、极差、偏度系数和峰度系数等。这些统计数有助于了解数据的集中趋势、离散程度、分布形态等特征。其一般使用格式为:

```
describe(x, na .rm = TRUE, interp = FALSE, skew = TRUE, ranges = TRUE, trim =.1,
                type = 3, check = TRUE, fast = NULL, quant = NULL, IQR = FALSE, omit = FALSE)
describeData(x, head = 4, tail = 4)
describeFast(x)
```

x:要进行描述性统计分析的对象,通常是一个数据框(data frame)或向量(vector)。

na.rm:逻辑值,表示是否在计算统计量时忽略 NA 值。默认值为 TRUE,表示忽略 NA 值。

interp:逻辑值,表示是否使用插值法计算分位数。默认值为 FALSE。

skew:逻辑值,表示是否计算偏度(skewness)。默认值为 TRUE,表示计算偏度。

ranges:逻辑值,表示是否计算极差(range)。默认值为 TRUE,表示计算范围。

trim:数值,表示计算截尾均值(trimmed mean)时去除的尾部比例。默认值为 0.1,表

示去除 10%的最小和最大观测值。

type:数值,指定分位数的计算方法。默认值为 3,表示使用默认的分位数计算方法。

check:逻辑值,表示是否检查数据中的异常值。默认值为 TRUE。

fast:逻辑值,TRUE 表示快速计算模式,只计算观测值数目、平均数、标准差、最小值、最大值和极差,FALSE 表示计算全部的描述统计数,可能运行速度较慢,NULL 表示在数据量较大时自动切换到快速模式。

quant:向量,指定要计算的分位数。默认值为 NULL,表示使用默认的分位数(0.25, 0.5, 0.75)。

IQR:逻辑值,表示是否计算四分位距(Interquartile Range, IQR)。默认值为 FALSE。

omit:逻辑值,TRUE 表示忽略非数值型变量。默认值为 FALSE。

head:数值,表示要显示数据头部的行数。默认值为 4,表示显示数据的前 4 行。

tail:数值,表示要显示数据尾部的行数。默认值为 4,表示显示数据的后 4 行。

使用 psych 包的 describe()函数计算例 3-2 中的描述性统计数,代码和运行结果如下:

代码 3-5　通过 psych 包的 describe()函数计算例 3-2 数据的描述性统计数

```
> library(psych)
>   example3_2 <- read.table("E:/RinBio/Chapt3/Example3_2.csv", header = TRUE, sep =",")
> describe(example3_2)
  vars  n  mean   sd median trimmed  mad  min  max  range   skew  kurtosis  se
x    1  10  4.98  4.04   4.16   4.81  4.08  0.3  11.01  10.71  0.29  -1.77  1.28
y    2  10  109.00  61.95  118.50  110.62  76.35  25.0  180.00  155.00  -0.22  -1.80  19.59
```

运行代码 3-5 后,依次输出了温雨系数(x)和虫口密度(y)的变量编号(vars)、样本容量(n)、平均数(mean)、标准差(sd)、中位数(median)、截尾均值(trimmed)、绝对中位差(mad)、最小值(min)、最大值(max)、极差(range)、偏度系数(skew)、峰度系数(kurtosis)以及平均数标准误(se)。

3.3　分组描述性统计

在前面的分析中,我们将每个变量的全部观察值作为一个整体来进行统计。然而有些情况下,数据可根据某一个或某几个变量的值不同而划分为不同的组,并且需要对每组数据分别进行描述性统计。在 R 中,可以使用基础安装包中的 aggregate()函数或其他软件包来完成分组描述性统计的任务。

3.3.1　aggregate()函数

aggregate()函数可以将数据按单个或多个变量进行分组,然后对每一组数据分别进

行函数统计，以表格形式输出统计结果，其一般使用格式为：

```
aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)
```

x：要进行描述性统计分析的对象，通常是一个数据框(data frame)或列表(list)，包含要进行聚合的变量。

by：一个列表，列表中的每个元素都是一个因子(factor)或用于分组的变量，这些变量决定了数据如何被分组。

FUN：应用于每个分组的函数，如 sum()、mean()、max()、min()等，或者用户自定义的函数。

...：其他参数，这些参数会被传递给 FUN 函数。

simplify：逻辑值，如果为 TRUE(默认值)，则聚合结果会尽可能地被简化为数组或矩阵；如果为 FALSE，则结果总是以数据框的形式返回。

drop：逻辑值，如果为 TRUE(默认值)，当分组后的结果只有一列(除了分组变量)时，结果会被简化为向量或矩阵；如果为 FALSE，则结果总是以数据框的形式返回。

例 3－3　R 内置数据集 iris 中包含 150 个鸢尾花样本，每个样本有 4 个特征：Sepal.Length(花萼长度)、Sepal.Width(花萼宽度)、Petal.Length(花瓣长度)和 Petal.Width(花瓣宽度)(单位：cm)，以及对应的品种(Species)，试分别计算每个品种的上述 4 项指标的算术平均数。

在例 3－3 中，我们使用 R 内置数据集 iris。通过 head()命令查看数据，可以看到数据集包含 5 个变量，本例中使用的分组变量为品种(Species)，使用 aggregate()函数分组计算描述性统计数，代码和运行结果如下：

代码 3－6　通过 aggregate()函数分组计算例 3－3 中数据的描述性统计数

```
> data(iris)
> aggregate(Petal.Length~Species, data = iris, FUN = mean)
    Species Petal.Length
1    setosa        1.462
2    versicolor    4.260
3    virginica     5.552
> aggregate(.~Species, data = iris, FUN = sum, na.rm = TRUE)
    Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1    setosa       250.3        171.4        73.1         12.3
2    versicolor   296.8        138.5        213.0        66.3
3    virginica    329.4        148.7        277.6        101.3
```

在上例中，data(iris)表示加载 R 语言内置的 iris 数据集，第二行使用 aggregate()函数按 Species 列对 Petal.Length 列进行分组，Petal.Length～Species 是一个公式，表示对 Petal.Length 列按照 Species 列进行分组。data = iris 指定了数据框，FUN = mean 指定了我们要计算的统计量(平均数)。第三行使用“.”符号(dot notation)来表示对数据框中的所有数值列进行分析。.～Species 表示对除了 Species 列之外的所有列按照 Species 列

进行分组，FUN = sum 指定了我们要计算的统计量（总和），na.rm = TRUE 表示在计算之前去除缺失值。

3.3.2 describeBy()函数

aggregate() 函数每次调用只能返回单个统计数，若要实现一次返回多个统计数，可以使用 psych 包中的 describeBy()函数。describeBy()函数用于按照指定的分组变量对数据进行描述性统计分析。该函数能够返回包括非缺失值的数量、平均数、标准差、中位数、截尾均值、绝对中位差、最小值、最大值、极差、偏度系数、峰度系数和平均数标准误等在内的多种统计量，为数据分析提供了全面的视角。其一般使用格式为：

```
describeBy(x, group, ..., mat = if (is.matrix(x)) TRUE else FALSE,
           trim = 0.1, type = 2, range = TRUE, na.rm = FALSE,
           describe.args = NULL)
```

x：数据框（data frame）或矩阵（matrix），包含需要分析的变量。

group：分组变量，通常是一个因子（factor）或可以转换为因子的变量，用于指定数据如何分组。

...：其他可选参数，用于传递额外的参数给内部函数。

mat：逻辑值，如果为 TRUE，则结果以矩阵形式返回；如果为 FALSE，则结果以数据框形式返回（对于数据框输入默认为 FALSE）。

data：如果使用公式输入（如 x～group），则需要指定数据框。

trim：用于计算截尾均值的比例（0 到 0.5 之间），默认值为 0.1，即去掉每端 10％的数据。

type：定义偏度（skewness）和峰度（kurtosis）的计算方式。默认值为 2，表示使用 G1 标准化方法。

range：逻辑值，如果为 TRUE，则计算最大值和最小值之间的范围；如果为 FALSE，则不计算。

na.rm：逻辑值，如果为 TRUE，则在计算前去除缺失值；如果为 FALSE，则保留缺失值（默认为 FALSE）。

describe.args：一个列表，包含传递给 describe() 函数的额外参数（describe() 是 psych 包中用于计算描述性统计量的函数）。

使用 psych 包中的 describeBy()函数分组计算例 3－3 中数据的描述性统计数，代码和运行结果如下：

代码 3－7　通过 psych 包的 describeBy()函数分组计算例 3－3 数据的描述性统计数

```
> library(psych)
> data(iris)
```

```
> describeBy(iris[, 1:4], group = iris$Species, na.rm = TRUE)

  Descriptive statistics by group
group: setosa
             vars  n mean  sd median trimmed  mad min max range skew kurtosis   se
Sepal.Length    1 50 5.01 0.35    5.0    5.00 0.30 4.3 5.8   1.5 0.11    -0.45 0.05
Sepal.Width     2 50 3.43 0.38    3.4    3.42 0.37 2.3 4.4   2.1 0.04     0.60 0.05
Petal.Length    3 50 1.46 0.17    1.5    1.46 0.15 1.0 1.9   0.9 0.10     0.65 0.02
Petal.Width     4 50 0.25 0.11    0.2    0.24 0.00 0.1 0.6   0.5 1.18     1.26 0.01
------------------------------------------------------------
group: versicolor
             vars  n mean  sd median trimmed  mad min max range  skew kurtosis   se
Sepal.Length    1 50 5.94 0.52   5.90    5.94 0.52 4.9 7.0   2.1  0.10    -0.69 0.07
Sepal.Width     2 50 2.77 0.31   2.80    2.78 0.30 2.0 3.4   1.4 -0.34    -0.55 0.04
Petal.Length    3 50 4.26 0.47   4.35    4.29 0.52 3.0 5.1   2.1 -0.57    -0.19 0.07
Petal.Width     4 50 1.33 0.20   1.30    1.32 0.22 1.0 1.8   0.8 -0.03    -0.59 0.03
------------------------------------------------------------
group: virginica
             vars  n mean  sd median trimmed  mad min max range  skew kurtosis   se
Sepal.Length    1 50 6.59 0.64   6.50    6.57 0.59 4.9 7.9   3.0  0.11    -0.20 0.09
Sepal.Width     2 50 2.97 0.32   3.00    2.96 0.30 2.2 3.8   1.6  0.34     0.38 0.05
Petal.Length    3 50 5.55 0.55   5.55    5.51 0.67 4.5 6.9   2.4  0.52    -0.37 0.08
Petal.Width     4 50 2.03 0.27   2.00    2.03 0.30 1.4 2.5   1.1 -0.12    -0.75 0.04
```

在上例中，使用 describeBy() 函数对 iris 数据集中的前四个变量（Sepal.Length、Sepal.Width、Petal.Length、Petal.Width）按照 Species 变量进行了分组描述性统计，并通过设置 na.rm＝TRUE 来去除缺失值（虽然在这个数据集中没有缺失值）。运行代码后，得到一个包含每个鸢尾花品种的四个变量描述性统计量的列表。每个列表项都是一个数据框，包含了该种类下对应变量的样本容量（n）、平均数（mean）、标准差（sd）、中位数（median）、截尾均值（trimmed）、绝对中位差（mad）、最小值（min）、最大值（max）、值域（range）、偏度系数（skew）、峰度系数（kurtosis）和平均数标准误（se）等统计量。

3.4　频数分布分析

频数分布（frequency distribution）是指按照某种规则将数据分成若干组，分别统计各组数据的观察值数目（频数），以反映数据分布的情况。频数分布分析是对数据进行描述性统计的重要方式。

3.4.1 频数分布表

在 R 语言中，table()函数用于创建频数分布表。它可以对单个变量进行计数，得到每个取值范围出现的频数；也可以对多个变量进行操作，构建交叉表(列联表)，展示不同变量取值组合的频数情况。这有助于分析数据的分布特征、变量之间的关系等。table()函数的语法结构如下：

```
table(x, ..., exclude = c(NA, NaN), useNA = c("no","ifany","always"), dnn = list.names(x, ...))
```

x：用于分析的数据，可以是向量、因子或者列表。如果是向量，函数将计算向量中每个不同元素的频数；如果是因子，将按照因子的水平计算频数；如果是列表，则会对列表中的每个元素递归地应用 table()函数。

...：可以传入更多的向量、因子或列表，用于构建交叉表。

exclude：指定要排除的值，默认排除 NA 和 NaN 值。例如，exclude＝NULL 表示不排除任何值。

useNA：控制 NA 值的处理方式，"no"表示不将 NA 作为一个单独的类别进行计数(默认情况)；"ifany"表示如果数据中存在 NA，则将 NA 作为一个单独的类别进行计数；"always"表示总是将 NA 作为一个单独的类别进行计数，即使数据中实际上没有 NA。

dnn：一个命名列表，用于指定结果的维度名称。如果不指定，将根据输入变量自动命名。

例 3－4 将某小麦品种 100 个麦穗的小穗数资料(见表 3－3)整理成次数分布表。

表 3－3 100 个小麦麦穗的小穗数

18	15	17	19	16	15	20	18	19	17
17	18	17	16	18	20	19	17	16	18
17	16	17	19	18	18	17	17	17	18
18	15	16	18	18	18	17	20	19	18
17	19	15	17	17	17	16	17	18	18
17	19	19	17	19	17	18	16	18	17
17	19	16	16	17	17	17	16	17	16
18	19	18	18	19	19	20	15	16	19
18	17	18	20	19	17	18	17	17	16
15	16	18	17	18	16	17	19	19	17

我们将表 3－3 中的数据录入 Excel 表格中的一列，列名称为 number，然后保存为 Example3_3.csv 文件。

代码 3-8　利用 table()函数对小麦小穗数进行频数分布分析

```
> example3_3 <- read.table("E:/RinBio/Chapt3/Example3_3.csv", header = TRUE, sep =",")
> table(example3_3)
number
15 16 17 18 19 20
 6 15 32 25 17  5
```

执行上述代码后，将对每一个出现的小穗数进行计数，由结果可知：小穗数在 15、16、17、18、19、20 的时候，分别出现了 6、15、32、25、17 和 5 次。

对于字符型和因子型的向量，table()函数可以直接计算其每个水平的频数。对于数值型向量，则需要通过 cut()函数将其转化为因子型。cut()函数在 R 语言中是一个功能强大的工具，它允许我们将连续的数值型数据分割成一系列的区间(也称为分组或桶)，并将这些数值映射到相应的区间标签上。这在进行数据分组分析、可视化以及构建分类模型时非常有用。cut()函数的语法结构为：

```
cut(x, breaks, labels = NULL, include.lowest = FALSE, right = TRUE, digits = 5,
  ordered = FALSE, na.omit = FALSE, ...)
```

x：一个数值型向量或表达式，表示要分割的数据。

breaks：定义分割区间的向量、数值或函数。可以是指定的区间边界，也可以是生成区间边界的函数(如 seq()、pretty()等)，或者是表示区间数量的单个整数。

labels：一个可选的字符向量，用于为分割后的区间指定标签。如果省略，则默认使用区间的边界值作为标签。

include.lowest：逻辑值，表示是否包含最小值的区间。当 breaks 为数值或函数生成的区间时，若设为 TRUE，则最小值的区间会包含最小值本身。

right：逻辑值，表示区间是否为右闭区间(即每个区间的右边界是包含的)，默认为 TRUE。

digits：用于控制标签中小数点后的位数。

ordered：逻辑值，表示返回的因子是否为有序因子。默认为 FALSE。

na.omit：逻辑值，表示是否忽略 NA 值。默认为 FALSE。

...：其他可能的参数，用于控制函数的某些特定行为。

在下面的例子中，我们对 iris 数据集的 Species 和 Petal.Length 两个变量进行频数统计，其中 Petal.Length 为数值型变量。

代码 3-9　利用 table()函数对 iris 数据集进行频数统计

```
> myiris <- iris
> table(myiris$Species)

    setosa versicolor  virginica
        50         50         50
```

```
> myiris$Petal.Length <- cut(myiris$Petal.Length,
+                                    breaks = c(0,3,5,8),
+                                    labels = c("low","middle","high") )
> table(myiris$Species, myiris$Petal.Length)

            low middle high
 setosa      50      0    0
 versicolor  1       48   1
 virginica   0       9    41
```

在上例中，为了便于对数据进行编辑，我们先将内置数据集 iris 赋值给 myiris 数据框。通过对因子型变量 Species 进行频数分析，可以得知每个品种包含的观察值数目均为 50。随后我们通过 cut()函数将数值型变量 Petal.Length 转化成了具有三个水平的因子，并设置每个水平的标签。使用 table()函数对 Species 和 Petal.Length 两个变量做频数分析，可以得到一个二维列联表，其中包括 2 个变量 3 个水平的全部组合所包含的观察值数目。

在上例中，我们简单地根据经验值设置了变量 Petal.Length 的分类标准。在对连续性数值变量进行统计分析时，需要按照一定的规则对数据进行分组。制作连续性变量的频数分布表一般包含以下四个步骤：

① 计算数据的最大值、最小值和极差，为决定组距与组数提供依据；

② 决定组数与组距：组数以能够反映出频数分布的特征为原则，一般为 10－15 个，组距＝极差/组数，可通过适当的调整，使组距尽量取便于计算的数值；

③ 决定组限：组限就是表明每组两端的数值，通常区间是左闭右开型的；

④ 制作频数分布表：统计落在各个小组内的数据个数，即为频数。

我们按照上述四个步骤对例 3－1 中的数据制作频数分布表，代码及运行结果如下：

代码 3－10　制作例 3－1 数据的频数分布表

```
> example3_1 <- read.table("E:/RinBio/Chapt3/Example3_1.csv", header = TRUE, sep =",")
> range(example3_1$length)
[1] 27.25 32.38
> breaks <- seq(from = 27, to = 32.5, by = 0.5)
> breaks
 [1] 27.0 27.5 28.0 28.5 29.0 29.5 30.0 30.5 31.0 31.5 32.0 32.5
> length <- cut(example3_1$length, breaks = breaks)
> table(length)
length
(27,27.5] (27.5,28] (28,28.5] (28.5,29] (29,29.5] (29.5,30]
        1         3         6        13        18        20
(30,30.5] (30.5,31] (31,31.5] (31.5,32] (32,32.5]
       16        16         7         4         2
```

在上例中，我们通过 range()函数求得变量的值域为 27.25－32.38，计算可得极差为 5.13；根据极差设置组数为 11、组距为 0.5 是较为合理的；根据变量的值域，将最低组的低限设置为 27，最高组的高限设置为 32.5，利用 seq()函数生成等差数列作为分组的组限值；然后，利用 cut()函数将向量转化为因子，并利用 table()函数生成频数统计表。

3.4.2　频数分布图

使用频数分布图可以直观地展示数据的分布情况，便于观察和发现规律。本小节中简要介绍如何绘制频数分布相关图形，包括直方图、核密度图、条形图。更加详细的绘图功能介绍请参考本书第 10 章的内容。

(1) 直方图

直方图(histograms)通常用来描述连续型变量，其绘制过程与制作频数分布表相似，首先需要将数据按照一定的间隔进行分组，然后分别统计每一组内的个体数目。直方图的横坐标代表数据的分组区间，纵坐标代表频数。绘制直方图通常使用 hist()函数，其基本用法如下：

```
hist(x, breaks =, labels = c(""), freq = , right = , main ="",
       xlim = c(), ylim = c(), xlab ="", ylab = "", plot =,)
```

x：数值型向量。

breaks：数值型向量或数值，若为向量，则用于设置组限；若为数值，则用于设置组数，自动计算组距。

labels：字符型向量，设置组区间标签。

freq：逻辑值，TRUE 表示显示每个区间内的频数，FALSE 表示显示频率(频数/总数)。

right：逻辑值，TRUE 表示组区间左开右闭，FALSE 表示组区间左闭右开。

main：字符串，设置图表标题。

xlim、ylim：数值型向量，设置 x 轴和 y 轴的取值范围。

xlab、ylab：字符串，设置 x 轴、y 轴标签。

plot：逻辑值，为 FALSE 时不绘制图形，而是返回 breaks、counts 等列表。

以上参数中，只有向量 x 是必须提供的，在没有指定其他参数的情况下，R 会自动根据数据分布情况计算出合适的组数和组距，并生成直方图。

(2) 核密度图

核密度图(density plot)是一条平滑曲线，其横坐标代表取值范围，纵坐标代表一定区间内的概率密度，核密度曲线与横轴围成的图形面积为 1。核密度估计实质上是根据数据的概率分布估计其概率密度函数的一种非参数测验方法，同时它也是用来观察连续性变量分布情况的有效方法，在本章中我们只介绍其简单用法。核密度估计通过 density()函数完成，其基本用法如下：

```
density(x, n =, from =, to = )
```

x:数值型向量。

n:数值,表示用于估算密度的等距点数目。

from、to:数值,指定核密度计算的范围的低限和高限。

完成核密度估计后,使用 plot()函数生成图形。plot()是 R 中的基础绘图函数,其基本用法如下:

```
plot(x, y = , type = "p", main ="", sub ="", xlab ="", ylab ="", xlim = c(), ylim = c(), ...)
```

x、y:绘图向量。

type:字符串,表示绘图类型,“p”表示绘制点图,“l”表示线图,“b”表示点线图,“s”表示阶梯图。

main、sub:字符串,分别设置图表标题和副标题。

xlab、ylab:字符串,分别设置 x 轴、y 轴标签。

xlim、ylim:数值型向量,分别设置 x 轴和 y 轴的取值范围。

对例 3-1 中“岱字棉”纤维长度数据绘制直方图和核密度图,代码和运行结果如下:

代码 3-11　对例 3-1 数据绘制直方图和核密度图

```
> example3_1 <- read.table("E:/RinBio/Chapt3/Example3_1.csv", header = TRUE, sep =",")
> hist(example3_1$length)
> d <- density(example3_1$length, n = length(example3_1$length))
> plot(d, main ="")
```

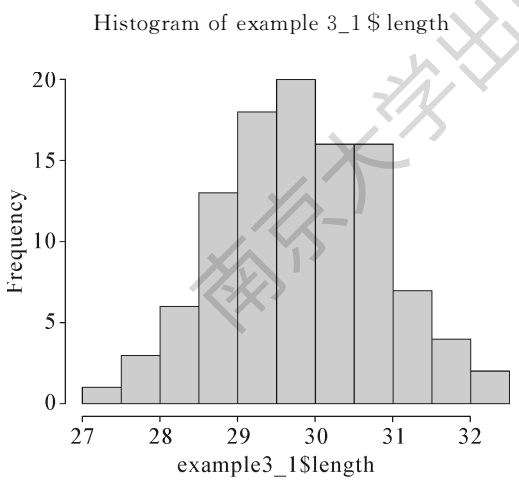

图 3-3　直方图

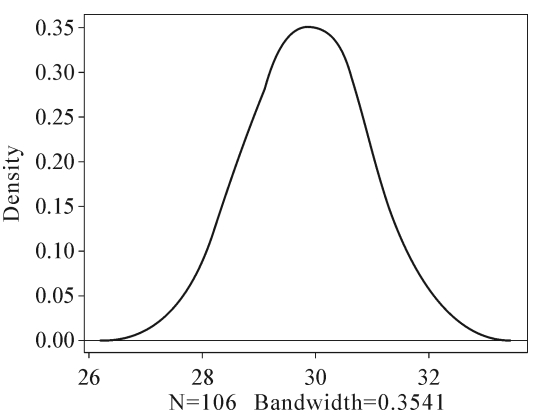

图 3-4　核密度图

通过绘制直方图和核密度图(图 3-3、图 3-4),我们可以看出数据分布的以下特点:①“岱字棉”的纤维长度分布范围为 27.0—32.5 mm 之间;② 大部分“岱字棉”的纤维长度分布在 28.5—31.0 mm 之间;③ “岱字棉”的纤维长度是以 29.5—30.0 mm 为中心左右对称分布的。

(3) 条形图

对于间断型变量,通常使用条形图(barplot)来绘制其分布模式,条形图与直方图结构

相似，横轴表示数据的分组区间，纵轴表示分布频数。由于变量是间断不连续的，条形图分组区间之间具有间隔。

在绘制条形图之前，首先需要通过 table()函数计算每个组区间的频数。绘制条形图通常使用 barplot()函数，其基本用法如下：

```
barplot(x,  horiz =, beside =)
```

x：数值型向量，包含每个柱子的高度。

horiz：逻辑值，TRUE 表示绘制水平条形图。

beside：逻辑值，TRUE 表示将多个柱形并列放置，FALSE 表示堆叠放置。

此外，基础函数 plot()的各项参数同样适用于 barplot()，因此不再赘述。

例 3－5　将例 3－4 中 100 个麦穗的小穗数绘制成频率分布图。

代码 3－12　对例 3－4 数据绘制频数分布条形图

```
> example3_3 <- read.table("E:/RinBio/Chapt3/Example3_3.csv", header = TRUE, sep =",")
> counts <- table(example3_3$number)
> barplot(counts, xlab ="number", ylab ="frequency")
```

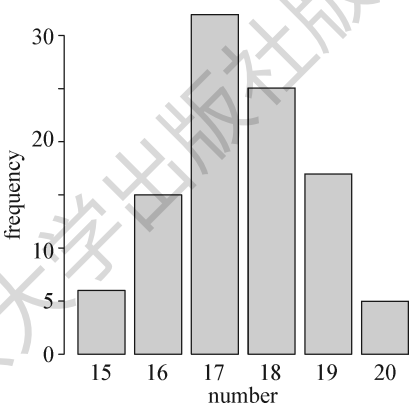

图 3－5　频数分布条形图

在上例中，使用 table() 函数对 example3_3 数据框中的 number 列进行计数，生成一个频数分布表，存储在对象 counts 中。随后使用 barplot() 函数绘制条形图(如图 3－5)，使用的数据为 counts，并设置 x 轴标签为"number"，y 轴标签为"frequency"。输出的条形图结果显示，麦穗的小穗数的取值范围为 15－20 区间内的整数，出现频率最高的小穗数为 17 个，其次为 18 个。

习　题

3－1　一组数据的分布特征可以从哪几个方面进行测度？

3－2　下表为 140 行水稻的产量，试用 R 对该数据进行描述性统计分析。要求如下：

(1) 输出结果至少包括下列统计数，平均数、中位数、众数、标准差、方差、极差、标准

误差；

（2）制作频数分布表；

（3）绘制频数分布直方图和核密度图。

140 行水稻产量(单位:克)

177	215	197	97	123	159	245	119	119	131	149	152	167	104
161	214	125	175	219	118	192	176	175	95	136	199	116	165
214	95	158	83	137	80	138	151	187	126	196	134	206	137
98	97	129	143	179	174	159	165	136	108	101	141	148	168
163	176	102	194	145	173	75	130	149	150	161	155	111	158
131	189	91	142	140	154	152	163	123	205	149	155	131	209
183	97	119	181	149	187	131	215	111	186	118	150	155	197
116	254	239	160	172	179	151	198	124	179	135	184	168	169
173	181	188	211	197	175	122	151	171	166	175	143	190	213
192	231	163	159	158	159	177	147	194	227	141	169	124	159

3－3 测定了 4 种密度下“金皇后”玉米的千粒重(克)，得结果如下表，试利用 R 分组计算每种密度之下玉米千粒重的描述性统计数。

种植密度(千株/亩)			
2	4	6	8
247	238	214	210
258	244	227	204
256	236	221	200
251	246	218	210

第4章

假设测验

假设测验，又称为显著性检验(significant testing)，指的是运用抽样分布等概率原理，利用样本资料测验这些样本所在总体(即处理)的参数有无差异，并对测验的可靠程度做出度量的过程。

假设测验具体操作过程为根据某种实际需要对未知的或不完全知道的统计总体提出意义相反的两种假设，然后由样本的实际结果，经过一定的计算，推断在概率意义上应当接受哪一种假设。例如要比较两个品种的产量有无差异，一个新选育出的棉花品种的纤维长度是否达到相应的国家标准，两种农药对某种虫害的防治效果是否一样，等等。这些问题不能通过简单的比较来下结论，必须通过概率计算做出选择，这就是统计假设测验要研究的问题。根据实际问题进行统计分析时数据资料所符合的理论分布模型不同，以及样本和统计量的不同，统计假设测验方法也不同，如对连续性变量资料的平均数进行测验可使用 u 测验和 t 测验，而对方差进行测验可使用 F 测验和 χ^2 测验，对于次数资料的分析可使用 χ^2 测验等。

4.1 F 测验

F 测验也称为方差的齐性测验，目的是测验两个抽自正态总体的独立样本的方差 s_1^2 和 s_2^2 所属的总体方差 σ_1^2 和 σ_2^2 是否有显著差异的测验。已知样本方差 s_1^2 和 s_2^2 的比率遵循 F 分布，因此需采用 F 测验。需要注意的是，F 测验也分为一尾测验和两尾测验。

两尾测验的无效假设和备择假设分别为：对 H_0，$\sigma_1^2=\sigma_2^2$；对 H_A，$\sigma_1^2\neq\sigma_2^2$。一尾测验的无效假设和备择假设分别为：对 H_0，$\sigma_1^2\leqslant\sigma_2^2$；对 H_A，$\sigma_1^2>\sigma_2^2$。

在 R 中，可以通过 var.test()函数进行方差的齐性测验，其一般使用格式为：

```
var.test(x, y , ratio = 1, alternative = c("two.sided", "less", "greater"),
             confidence.level = 0.95, ...)
```

x，y：这是两个必需的参数，代表用于比较的两组数据。它们可以是数值向量，分别包含两组独立样本的观测值。

ratio：假设的方差比(x 的方差与 y 的方差之比)。默认值为 1，即假设两个样本的方差相等。

alternative:指定备择假设的类型。可以是"two.sided"(双侧检验,即检验两个方差是否不相等)、"less"(单侧检验,即检验第一个样本的方差是否小于第二个样本的方差)或"greater"(单侧检验,即检验第一个样本的方差是否大于第二个样本的方差)。默认值为"two.sided"。

confidence.level:置信水平,用于计算置信区间。默认值为 0.95。

...:其他参数,通常用于传递额外的参数给底层函数,但 var.test()函数中很少使用。

var.test()函数返回值是一个列表对象,主要包括:statistic 是 F 检验统计量,为两组样本方差的比值;parameter 给出 F 分布的分子和分母自由度;p.value 是检验的 p 值,可用于判断是否拒绝两总体方差相等的原假设;conf.int 是总体方差比的置信区间;estimate 是样本方差比的估计值;null.value 为原假设设定的方差比,默认是 1;alternative 指明备择假设类型,如双侧、左侧或右侧;method 显示使用的是"F test to compare two variances"方法;data.name 记录参与检验的数据名称。

例 4-1 为分析一种新研制的杀虫剂在白菜中残留量,测定生长季节喷过该杀虫剂的白菜叶片混合样品 5 次,得药剂含量(mg)分别为:13.4,12.8,14.6,13.1,14.2;同时取未喷过该杀虫剂(对照)的叶片混合样品 4 次,测定得药剂含量(mg)分别为:8.8,9.1,8.5,7.9,试分析两个样本的方差是否存在显著差异。

代码 4-1 F 测验

```
> x <- c(13.4, 12.8, 14.6, 13.1, 14.2)
> y <- c(8.8, 9.1, 8.5, 7.9)
> var_test_result <- var.test(x, y)
> print(var_test_result)

    F test to compare two variances
data:  x and y
F = 2.179, num df = 4, denom df = 3, p-value = 0.5483
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.1442984 21.7451488
sample estimates:
ratio of variances
          2.179048
```

本例中,首先定义了两个数值向量 x 和 y,其中 x 包含值 13.4、12.8、14.6、13.1、14.2,y 包含值 8.8、9.1、8.5、7.9;接着使用 var.test() 函数对这两个向量 x 和 y 进行方差齐性检验,将检验结果存储在变量 var_test_result 中;最后使用 print()函数输出方差齐性检验的结果。根据计算的结果,F 值为 2.179,x 变量的自由度为 4,y 变量的自由度为 3,无效假设 H_0 发生的概率值为 0.548 3,大于设定的显著性水平 0.05,因此在 0.05 的显著性水平上接受无效假设 H_0,即两个样本的方差不存在显著差异。

4.2　t 测验

t 测验，也被称为学生氏 t 测验(Student's t test)，主要用于样本含量较小(例如 $n<30$)，总体标准差 σ^2 未知的正态分布。t 测验是用 t 分布理论来推论差异发生的概率，从而比较两个平均数的差异是否显著。t 测验可分为单个样本平均数的 t 测验、两个独立样本的 t 测验和两个配对样本的 t 测验。

在 R 中，t.test()函数是用于执行 t 检验的函数，t.test()函数可以执行单侧或双侧的 t 检验，包括单个样本 t 检验、两个独立样本 t 检验(假设方差相等或不相等)以及配对样本 t 检验。其一般使用格式为：

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95,...)
```

x：对于单样本 t 检验，x 是一个数值向量，代表样本数据；对于双样本 t 检验，x 是第一个样本的数据向量。

y：对于双样本 t 检验，y 是第二个样本的数据向量，默认为 NULL。

alternative：字符型，指定备择假设，有"two.sided"(双侧)、"less"(左侧)、"greater"(右侧)三种选择，默认为 "two.sided"。"two.sided"表示检验均值是否不等于 mu(单样本)或两个样本均值是否不等(双样本)。"less"表示检验均值是否小于 mu(单样本)或第一个样本均值是否小于第二个样本均值(双样本)。"greater"表示检验均值是否大于 mu(单样本)或第一个样本均值是否大于第二个样本均值(双样本)。

mu：对于单样本 t 检验，mu 是假设的总体均值，默认为 0。

paired：逻辑型，对于配对样本 t 检验，设置为 TRUE，默认为 FALSE。

var.equal：逻辑型，对于双样本 t 检验，设置为 TRUE 表示假设两个样本方差相等，默认为 FALSE。

conf.level：数值，指定置信水平，默认为 0.95，即 95%的置信区间。

...：用于传递其他参数，通常在默认情况下不需要使用。

t.test()函数返回值是一个列表，包含了多个对检验结果进行解读的关键信息。statistic 是计算得出的 t 统计量，反映样本均值与假设值或两样本均值间差异和样本变异性的关系；parameter 代表 t 分布的自由度，不同类型 t 检验自由度计算方式有别；p.value 为在零假设成立时得到当前样本数据或更极端数据的概率，用于判断是否拒绝零假设；conf.int 是总体均值或两总体均值差的置信区间，体现总体参数可能的取值范围；estimate 是样本均值或两样本均值差的估计值；null.value 是零假设中设定的总体均值或均值差值；alternative 表明备择假设的类型，有双侧、左侧和右侧三种；method 指出具体的 t 检验方法，如单样本、两独立样本或配对样本 t 检验；data.name 显示参与检验的数据对象名称，便于识别数据来源。

4.2.1 单个样本平均数的假设测验

单个样本平均数 t 测验目的是测验一个样本平均数 $\bar{y}$ 的总体平均数 μ 与某一指定的总体平均数 μ_0 是否相等。

单个样本平均数 t 测验的应用条件包括：总体方差未知；小样本($n<30$)连续型变量资料；样本来自正态分布总体。

单个样本平均数 t 测验中 t 值的计算方式为：

$$t=\frac{\bar{y}-\mu_0}{s_{\bar{y}}}$$

其中 $s_{\bar{y}}$ 为样本平均数的标准误，计算方法为：

$$s_{\bar{y}}=s/\sqrt{n}$$

t 分布是随自由度 df 的不同而变化的一组曲线，此处 t 分布的自由度为 $n-1$。

单个样本平均数 t 测验的步骤包括：

① 提出无效假设 $H_0:\mu=\mu_0$，即两个总体平均数间不存在差异，实得差异是由于试验误差造成的；备择假设 $H_A:\mu\neq\mu_0$，即两个总体平均数间存在差异。

② 确定显著水平 α，在农学与生物学试验中，通常取 0.05 或 0.01。

③ 在 H_0 为正确的假设下，计算相应的 t 测验统计数，并基于 t 分布规律计算显著性测验概率值。

④ 如果此概率小于 α，则在 α 水平上否定 H_0，接受 H_A；即推断实得差异表明总体差数不同。如果这个概率大于或等于 α，则接受 H_0，即推断实得差异由误差造成，或者说要否定 H_0 尚证据不足。

例 4-2 某当地推广马铃薯品种单薯块重 $\mu_0=92$(g)，现有一新育成品种，在 10 个小区种植，得单薯块重(g)：93.1，92.9，95.5，93.8，94.6，96.2，90.5，91.7，94.8，93.6。试测验新品种的单薯块重与原当地品种单薯块重有无显著差异？

代码 4-2 单个样本平均数的假设测验

```
> t <- c(93.1, 92.9, 95.5, 93.8, 94.6, 96.2, 90.5, 91.7, 94.8, 93.6)
> one_sample_t_res <- t.test(t, mu = 92)
> print(one_sample_t_res)
    One Sample t - test
data: t
t = 3.0626, df = 9, p - value = 0.01352
alternative hypothesis: true mean is not equal to 92
95 percent confidence interval:
 92.43646 94.90354
sample estimates:
mean of x
     93.67
```

本例中，首先将新育成品种在 10 个小区种植所得的单薯块重数据存储在向量 t 中；使用 t.test()函数进行单样本 t 检验，原假设为新品种单薯块重的总体均值等于 92(即与原当地品种单薯块重无显著差异)，将检验结果存储在 one_sample_t_res 中；最后通过 print()函数输出单样本 t 检验的结果，以此判断新品种的单薯块重与原当地品种单薯块重是否存在显著差异。根据计算的结果，可知新育成品种单薯块重的平均值为 93.67。显著性测验的 t 值为 3.062 6，自由度为 9，无效假设发生的概率值为 0.013 52，该概率值小于显著水平 0.05，因此接受备择假设，否定无效假设，认为新品种的单薯块重与原当地品种单薯块重有显著差异。95%置信区间是[92.436 46，94.903 54]。

4.2.2　独立样本的 t 测验

两个独立样本的 t 测验，又称成组比较或组群比较。设有两个独立样本，第一样本具容量 n_1、平均数 $\bar{y}_1$，第二样本具容量 n_2、平均数 $\bar{y}_2$。其假设测验就是测验 $\bar{y}_1$ 的总体平均数 μ_1 是否显著不同于 $\bar{y}_2$ 的总体平均数 μ_2，可将无效假设设为 $H_0:\mu_1=\mu_2$，对应的备择假设为 $H_A:\mu_1\neq\mu_2$。

当两样本的总体方差相等时，即 $\sigma_1^2=\sigma_2^2$，在分析时宜先算得两样本的合并均方值，即均方平均数 $\bar{s}^2$：

$$\bar{s}^2=\frac{\sum(Y_1-\bar{y}_1)^2+\sum(Y_2-\bar{y}_2)^2}{(n_1-1)+(n_2-1)}$$

然后算得平均数差数的标准误 $s_{\bar{y}_1-\bar{y}_2}$ 为：

$$s_{\bar{y}_1-\bar{y}_2}=\sqrt{\bar{s}^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$$

进而计算出 t 测验统计数：

$$t=\frac{(\bar{y}_1-\bar{y}_2)-(\mu_1-\mu_2)}{s_{\bar{y}_1-\bar{y}_2}}$$

其服从 $v=(n_1+n_2-2)$ 的 t 分布。

当两样本的总体方差不相等时，即 $\sigma_1^2\neq\sigma_2^2$，用 s_1^2 和 s_2^2 作为 σ_1^2 和 σ_2^2 的估计，得到 t 统计量：

$$t=\frac{(\bar{y}_1-\bar{y}_2)-(\mu_1-\mu_2)}{\sqrt{s_1^2/n_1+s_2^2/n_2}}$$

严格地说，所得到的统计量 t 已不再服从 t 分布，但与适当自由度 ν' 的 t 分布很接近。ν' 由公式

$$\nu'=\frac{(s_1^2/n_1+s_2^2/n_2)^2}{s_1^4/(n_1-1)n_1^2+s_2^4/(n_2-1)n_2^2}$$

决定。ν' 一般不是整数，可以取与之最接近的整数代替之。这样近似地有

$$t=\frac{(\bar{y}_1-\bar{y}_2)-(\mu_1-\mu_2)}{\sqrt{s_1^2/n_1+s_2^2/n_2}}$$

所以在分析两个独立样本的 t 测验数据资料时，宜采用本章前面介绍的 var.test()函数检验两样本总体方差之间是否存在显著差异。若两样本总体方差之间不存在显著差异，则 var.equal＝FALSE；若两样本总体方差之间存在显著差异，则采用默认值。

例 4－3 测定前作喷洒过某种有机砷杀雄剂的麦田植株样本 4 次，植株体中的砷残留量为 7.5、9.7、6.8、6.4(毫克)；测定对照(前作未用过有机砷杀雄剂)的植株样本 3 次，得植株体中砷含量为 4.2、7.0、4.6。试测验两种情况下砷残留量是否具有显著差异。

代码 4－3 组群比较

```
> x <- c(7.5,9.7,6.8,6.4)
> y <- c(4.2,7.0,4.6)
> var.test(x,y)

    F test to compare two variances

data:  x and y
F = 0.94477, num df = 3, denom df = 2, p-value = 0.8978
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.02412244 15.15794939
sample estimates:
ratio of variances
           0.9447674

> independent_samples_t_res <- t.test(x, y, alternative = "two.sided",var.equal = TRUE)
> print(independent_samples_t_res)

    Two Sample t-test

data:  x and y
t = 2.0516, df = 5, p-value = 0.09544
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5901817  5.2568483
sample estimates:
mean of x mean of y
 7.600000  5.266667
```

本例中，首先代码定义了两个向量 x 和 y，x 存储了前作喷洒过杀雄剂的麦田植株样本的 4 次砷残留量数据；y 存储了对照(前作未用过杀雄剂)的植株样本的 3 次砷含量数据；接着使用 var.test(x,y)对两个样本进行 F 检验，以判断两样本方差是否相等；最后调用 t.test()函数进行两独立样本 t 检验，根据代码所示的结果，在 0.05 的显著性水平上，两

个方差并没有显著差异($F=0.944\ 77$，p value$=0.897\ 8$)，因此在 t.test()函数中我们将参数 var.equal 设置为“TRUE”，其次，因该例是测验两样本总体平均数是否具有显著差异，所以应该采用两尾测验，将参数 alternative 设置为“two.sided”，并将检验结果存储在 independent_samples_t_res 中，最后使用 print()函数输出该 t 检验的结果。以此判断测定前喷洒过有机砷杀雄剂和未喷洒过的麦田植株体中砷残留量是否存在显著差异。

结果显示，计算的 t 值为 2.051 6，自由度为 5，概率值大于 0.05，表明两处理平均数间的差异不显著，有机砷杀雄剂在株体中没有显著残留。95%置信区间是[$-0.590\ 181\ 7$，$5.256\ 848\ 3$]，表明可以有 95%的信心认为两个样本的均值之差在这个范围内。

例 4-4　测定甲玉米品种的小区产量 10 次，小区记产面积 24 平方米，得结果为：22.4、22.8、25.0、23.2、23.3、25.2、23.7、21.6、22.3、22.6(kg)；测得乙玉米品种的小区产量为：21.8、21.4、21.4、21.3、20.9、22.1、21.4、20.2、21.6、21.0。试分析甲、乙玉米品种的产量是否存在显著差异。

代码 4-4　组群比较

```
> x <- c(22.4,22.8,25.0,23.2,23.3,25.2,23.7,21.6,22.3,22.6)
> y <- c(21.8,21.4,21.4,21.3,20.9,22.1,21.4,20.2,21.6,21.0)
> var.test(x,y)

    F test to compare two variances
data:  x and y
F = 4.872, num df = 9, denom df = 9, p-value = 0.0273
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  1.210139 19.614696
sample estimates:
ratio of variances
             4.872013

> independent_samples_t_res_2 <- t.test(x, y, alternative = "two.sided",var.equal = FALSE)
> print(independent_samples_t_res_2)

Welch Two Sample t-test
data:  x and y
t = 4.7339, df = 12.545, p-value = 0.0004301
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.029711 2.770289

sample estimates:
mean of x mean of y
    23.21     21.31
```

本例中，首先定义了两个向量 x 和 y，x 存储甲玉米品种 10 次小区产量数据，y 存储乙玉米品种的 10 次小区产量数据；使用 var.test(x,y)对两个样本进行方差齐性检验，结果显示 F 值为 4.872，p 值为 0.027 3，说明两样本方差不相等；随后调用 t.test()函数进行两独立样本 t 检验，因方差不相等，设置 var.equal＝FALSE，备择假设为双侧检验，并将结果存储在 independent_samples_t_res_2 中；最后使用 print()函数输出检验结果，以此判断甲、乙两个玉米品种的产量是否存在显著差异。

根据代码 4－4 所示的结果，两个变量的平均数分别为 23.21 和 21.31，平均数差数的 95％的置信区间为[1.029 711，2.770 289]，计算的 t 值为 4.733 9，自由度为 12.545，概率值小于 0.01，表明甲、乙两个品种的产量存在显著差异。

4.2.3 成对样本的 t 测验

若两个样本的观察值因某种联系而一一配对（例如相邻两区的产量，同窝两猪的体重等），我们根据一定的专业知识或经验可以判断一对的比较更为合理，则应作成对比较。两个成对样本的 t 测验，又称配对样本的 t 测验或成对比较。在分析时，只需假设两样本的总体差数的平均数 $\mu_d = \mu_1 - \mu_2 = 0$，而不必考虑两样本的总体方差 σ_1^2 和 σ_2^2 是否相等。

设样本 1 的 y_{1j} 和样本 2 的 y_{2j} 构成一成对的数据，差数 $d_j = y_{1j} - y_{2j}$，且共有 n 对独立的比较，即 $j=1,2,\cdots,n$，则差数的平均数 $\bar{d}$ 、标准差 s_d 和差数平均数的标准误 $s_{\bar{d}}$ 依次为：

$$\bar{d} = \frac{1}{n}\sum d, s_d = \sqrt{\frac{ss_d}{n-1}} = \sqrt{\frac{\sum (d-\bar{d})^2}{n-1}} = \sqrt{\frac{\sum d^2 - \frac{(\sum d)^2}{n}}{n-1}}, s_{\bar{d}} = \frac{s_d}{\sqrt{n}}$$

则：

$$t = \frac{\bar{d} - \mu_d}{s_{\bar{d}}}$$

具有 $v = n - 1$，μ_d 为差数的总体平均数。

例 4－5 为测定 A、B 两种病毒对烟草的致病力，取 8 株烟草，每一株皆半叶接种 A 病毒，另半叶接种 B 病毒，以叶面出现枯斑数的多少作为致病力强弱的指标，得结果于表 4－1。试测验两种病毒致病力的差异显著性。

表 4－1 两种病毒在烟叶上产生的枯斑数

株号	1	2	3	4	5	6	7	8
病毒 A	9	17	31	18	7	8	20	10
病毒 B	10	11	18	14	6	7	17	5

代码 4－5　成对比较

```
> x <- c(9, 17, 31, 18, 7, 8, 20, 10)
> y <- c(10, 11, 18, 14, 6, 7, 17, 5)
> paired_samples_t_res <- t.test(x, y, paired = TRUE)
> print(paired_samples_t_res)

    Paired t - test

data:  x and y
t = 2.6253, df = 7, p - value = 0.03414
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.3972029  7.6027971
sample estimates:
mean difference
              4
```

本例中，首先定义了两个向量 x 和 y，x 存储了 A 病毒在 8 株烟草上产生的枯斑数；y 存储了 B 病毒在对应的 8 株烟草上产生的枯斑数；由于是对同一株烟草的半叶分别接种 A、B 病毒，属于配对样本，所以调用 t.test()函数时设置 paired＝TRUE 进行配对样本 t 检验，并将检验结果存储在 paired_samples_t_res 中；最后使用 print()函数输出该配对样本 t 检验的结果，以此判断两种病毒致病力是否存在显著差异。

根据计算结果可知，平均数间的差数为 4，该差数 95% 的置信区间为 [0.397 202 9，7.602 797 1]，测验的 t 值为 2.625 3，自由度为 7，概率值为 0.034 14，该概率值小于 0.05，表明两种病毒的致病力具有显著差异。

4.3　卡方测验

在农业和生物学试验中有许多质量性状。这些性状往往难以用数量水平表示，而只能用出现的次数表示。例如将有芒白粒小麦与无芒红粒小麦杂交，在 F_2 代会出现有芒白粒、有芒红粒、无芒白粒、无芒红粒等植株类型。对于每种类型的各个个体，其量值很难测定，但观察各种类型植株的出现却显然是合理且方便的。此外，数量性状也可能用次数来表示。例如在玉米群体中，有无穗株（空杆）、单穗株、双穗株等类型。我们当然可以记录其每株穗数求得平均数，但如按每株穗数为 0、1、2 等而列成次数，当然也是一种正确的表示方式。因此，不论质量性状或数量性状，都是可能用次数表示的。

设某次数资料的总体，共有 k 个类型或组，每组个体的出现概率依次为 p_1，p_2，…，p_k，则在 n 次独立的观察中，各组的期望（理论）次数依次为 $E_1=np_1$，$E_2=np_2$，…，$E_k=np_k$。若各组的观察次数依次为 O_1，O_2，…，O_k，则数理统计学已经证明：

$$\chi^2=\sum_{i=1}^{k}\frac{(O_i-np_i)^2}{np_i}=\sum_{1}^{k}\frac{(O_i-E_i)^2}{E_i}$$

遵循 $v=(k-m)$ 的 χ^2 分布(这里的 m 是独立约束条件的个数)。所以,有 χ^2 分布可对次数资料作出有关测验。

在 R 中,进行卡方(χ^2)测验的函数为 chisq.test(),其一般的使用格式为:

```
chisq.test(x, y = NULL, correct = TRUE,
           p = rep(1 / length(x), length(x)), rescale.p = FALSE,
           simulate.p.value = FALSE, B = 2000)
```

x:一个二维表或矩阵,代表观测频数;也可以是因子对象。如果 x 是一个二维表,则每一行的和(或列的和,取决于 simulate.p.value 参数)代表分类变量的观测频数;如果 x 是一个因子向量且 y 也是一个因子向量,则 x 和 y 的交叉表会被计算并用于检验。

y:当 x 是一个因子向量时,y 也应为一个因子向量。此时,x 和 y 的交叉表会被用于卡方检验;如果 x 已经是一个二维表或矩阵,则 y 应设为 NULL。

correct:逻辑值,用于指示是否应用连续性校正。当样本量较小时,连续性校正可以提高检验的准确性。默认为 TRUE。

p:一个向量,用于指定期望频数的比例。如果提供,卡方检验会基于这些比例而非边际频数来计算期望频数。默认为 NULL。

rescale.p:逻辑值,当 p 参数被提供时有效。如果为 TRUE,p 参数中的比例会被重新缩放,以确保它们的和为 1。默认为 FALSE。

simulate.p.value:逻辑值,用于指示是否通过模拟来计算 p 值。这在观测频数非常小时特别有用,因为传统的卡方检验在这种情况下可能不准确。默认为 FALSE。

B:当 simulate.p.value=TRUE 时,B 指定模拟的次数。默认为 2 000 次。

chisq.test()函数返回值是一个包含多个元素的列表。statistic 为计算得到的卡方统计量,反映了观察频数与期望频数之间的偏离程度;parameter 表示卡方分布的自由度,由数据的行数和列数决定,不同情形下计算方式不同;p.value 是在原假设成立时,得到当前样本数据或更极端数据的概率,用于判断是否拒绝原假设,若小于设定的显著性水平(如 0.05),则拒绝原假设;method 明确了具体执行的卡方检验类型,例如卡方独立性检验或拟合优度检验等;data.name 给出参与检验的数据名称,方便识别数据来源;对于独立性检验,还会返回 expected,即根据原假设计算出的期望频数矩阵,用于和观察频数对比;部分情况下会有 residuals(皮尔逊残差)和 stdres(标准化残差),帮助分析各单元格对卡方统计量的贡献。

4.3.1 适合性测验

适合性测验(test of goodness-of-fit)是测验观察的实际次数和根据于某种理论或需要预期的理论次数是否相符合。所作的假设是 H_0:符合,H_A:不相符。

例 4-6 两对等位基因控制的两对相对性状遗传,如果两对等位基因完全显性且无连锁,则 F_2 的四种表现型在理论上应有 9∶3∶3∶1 的比例。有一水稻遗传试验,以稃尖

有色非糯品种与稃尖无色糯性品种杂交，其 F_2 的观察结果为稃尖有色非糯 491 株，稃尖有色糯稻 76 株，稃尖无色非糯 90 株，稃尖无色糯稻 86 株。试测验是否符合 9∶3∶3∶1 的理论比例。

代码 4-6　适合性测验

```
> x <- c(491, 76, 90, 86)
> goodness_fit_res <- chisq.test(x, p = c(9, 3, 3, 1), rescale.p = TRUE)
> print(goodness_fit_res)

    Chi - squared test for given probabilities

data:  x
X - squared = 92.706, df = 3, p - value < 2.2e - 16
```

本例中，首先定义向量 x，将稃尖有色非糯 491 株、稃尖有色糯稻 76 株、稃尖无色非糯 90 株、稃尖无色糯稻 86 株这些观察频数存入其中；接着使用 chisq.test()函数进行卡方拟合优度检验，p＝c(9, 3, 3, 1)设定了理论比例，rescale.p＝TRUE 确保理论比例总和为 1，并将检验结果存储在 goodness_fit_res 中；最后用 print()函数输出检验结果，以判断 F_2 代四种表现型的观察结果是否符合 9∶3∶3∶1 的理论比例。

根据计算结果可知，χ^2 值为 92.706，自由度为 3，测验的概率值小于 0.01，表明四种表型偏离了 9∶3∶3∶1 的理论比例。

4.3.2　独立性测验

独立性测验是测验不同因素的列联次数是彼此独立，还是相互关联。所谓独立，是指因素之间没有相关，或者说任一行(列)的次数比率都是齐性的；所谓关联，是指因素之间存在相互作用，或者说各行(各列)的次数比率是非齐性的。

进行独立性测验(test of independence)，需先将次数资料作成列联表(contingency table)；然后在“H_0 独立”为正确的假设下，算得表中每一细格与实际次数相应的理论次数；最后，根据公式算得 χ^2 值，作为 H_2 被接受或否定的依据。设列联表有 R 行 C 列，则所得 χ^2 值遵循 $v=(R-1)(C-1)$ 的 χ^2 分布。当所得 $\chi^2>\chi^2_\alpha$ 时，为在 α 水平上否定 H_0。

例 4-7　检测甲、乙、丙 3 种农药对烟蚜的毒杀效果：用甲农药处理 187 头烟蚜，其中 37 头死亡，150 头未死亡；用乙农药处理 149 头烟蚜，其中 49 头死亡，100 头未死亡；用丙农药处理 80 头烟蚜，其中 23 头死亡，57 头未死亡。试分析这三种农药对烟蚜的毒杀效果是否一致？

代码 4-7　独立性测验

```
> x <- matrix(c(37, 150, 49, 100, 23, 57), nrow = 2, ncol = 3)
> print(x)
     [,1] [,2] [,3]
```

```
[1,]   37   49   23
[2,]  150  100   57
> independence_res <- chisq.test(x)
> print(independence_res)

    Pearson's Chi-squared test

data:  x
X-squared = 7.6919, df = 2, p-value = 0.02137
```

本例中，首先，使用 matrix()函数创建了一个 2 行 3 列的矩阵 x，矩阵中的元素分别对应三种农药处理烟蚜后死亡和未死亡的数量，第一行依次为甲、乙、丙农药处理后烟蚜的死亡数，第二行依次为未死亡数；使用 print(x)输出矩阵 x 以便查看数据；调用 chisq.test()函数对矩阵 x 进行卡方独立性检验，将检验结果存储在 independence_res 中，此检验用于判断农药种类和烟蚜死亡情况是否相互独立，即三种农药毒杀效果是否一致；最后使用 print()函数输出卡方独立性检验的结果，根据结果可判断三种农药对烟蚜的毒杀效果是否存在显著差异。

根据计算结果可知，χ^2 值为 7.691 9，自由度为 2，测验的概率值小于 0.05，表明 3 种农药对烟蚜的毒杀效果有显著差异。

习 题

4-1 某春小麦良种的千粒重 34g，现自外地引入一高产品种，在 8 个小区种植，得其千粒重(g)为：35.6、37.6、33.4、35.1、32.7、36.8、35.9、34.6。试测验新引入品种的千粒重与当地良种有无显著差异。

4-2 测定了两个玉米自交系的株高，从甲中抽出 8 株，株高(cm)分别为 166，165，166，168，162，165，163，166。又从乙中抽出 8 株，株高分别为 164，161，157，165，165，163，162，163。试分析两个玉米自交系的株高整齐度是否存在显著差异。

4-3 为研究矮壮素矮化的效果，在抽穗期测定喷矮壮素小区 8 株玉米的株高、9 株对照区玉米的株高，其株高(cm)结果如下表。试测验矮壮素的矮化效果是否显著。

y_1(喷矮壮素)	y_2(对照)
160	170
160	270
200	180
160	250
200	270
170	290

续　表

y_1(喷矮壮素)	y_2(对照)
150	270
210	230
	170

4-4　选生长期、发育进度、植株大小和其他方面皆比较一致的两株番茄构成一组，共得 7 组，每组中一株接种 A 处理病毒，另一株接种 B 处理病毒，以研究不同处理方法的钝化病毒效果，下表结果为病毒在番茄上产生的病痕数目。试测验两种处理方法的差异显著性。

组别	y_1(A 法)	y_2(B 法)
1	10	25
2	13	12
3	8	14
4	3	15
5	5	12
6	20	27
7	6	18

4-5　在红稃尖绿叶鞘水稻和紫稃尖紫叶鞘水稻杂交的 F_2 代，观察到紫尖紫鞘 249 株、紫尖绿鞘 87 株、红尖紫鞘 79 株、红尖绿鞘 25 株。试测验 F_2 代比例是否符合 9∶3∶3∶1 的理论比例。

4-6　调查某苹果不同树龄各类枝组坐果数列于下表。试测验坐果率是否与枝组大小有关？

树龄	大枝组坐果	中枝组坐果	小枝组坐果	总和
A	119	50	91	260
B	160	48	112	320
C	158	116	118	392
总 和	437	214	321	972

第5章

方差分析

对于两个平均数的假设测验，一般可以采用 u 测验或 t 测验的方法完成。对于多个平均数的假设测验，由于测验程序繁杂、误差估计不精确，尤其是犯 α 错误的概率增加等原因，一般不再利用 t 测验两两进行，而采用一种统计方法——方差分析法（analysis of variance，简记为 ANOVA）。

方差分析方法最早是由 R. A. Fisher 于 1920 年前后对农业试验进行统计分析的时候提出的。方差分析的基本思想是将所有观察值的总变异分解成不同的变异来源，即对总变异来源的自由度和平方和进行分解，进而获得不同变异来源的总体方差的估值。通过构建适当的 F 值，进行 F 测验，完成多个样本平均数之间差异显著性测验。

方差分析不仅能够分析单因素多水平（处理）效应值间平均数的差异，还能同时分析两个因素甚至多个因素多水平间平均数的差异，以及各因素间的交互作用。当处理效应为固定效应时，可对各个处理平均数进行多重比较。当处理效应为随机效应时，进行方差分量的估计。

方差分析是生物领域中应用最为广泛的统计方法之一，在科学研究和生产实践中有着极重要的用途。

5.1 方差分析常用函数和包简介

5.1.1 aov()函数

在 R 中，lm()函数和 aov()函数都可以对数据进行方差分析，本章主要介绍 aov()函数。aov()函数是 R 语言中用于拟合和分析方差分析模型的函数，它属于 stats 包，因此不需要单独安装。aov()函数用于拟合固定效应的方差分析模型，可以处理单因素方差分析、多因素方差分析以及带有协变量的方差分析。aov()函数返回的对象可以用于进一步的分析，如计算平方和、均方、F 值、p 值等。其一般的使用格式为：

```
aov(formula, data = NULL, projections = FALSE, qr = TRUE,
    contrasts = NULL, ...)
```

formula：为表达式，定义方差分析的模型，其主要内容为：

(1) ～为分隔符号,左边为因变量,右边为自变量,例如某试验有 A 和 B 两个试验因素,指标变量用 y 表示,则表达式 y～A＋B 表示分析 A 和 B 因素主效,不计算互作效应;

(2) ＋用于区分自变量;

(3) :表示交互作用项,例如求取 A 和 B 的主效和交互作用效应,则可以使用表达式:y～A＋B＋A:B;

(4) * 表示所有可能的交互作用效应,如代码 y～A * B * C 与代码 y～A＋B＋A：B＋C＋A：C＋B：C＋A：B：C 等效;

(5) ^表示交互作用达到的层级,如代码 y～(A＋B＋C)^2 表示 y～A＋B＋A：B＋C＋A：C＋B：C;

(6) .表示包含除因变量外的所有变量,例如若一个数据框中包含变量 y、A、B 和 C,则代码 y～.表示 y～A＋B＋C。

data:一个可选的数据框(data frame),其中包含了公式中引用的所有变量。如果未指定,则使用当前环境中的变量。

projections:逻辑值,指示是否计算投影矩阵。默认为 FALSE。投影矩阵用于评估模型拟合的好坏,但在大多数情况下,用户不需要直接访问这个矩阵。

qr:逻辑值,指示是否返回 QR 分解。默认为 TRUE。QR 分解是线性代数中的一种方法,用于解决线性方程组或进行矩阵分解。在 aov()函数中,QR 分解用于内部计算,但用户通常不需要直接访问这个结果。

contrasts:用于指定因素的对比矩阵。默认情况下,R 会根据因素的水平自动选择合适的对比方式。可以通过这个参数来指定自定义的对比矩阵,以满足特定的分析需求。

...:其他参数,可以传递给底层函数以控制模型拟合的某些方面。

aov()函数返回一个列表对象,包含方差分析关键结果。terms 呈现模型公式结构,明确分析因素;residuals 为观测值与模型预测值的残差,可用于评估拟合效果;effects 体现各因素水平对响应变量的影响程度;coefficients 是模型系数估计值。fstatistic 包含 F 统计量、分子与分母自由度,用于检验组间差异显著性。rank 是模型的秩,与独立参数数量有关;call 记录函数调用语句;model 存储分析用的原始数据框。

5.1.2　TukeyHSD()函数

若 F 测验接受 H_0,说明处理间无显著差异,没有必要继续分析。若 F 测验否定 H_0,表明 k 个处理平均数间存在显著差异。这时,虽然至少有两个处理平均数间存在显著差异,但是并不清楚哪些平均数间有显著差异,哪些平均数间无显著差异。为了进一步弄清平均数间的差异显著性,有必要进行 k 个平均数间的两两比较,这在统计学上称为多重比较(multiple comparison)。Tukey(1952,1953)以学生化极差为理论根据,提出了专门用于两两比较的真实显著差数检验法(honestly significant difference,简称 HSD 法)。

在 R 中,TukeyHSD()函数是 R 语言中用于进行 Tukey's Honest Significant Difference (HSD)多重比较测试的函数,它通常在进行了方差分析(ANOVA)之后使用,以确定哪些组之间的均值存在统计学上的显著差异。其一般使用格式为:

```
TukeyHSD(x, which, ordered = FALSE, conf.level = 0.95, ...)
```

x:一个 aov 对象,即已经使用 aov()函数拟合的方差分析模型。

which:一个整数或字符向量,用于指定要进行比较的模型项。默认值为 1,表示比较第一个项;如果模型中有多个因子,可以通过提供因子名称或对应的数字索引来指定。

conf.level:置信区间的水平,默认为 0.95,即 95%置信区间。

ordered:逻辑值,如果为 TRUE,则结果将按照均值差异的大小进行排序。

TukeyHSD()函数返回值是一个列表对象,主要包括:列表中的每个元素对应方差分析模型里的一个因子,元素名即因子名;每个因子元素是一个矩阵,矩阵的行代表不同组间比较组合,列中 diff 是两组均值的差值,展示组间均值差异大小;lwr 和 upr 分别是均值差置信区间的下限和上限,用于判断两组均值差异有无统计学意义;p adj 是调整后的 p 值,考虑多重比较后判断组间均值差异的显著性;此外,对象有 conf.level 属性,记录构建置信区间所用的置信水平,默认是 0.95。

5.1.3 agricolae 包

agricolae 是一个专门针对农业领域的统计分析开发的 R 包,全称为 Statistical Procedures for Agricultural Research。其功能包括随机区组、完全随机、多因素随机、增广、拉丁方、格子方、alpha-lattice 等试验的设计及相应试验的统计分析、多重比较、混合线性分析、通径分析、聚类分析、AMMI 模型分析等。在 R 语言中可利用 agricolae 包中的 LSD.test、duncan.test、HSD.test、SNK.test 实现多种不同方法的多重比较,并用字母展示多重比较结果。本节主要介绍 LSD.test()函数,该函数可以实现保护性最小显著差数法(protected least significant difference,简称 LSD 法)。

该法是具有保护意义的 t 测验方法。其做法是:若处理平均数间的 F 测验差异不显著,则不进行多重比较;若处理平均数间的 F 测验差异显著,计算出显著水平 α 下的最小显著差数

$$LSD_{\alpha}=t_{\alpha,df_e}s_{\bar{y}i.-\bar{y}j.}$$

其中,t_{α,df_e} 是在自由度为 df_e 和显著水平为 α 条件下的两尾临界值,$s_{\bar{y}i.-\bar{y}j.}$ 是处理平均数差数标准误

$$s_{\bar{y}i.-\bar{y}j.}=\sqrt{2MS_e/n}$$

这里,MS_e 是方差分析中的误差均方。将任意两个处理平均数差数的绝对值 $|\bar{y}_{i.}-\bar{y}_{j.}|$ 与 LSD_{α} 比较。若 $|\bar{y}_{i.}-\bar{y}_{j.}|>LSD_{\alpha}$,则 $\bar{y}_{i.}$ 与 $\bar{y}_{j.}$ 在 α 水平上差异显著;反之,在 α 水平上差异不显著。

LSD.test()函数的一般使用格式为:

```
LSD.test(y, trt, DFerror, MSerror, alpha = 0.05, p.adj = c("none","holm","hommel",
"hochberg", "bonferroni", "BH", "BY", "fdr"), group = TRUE, main = NULL,console = FALSE)
```

y:方差分析对象,通常是 aov()函数返回的对象,代表已经进行过方差分析的数据。

trt:要进行多重比较的分组变量,通常是因子(factor)类型,表示不同的处理或组别。

DFerror：误差的自由度(degrees of freedom for error)。这个参数通常可以从方差分析的结果中获取。

MSerror：误差的均方(mean square for error)。这个参数也通常可以从方差分析的结果中获取。

alpha：显著性水平，默认为 0.05。用于确定差异的显著性。

p.adj：P 值矫正方法，可选值包括"none"(不进行矫正)、"holm"、"hommel"、"hochberg"、"bonferroni"、"BH"(Benjamini / Hochberg)、"BY"(Benjamini / Yekutieli)、"fdr"(false discovery rate)。不同的矫正方法适用于不同的实验设计和数据特征。

...：其他可能的参数，具体取决于函数的实现和版本。

LSD.test()函数返回值是一个列表对象，主要包括：groups 是一个数据框，展示了各处理组对应的均值、分组信息，通过不同字母标识哪些组之间差异不显著；means 给出各处理组的均值；df 为自由度，包含误差自由度等信息，是进行统计推断的重要参数；test 显示所使用的检验统计量(如 t 统计量)的值，用于判断组间差异；p.value 是检验的 p 值，能帮助我们判断不同组均值之间差异是否显著；alpha 是设定的显著性水平，一般默认是 0.05；description 简要描述了检验的基本信息。

5.2　单因素试验资料的方差分析

当在试验中考虑的因素只有一个时，则称该试验为单因素试验。单因素试验资料也称之为单向分组资料。该类试验假设有 k 个独立样本，每个样本皆有 n 个观察值，该资料就叫作具有 kn 个观察值的单向分组资料，其数据模式如表 5-1。

表 5-1　kn 个观察值的单向分组资料的模式

样本(处理)号	观察值 Y_{ij}			
样本(处理)1(Y_{1j})	Y_{11}	Y_{12}	…	Y_{1n}
样本(处理)2(Y_{2j})	Y_{21}	Y_{22}	…	Y_{2n}
⋮	⋮	⋮	⋮	⋮
样本(处理)k(Y_{kj})	Y_{k1}	Y_{k2}	…	Y_{kn}

方差分析采用的方法为 F 测验。设总变异的平方和为 SS_T，其由两部分组成，一部分是样本(处理)间的平方和 SS_t，另一部分为随机误差平方和(或样本内平方和) SS_e，于是有 $SS_T = SS_t + SS_e$。

其中，$SS_t = n\sum_{i=1}^{k}(\bar{y}_{i.} - \bar{y}_{..})^2$，$SS_e = \sum_{i=1}^{k}\sum_{j=1}^{n}(Y_{ij} - \bar{y}_{i.})^2$。

根据平方和可以获得样本间和样本内的方差(均方)，分别为：

$$MS_t = \frac{SS_t}{df_t}, MS_e = \frac{SS_e}{df_e}$$

其中 df_t 和 df_e 分别为样本间和样本内的自由度，分别为 $(k-1)$ 和 $k(n-1)$。

则根据样本间和样本内的方差可以获得 F 统计数，计算方法为：

$$F=\frac{MS_t}{MS_e}$$

如果样本平均数间的变异并不显著大于样本内(随机误差的变异)，则 F 测验不会给出显著的值(这时 F 的期望值是 1)；反之，将给出显著的值。

例 5-1 做一水稻施肥的盆栽试验，设 5 个处理：A 和 B 分别施用两种不同工艺流程的氨水，C 施碳酸氢铵，D 施尿素，E 不施氮肥。每个处理 4 盆(施肥处理的施肥量每盆皆为折合纯氮 1.2 克)，共 5×4=20 盆，随机放置于同一网室中。其稻谷产量(克/盆)列于表 5-2，试测验各处理平均数的差异显著性。

表 5-2 水稻施肥盆栽试验的产量结果

处理	观察值			
A(氨水 1)	24	30	28	26
B(氨水 2)	27	24	21	26
C(碳酸氢铵)	31	28	25	30
D(尿素)	32	33	33	28
E(不施)	21	22	16	21

该例中的数据，包含了需要进行方差分析的数值(产量)和处理，在 R 语言读取数据时，需读入到一个数据框，所以可以在 Excel 录入数据时，按照数据框的方式录入两列数据，trt 表示组别(处理)，y 表示水稻产量，将其保存为 Example5_1.csv(如图 5-1)。

	A	B
1	*y*	trt
2	24	A
3	27	B
4	31	C
5	32	D
6	21	E
7	30	A
8	24	B
9	28	C
10	33	D
11	22	E

	A	B
12	28	A
13	21	B
14	25	C
15	33	D
16	16	E
17	26	A
18	26	B
19	30	C
20	28	D
21	21	E
22		

图 5-1 例 5-1 的数据录入格式

代码 5－1　单因素试验资料的方差分析

```
> example5_1 <- read.csv("E:/RinBio/Chapt5/Example5_1.csv", header = TRUE)
> fit <- aov(y~trt, data = example5_1)
> summary(fit)
             Df Sum Sq Mean Sq F value   Pr(> F)
trt           4  301.2   75.30   11.18 0.000209 ***
Residuals    15  101.0    6.73
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> library(agricolae)
> mc_e5_1 <- LSD.test(fit,"trt")
> mc_e5_1
$statistics
     MSerror Df Mean        CV t.value       LSD
    6.733333 15 26.3 9.866413 2.13145 3.910886

$parameters
         test p.ajusted name.t ntr alpha
   Fisher-LSD       none    trt   5  0.05

$means
      y       std r        se       LCL      UCL Min Max   Q25   Q50   Q75
A  27.0  2.581989  4  1.297433  24.23459  29.76541  24  30  25.50  27.0  28.50
B  24.5  2.645751  4  1.297433  21.73459  27.26541  21  27  23.25  25.0  26.25
C  28.5  2.645751  4  1.297433  25.73459  31.26541  25  31  27.25  29.0  30.25
D  31.5  2.380476  4  1.297433  28.73459  34.26541  28  33  31.00  32.5  33.00
E  20.0  2.708013  4  1.297433  17.23459  22.76541  16  22  19.75  21.0  21.25

$comparison
NULL

$groups
      y groups
D  31.5      a
C  28.5      ab
A  27.0      bc
B  24.5      c
E  20.0      d
```

```
attr(,"class")
[1] "group"
> plot(mc_e5_1)
```

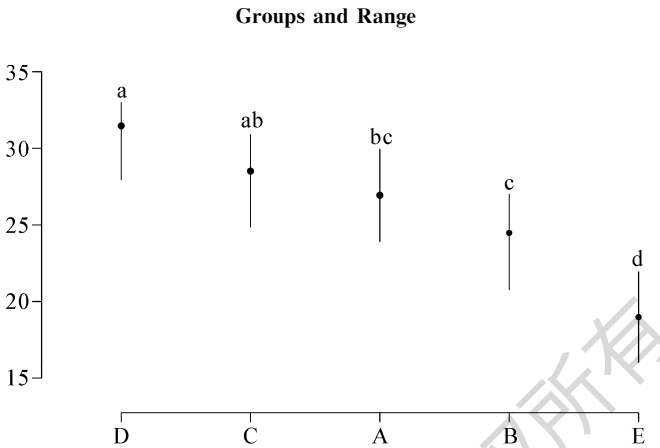

图 5-2　例 5-1 的多重比较结果

本例中，首先使用 read.csv()函数将存储水稻不同施肥处理下稻谷产量数据的文件"E:/RinBio/Chapt5/Example5_1.csv"读取到数据框 example5_1 中；接着使用 aov()函数以稻谷产量 y 为响应变量、施肥处理 trt 为因子变量构建方差分析模型，并将模型存储在 fit 中；然后使用 summary()函数输出方差分析模型的结果，初步判断不同施肥处理对稻谷产量是否有显著影响，输出结果涵盖了变异来源、自由度(Df)、平方和(Sum Sq)、均方(Mean Sq)、F 值(F Value)及概率(Pr)。其中，变异来源包括处理间变异和误差，从结果中可以得知，处理间的自由度为 4，平方和为 301，均方为 75.3，误差的自由度为 15，平方和为 101，均方为 6.7，5 个处理间水稻产量有极显著差异($F=11.2$，$p<0.01$)，* 表示在 0.05的水平上显著，** 表示在 0.01 的水平上显著，*** 表示在 0.001 的水平上显著。

由于处理间存在极显著差异，故需进行多重比较，利用 agricolae 包中的 LSD.test()函数实现 LSD 法多重比较。其中，$ statistics 返回多重比较的统计量，包括误差均方、自由度、t 值等；$ parameters 返回多重比较的相应参数值，包括测验方法、矫正类型、处理的名称、数量和显著水平 α；$ means 返回各处理的均值、标准差、样本容量、置信上限和置信下限、最小值、最大值和四分位数；$ group 返回多重比较的字母法标注结果，字母标注中有相同字母表明处理间差异不显著，不同字母表示处理间差异显著。

最后，使用 plot()函数绘制 LSD 检验结果图，直观展示不同施肥处理组的均值及分组情况，以判断各处理平均数的差异显著性，如图 5-2 所示。结果表明，处理 D(施尿素)产量最高(31.5)，显著高于处理 A，B，E，但与处理 C 差异不显著；处理 E 产量最低(20.0)，显著低于其他 4 个处理，因此处理 D 效果最好，其次为 C。

5.3　两向分组资料的方差分析

若试验资料分 k 行 n 列，则该资料为具 kn 个观察值的两向分组资料，或称交叉分组资料，或二因素无重复试验资料。

例 5－2　将一种生长激素配成 M_1，M_2，M_3，M_4，M_5 五种浓度，并用 H_1，H_2，H_3 三种时间浸渍某大豆品种的种子，45 天后得各处理每一植株的平均干物重（克）于表 5－3。试作方差分析并进行多重比较。

表 5－3　某激素对大豆干重的影响

M_i	H_j		
	H_1	H_2	H_3
M_1	13	14	14
M_2	12	12	13
M_3	3	3	3
M_4	10	9	10
M_5	2	5	4

在 Excel 录入数据时，按照数据框的方式录入 3 列数据，weight 表示植株的干物重，conc 表示生长激素的 5 种浓度，time 表示 3 种浸渍时间，将其保存为 Example5_2.csv（如图 5－3）。

	A	B	C
1	weight	conc	time
2	13	m1	h1
3	12	m2	h1
4	3	m3	h1
5	10	m4	h1
6	2	m5	h1
7	14	m1	h2
8	12	m2	h2
9	3	m3	h2
10	9	m4	h2
11	5	m5	h2
12	14	m1	h3
13	13	m2	h3
14	3	m3	h3
15	10	m4	h3
16	4	m5	h3

图 5－3　例 5－2 的数据录入格式

代码 5－2　两向分组资料的方差分析

```
> example5_2 <- read.csv("E:/RinBio/Chapt5/Example5_2.csv", header = TRUE)
> fit <- aov(weight~conc + time, data = example5_2)
> summary(fit)
            Df Sum Sq Mean Sq F value   Pr(> F)
conc         4 289.07   72.27 117.189 3.91e - 07 ***
time         2   1.73    0.87   1.405       0.3
Residuals    8   4.93    0.62
 - - -
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> library(agricolae)
> mc_e5_2 <- LSD.test(fit,"conc")
> mc_e5_2$groups
       weight groups
m1 13.666667      a
m2 12.333333      a
m4  9.666667      b
m5  3.666667      c
m3  3.000000      c
> plot(mc_e5_2)
```

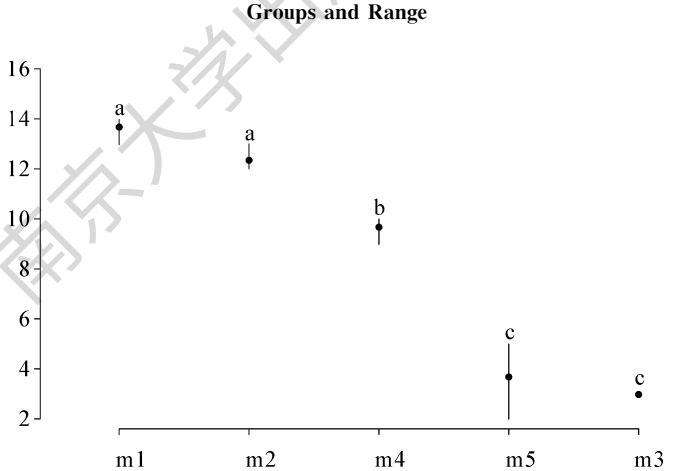

图 5－4　例 5－2 的多重比较结果

本例中，首先使用 read.csv()函数从"E:/RinBio/Chapt5/Example5_2.csv"文件读取数据到 example5_2 数据框；接着使用 aov()函数以植株平均干物重 weight 为响应变量，生长激素浓度 conc 和浸渍时间 time 为因子变量构建方差分析模型并存储在 fit 中；通过 summary()函数输出方差分析模型结果，判断各因子对干物重是否有显著影响。结果显示，浓度间的 F 测验差异为极显著($F=117.19$，$p<0.01$)，时间间的 F 测验差异为不显

著($F=1.41$, $p<0.01$)。

因此,有必要对不同浓度进一步开展多重比较分析,以明确各浓度水平间的具体差异。在此过程中,调用 LSD.test()函数执行多重比较,$ group 返回多重比较的字母法标注结果,使用 plot()函数绘制 LSD 检验结果图,直观展示不同浓度处理组均值及分组情况,如图 5-4 所示。通过对 5 种浓度的多重比较分析发现,m1 浓度处理下植株干物重最高,但与 m2 浓度处理间差异不显著,由此可判定 m1 和 m2 均为较优的浓度处理方案。

5.4　二因素完全随机化试验资料的方差分析

该类试验资料又被称为有重复观察值的两向分组试验资料,该类资料设有 A、B 两个试验因素,A 因素有 a 个水平,B 因素有 b 个水平,共有 ab 个处理组合,每一组合有 n 个观察值,则该资料有 abn 个观察值。

例 5-3　施用 A_1、A_2、A_3 3 种肥料于 B_1、B_2、B_3 3 种土壤,以小麦为指示作物,每处理组合种 3 盆,得产量结果(g)于表 5-4。试作方差分析(盖钧镒主编《试验统计方法》122 页例 6.14)。

表 5-4　二因素有重复观察值数据资料

肥料种类(A)	土壤种类(B)		
	B_1(油砂)	B_2(二合)	B_2(白僵)
A_1	21.4	19.6	17.6
	21.2	18.8	16.6
	20.1	16.4	17.5
A_2	12.0	13.0	13.3
	14.2	13.7	14.0
	12.1	12.0	13.9
A_3	12.8	14.2	12.0
	13.8	13.6	14.6
	13.7	13.3	14.0

在 Excel 录入数据时,录入 3 列数据,yield 表示植株的干物重,fert 表示 3 种肥料,soil 表示 3 种土壤,将其保存为 Example5_3.csv(如图 5-5)。

	A	B	C
1	yield	fert	soil
2	21.4	A1	B1
3	21.2	A1	B1
4	20.1	A1	B1
5	12	A2	B1
6	14.2	A2	B1
7	12.1	A2	B1
8	12.8	A3	B1
9	13.8	A3	B1
10	13.7	A3	B1
11	19.6	A1	B2
12	18.8	A1	B2
13	16.4	A1	B2
14	13	A2	B2

	A	B	C
15	13.7	A2	B2
16	12	A2	B2
17	14.2	A3	B2
18	13.6	A3	B2
19	13.3	A3	B2
20	17.6	A1	B3
21	16.6	A1	B3
22	17.5	A1	B3
23	13.3	A2	B3
24	14	A2	B3
25	13.9	A2	B3
26	12	A3	B3
27	14.6	A3	B3
28	14	A3	B3

图 5－5　例 5－3 的数据录入格式

代码 5－3　二因素完全随机化试验资料的方差分析

```
> example5_3 <- read.csv("E:/RinBio/Chapt5/Example5_3.csv", header = TRUE)
> fit <- aov(yield~fert + soil + fert * soil, data = example5_3)
> summary(fit)
             Df Sum Sq Mean Sq F value   Pr(> F)
fert          2  179.38   89.69  96.672  2.36e-10 ***
soil          2   3.96    1.98   2.135   0.14728
fert:soil     4  19.24    4.81   5.185   0.00587 **
Residuals    18  16.70    0.93
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> library(agricolae)
> mc_fert <- LSD.test(fit,"fert",alpha = 0.01)
> mc_fert
$statistics
     MSerror Df     Mean       CV t.value      LSD
     0.9277778  18  15.16296  6.352401  2.87844  1.306992

$parameters
        test p.ajusted name.t ntr alpha
  Fisher-LSD       none   fert   3  0.01

$means
     yield       std r         se       LCL       UCL  Min  Max  Q25  Q50  Q75
A1  18.80000  1.890106  9  0.3210707  17.87582  19.72418  16.4  21.4  17.5  18.8  20.1
A2  13.13333  0.900000  9  0.3210707  12.20915  14.05752  12.0  14.2  12.1  13.3  13.9
A3  13.55556  0.777996  9  0.3210707  12.63137  14.47974  12.0  14.6  13.3  13.7  14.0
```

```
$ comparison
NULL

$ groups
      yield groups
A1  18.80000      a
A3  13.55556      b
A2  13.13333      b

attr(,"class")
[1] "group"
> plot(mc_fert)
```

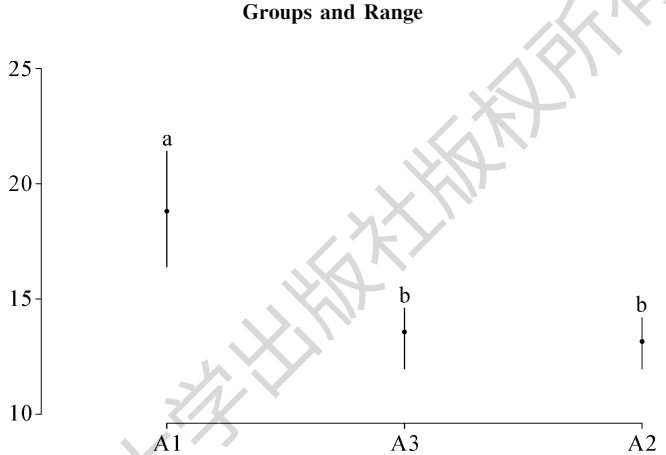

图 5－6　对肥料三个水平的多重比较图

本例中，首先使用 read.csv()函数从"E:/RinBio/Chapt5/Example5_3.csv"文件读取数据至 example5_3 数据框；接着用 aov()函数构建方差分析模型，以小麦产量 yield 为响应变量，肥料类型 fert、土壤类型 soil 以及它们的交互作用 fert * soil 为因子变量，模型存储在 fit 中；之后通过 summary()函数输出方差分析结果，以评估各因子对小麦产量的影响程度。结果显示，肥料(fert)间的 F 测验差异极显著($F=96.67$，$p<0.01$)，土壤(soil)间的 F 测验差异不显著($F=2.16$，$p=0.147>0.05$)，肥料与土壤间存在极显著的互作效应(fert:soil，$F=5.19$，$p<0.001$)。这表明肥料种类对小麦产量有极显著影响，土壤种类对小麦产量没有显著影响，但肥料与土壤间的交互效应对小麦产量有极显著影响。

为进一步明确不同肥料类型对小麦产量的具体影响，使用 LSD.test()函数对肥料类型 fert 进行 LSD 检验，并将检验结果存储于 mc_fert 对象中。通过输出 mc_fert 查看详细比较结果，同时运用 plot()函数绘制 LSD 检验结果图，直观展示不同肥料处理组的均值及其分组情况，如图 5－6 所示。检验结果显示了 3 种肥料处理下的平均小麦产量，其中 A_1 肥料处理下的平均小麦产量最高，达到 18.8，且与 A_2、A_3 肥料处理下的平均小麦产量间存在显著差异。这表明 A_1 肥料在提高小麦产量方面具有显著优势，是较为理想的肥料选择。

5.5 二因素随机区组试验资料的方差分析

设试验有 A 和 B 两个因素，A 因素有 a 个水平，B 因素有 b 个水平，随机区组设计，n 次重复，共有 ab 个处理组合，该试验共有 abn 个观察值。总变异可分解为处理间变异、区组变异与试验误差，而处理间变异又可分解为 A 因素变异、B 因素变异和 A×B 互作 3 部分。

例 5-4 将水稻的 3 个不同细胞质源的不育系（A_1、A_2、A_3）和 5 个恢复系（B_1、B_2、B_3、B_4、B_5）杂交，配成 15 个 F_1。采用随机区组设计，重复 2 次，小区计产面积 6 m^2。得结果见表 5-5。试对该资料进行方差分析，并对不育系和恢复系的各个水平间的平均数按照图基氏方法进行多重比较。

表 5-5 二因素随机区组试验资料

处理		区组	
		Ⅰ	Ⅱ
A_1	B_1	4.3	4.1
	B_2	4.9	4.8
	B_3	3.9	3.6
	B_4	4.8	4.0
	B_5	4.7	4.5
A_2	B_1	5.2	4.7
	B_2	5.0	5.2
	B_3	3.8	3.4
	B_4	4.9	4.8
	B_5	5.0	5.8
A_3	B_1	4.6	4.7
	B_2	4.4	4.2
	B_3	3.5	3.4
	B_4	3.4	3.6
	B_5	3.7	4.2

在 Excel 录入 4 列数据，block 表示区组，stline 表示不育系，reline 表示恢复系，yield 表示产量，将其保存为 Example5_4.csv（如图 5-7）。

	A	B	C	D
1	block	stline	reline	yield
2	R1	A1	B1	4.3
3	R2	A1	B1	4.1
4	R1	A2	B1	5.2
5	R2	A2	B1	4.7
6	R1	A3	B1	4.6
7	R2	A3	B1	4.7
8	R1	A1	B2	4.9
9	R2	A1	B2	4.8
10	R1	A2	B2	5
11	R2	A2	B2	5.2
12	R1	A3	B2	4.4
13	R2	A3	B2	4.2
14	R1	A1	B3	3.9
15	R2	A1	B3	3.6
16	R1	A2	B3	3.8

	A	B	C	D
17	R2	A2	B3	3.4
18	R1	A3	B3	3.5
19	R2	A3	B3	3.4
20	R1	A1	B4	4.8
21	R2	A1	B4	4
22	R1	A2	B4	4.9
23	R2	A2	B4	4.8
24	R1	A3	B4	3.4
25	R2	A3	B4	3.6
26	R1	A1	B5	4.7
27	R2	A1	B5	4.5
28	R1	A2	B5	5
29	R2	A2	B5	5.8
30	R1	A3	B5	3.7
31	R2	A3	B5	4.2
32				

图 5－7　例 5－4 的数据录入格式

代码 5－4　二因素随机区组试验资料的方差分析

```
> example5_4 <- read.csv("E:/RinBio/Chapt5/Example5_4.csv", header = TRUE)
> fit <- aov(yield~block + stline + reline + stline:reline, example5_4)
> summary(fit)
              Df Sum Sq Mean Sq F value    Pr(> F)
block          1  0.040  0.0403   0.516    0.4844
stline         2  3.282  1.6410  20.987   6.12e-05 ***
reline         4  5.298  1.3245  16.939   2.94e-05 ***
stline:reline  8  2.048  0.2560   3.274    0.0253 *
Residuals     14 1.095  0.0782
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> library(agricolae)
> mc_stline <- LSD.test(fit,"stline")
> mc_stline$groups
   yield groups
A2  4.78      a
A1  4.36      b
A3  3.97      c
> mc_retline <- LSD.test(fit,"reline")
> mc_retline$groups
   yield groups
B2  4.75      a
B5  4.65      a
B1  4.60      a
B4  4.25      b
B3  3.60      c
```

本例中，首先使用 read.csv()函数从"E:/RinBio/Chapt5/Example5_4.csv"文件读取数据到 example5_4 数据框；接着使用 aov()函数构建方差分析模型，以 F_1 产量 yield 为响应变量，区组 block、不育系 stline、恢复系 reline 以及不育系和恢复系的交互作用 stline:reline 为因子变量，模型存储在 fit 中；然后通过 summary()函数输出方差分析结果，判断各因子及其交互作用对产量是否有显著影响。结果可知，区组间(block)差异不显著($F=0.52$，$p=0.484$)，不育系(stline)间的 F 测验为极显著($F=20.99$，$p<0.001$)，恢复系(reline)间的 F 测验为极显著($F=16.94$，$p<0.001$)；不育系和恢复系间存在显著的互作效应(stline:reline，$F=3.27$，$p<0.05$)，表明不育系和恢复系均对于水稻产量具有极显著影响，不育系与恢复系的互作也对于水稻产量具有显著影响。

为了进一步明确不同不育系和恢复系对水稻产量的具体影响，并使用 LSD.test()函数分别对不育系 stline 和恢复系 reline 进行 LSD 检验，结果存储于 mc_stline 和 mc_retline 对象中，通过查看 mc_stline$groups 和 mc_retline$group 获取不育系各水平分组信息，以此实现对不育系和恢复系各水平平均数的多重比较。结果表明有 A_2 不育系参加的各杂交种的平均产量最高，并与 A_1、A_3 不育系有显著的差异。有 B_2 恢复系参加的各杂交种的平均产量最高，但与 B_5、B_1 恢复系无显著差异；而 B_2、B_5、B_1 的平均产量均显著高于 B_4；B_4 的平均产量则显著高于 B_3。这些结果为选择优良的不育系和恢复系组合提供了科学依据，有助于提高水稻产量。

5.6 系统分组试验资料的方差分析

如果试验资料分若干个组，每个组又分若干个亚组，每个亚组又分若干个小亚组，每个小亚组又分若干个小小亚组，如此一直下去，直至最后每一个小小亚组具有若干个观察值，这种资料叫作系统分组资料。系统分组资料像一棵倒着的“树”，越向前分，枝杈越多，而观察值则是这棵“树”的“叶片”。

最简单的系统分组资料是二级系统分组资料，它具 l 个组，每个组具 m 个亚组，每个亚组具 n 个观察值，共有 lmn 个观察值。

例 5-5 研究 A、B、C 和 D 这四种水生蔬菜对砷污染的“抗性”，每种蔬菜种 3 盆，每盆 5 个植株。生长期间施用一次有机砷农药，在收获时对每一植株的砷含量作一次分析，得表 5-6 的结果。试分析四种供试蔬菜对砷污染抗性的差异显著性。(莫惠栋著《农业试验统计(第二版)》183 页例 8.10)

表 5-6 例 5-5 中数据

品种	盆号	砷含量				
1	1	0.7	0.6	0.9	0.5	0.6
	2	0.9	0.9	0.7	1.1	0.7
	3	0.8	0.6	0.9	1	0.8

续　表

品种	盆号	砷含量				
2	4	1.2	1.4	1.6	1.2	1.5
	5	1.1	0.9	1.3	1.2	1
	6	1.5	1.4	0.9	1.3	1.6
3	7	0.6	0.6	0.8	0.9	0.7
	8	0.5	0.8	0.9	1	0.6
	9	0.6	1.2	0.8	0.9	1
4	10	4.2	3.7	2.9	3.5	3.6
	11	2.9	3.5	3.8	3.1	3.5
	12	3.6	3.5	4	3.3	3.7

	A	B	C
1	cult	pot	res
2	c1	p1	0.7
3	c1	p1	0.6
4	c1	p1	0.9
5	c1	p1	0.5
6	c1	p1	0.6
7	c1	p2	0.9
8	c1	p2	0.9
9	c1	p2	0.7
10	c1	p2	1.1
11	c1	p2	0.7
12	c1	p3	0.8
13	c1	p3	0.6
14	c1	p3	0.9
15	c1	p3	1
16	c1	p3	0.8
17	c2	p4	1.2
18	c2	p4	1.4
19	c2	p4	1.6
20	c2	p4	1.2
21	c2	p4	1.5

	A	B	C
22	c2	p5	1.1
23	c2	p5	0.9
24	c2	p5	1.3
25	c2	p5	1.2
26	c2	p5	1
27	c2	p6	1.5
28	c2	p6	1.4
29	c2	p6	0.9
30	c2	p6	1.3
31	c2	p6	1.6
32	c3	p7	0.6
33	c3	p7	0.6
34	c3	p7	0.8
35	c3	p7	0.9
36	c3	p7	0.7
37	c3	p8	0.5
38	c3	p8	0.8
39	c3	p8	0.9
40	c3	p8	1
41	c3	p8	0.6
42	c3	p9	0.6

	A	B	C
43	c3	p9	1.2
44	c3	p9	0.8
45	c3	p9	0.9
46	c3	p9	1
47	c4	p10	4.2
48	c4	p10	3.7
49	c4	p10	2.9
50	c4	p10	3.5
51	c4	p10	3.6
52	c4	p11	2.9
53	c4	p11	3.5
54	c4	p11	3.8
55	c4	p11	3.1
56	c4	p11	3.5
57	c4	p12	3.6
58	c4	p12	3.5
59	c4	p12	4
60	c4	p12	3.3
61	c4	p12	3.7
62			
63			

图 5－8　例 5－5 的数据录入格式

代码 5－5　二级系统分组资料的方差分析

```
> example5_5 <- read.csv("E:/RinBio/Chapt5/Example5_5.csv", header = TRUE)
> fit1 <- aov(res~cult + pot, example5_5)
> summary(fit1)
             Df Sum Sq Mean Sq F value Pr(> F)
cult          3  76.74  25.580  423.389 < 2e - 16 ***
pot           8  0.63   0.078   1.2970.268
Residuals    48 2.90   0.060
 - - -
```

```
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> fit2 <- aov(res~cult, example5_5)
> summary(fit2)
             Df Sum Sq Mean Sq F value Pr(> F)
cult          3  76.74  25.580  406.2 < 2e - 16 ***
Residuals    56 3.53   0.063
 - - -
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> library(agricolae)
> mc_e5_5 <- LSD.test(fit2,"cult",MSerror = 0.063)
> mc_e5_5 $ groups
          res groups
c4 3.5200000       a
c2 1.2733333       b
c3 0.7933333       c
c1 0.7800000       c
```

本例中，首先使用 read.csv()函数从"E:/RinBio/Chapt5/Example5_5.csv"文件读取数据到 example5_5 数据框；接着构建第一个方差分析模型 fit1，以蔬菜抗性指标 res 为响应变量，蔬菜品种 cult 和盆号 pot 为因子变量，通过 summary(fit1)查看该模型方差分析结果；然后构建第二个方差分析模型 fit2，只将蔬菜品种 cult 作为因子变量，同样用 summary(fit2)查看其方差分析结果；之后加载 agricolae 包，利用 LSD.test()函数对模型 fit2 中的蔬菜品种 cult 进行最小显著差异(LSD)检验，指定误差均方 MSerror 为 0.063，结果存于 mc_e5_5；最后通过 mc_e5_5 $ groups 查看各蔬菜品种处理组的分组信息，以此判断四种供试蔬菜对砷污染抗性的差异显著性。

在此过程中，首先将变异分解为品种间和品种内不同盆间，根据代码 5-5 的计算结果可知，品种内不同盆间(pot)差异不显著($F=1.297$，$p=0.268$)，因而处理间差异的测验可以用处理内盆间和试验误差的合并均方作为被比量，结果表明处理间差异达到极显著水平($F=406.2$，$p<0.001$)。在进行多重比较时，MSerror 需设为 0.063。若处理内盆间差异显著时，则只能用处理内盆间的均方为误差项均方进行测验。通过 LSD 法多重比较分析，结果显示品种 1 和品种 3 对有机砷的“抗性”最强，其体内积累的砷含量最低，并且与品种 2 和品种 4 之间存在显著差异。这一结论为评估不同蔬菜品种对砷污染的抗性提供了科学依据，有助于在实际应用中选择适宜的蔬菜品种以应对砷污染问题。

5.7 品种区域试验资料的方差分析

新育成的品种或品系，在推广于大面积生产之前，必须经过区域化试验。区域化试验中需要研究的主要因素有：

(1) 品种效应:这是供试品种的产量和品质等的主效,属固定型;

(2) 地点效应:这是地点间的土壤类型、耕作制度、管理方法等可以预知的环境差异的效应,一般亦属固定型;

(3) 年份效应:这是不同年份的温度、雨量、日照天数、偶然性灾害等难以预知的环境差异的效应,一般属于随机型;

(4) 品种×地点互作:这是品种对于可预知的环境差异是否具有特殊的适应性,一般属于固定型;

(5) 品种×年份互作:这是品种对于难以预知的环境差异是否有特殊的适应性,一般属于随机型;

(6) 品种×地点×年份互作:这是品种×地点的互作是否随难以预知的环境差异而有不同,一般属随机型。

广为应用的品种必须具有品种主效大而互作小,这种品种在可预知或难以预知的环境变异下,能通过其本身基因系统的调节,在较高水平上保持相对稳定。互作大的品种则只能供特定地区或特定气候条件下利用,不能广泛应用。

AMMI(Additive main effects and multiplicative interaction)模型是农业领域常用的一种分析基因型与环境互作效应的统计方法,它可以有效地解析基因型在不同环境下的表现,为品种评价和试验点评价提供重要依据。在 R 中,agricolae 包中的 AMMI()函数可对品种区域试验资料进行方差分析,通过从加性模型的残差中分离模型误差和干扰,提高估计的准确度,并且借助于双标图可以更直观地描述和分析基因型与环境互作模式。AMMI()函数返回一个包含 AMMI 模型分析结果的对象,该对象通常包含以下组成部分:① ANOVA 表,用于展示方差分析的结果,包括环境、基因型、环境×基因型互作以及残差等部分的方差、均方、F 值和 P 值。② 主成分分析结果,包括各主成分的解释方差比例、累计解释方差比例、主成分得分等。③ 其他相关信息,如基因型和环境的均值、双标图数据等。其一般的使用格式为:

```
AMMI(ENV, GEN, REP, Y, console = FALSE, PC = FALSE)
```

ENV:环境因子,通常是一个因子(factor)变量,表示试验中的不同环境(如地点、年份等)。

GEN:基因型因子,也是一个因子变量,表示试验中的不同基因型(或品种)。

REP:重复因子,用于指定在每个环境－基因型组合中的重复次数或重复编号,通常也是一个因子变量。

Y:响应变量,通常是一个数值型变量,表示在不同环境－基因型组合下的观测值(如产量、性状值等)。

console:逻辑值(TRUE 或 FALSE),用于控制是否在控制台上输出主成分分析的结果。如果设置为 TRUE,则会在控制台上打印出主成分分析的相关统计量。

PC:逻辑值(TRUE 或 FALSE),用于控制是否计算和返回主成分分析的结果。如果设置为 TRUE,则函数会返回包含主成分分析结果的对象。

例 5－6　为了加强抗条纹叶枯病粳稻新品种的选育和改良,扬州大学农学院与江苏

省常州市武进区农业科学研究所进行了合作研究，在江苏扬州、常州、盐城、镇江等地开展了品种区域试验。每个品种在各自的小区内种植 4 行，每行 10 株，重复 2 次，并进行随机区组排列，25 个品种的产量如表 5－7 所示。试对该结果进行方差分析。

表 5－7　例 5－6 中的数据

序号	品种	扬州		常州		盐城		镇江	
		区组 1	区组 2	区组 1	区组 2	区组 1	区组 2	区组 1	区组 2
v1	武运粳 8 号	11.76	10.80	11.92	10.88	11.84	11.36	11.28	11.28
v2	扬粳 9538	10.24	9.84	10.08	9.60	10.40	10.00	9.92	10.00
v3	南粳 39	9.68	9.44	9.52	9.12	9.76	9.36	9.28	9.36
v4	武运粳 11	11.04	10.64	10.72	10.48	10.96	10.56	10.72	10.80
v5	镇稻 8 号	10.32	10.16	10.24	9.92	10.40	10.08	9.84	10.24
v6	南粳 40	10.16	10.00	10.08	9.84	10.24	9.92	9.84	9.84
v7	华粳 1 号	9.84	9.60	9.68	9.36	9.76	9.52	9.44	9.84
v8	广陵香粳	10.00	9.84	9.92	9.60	10.08	9.68	9.60	10.00
v9	通育粳 1 号	10.56	10.40	10.48	10.16	10.64	10.24	10.16	10.16
v10	南粳 41	10.72	10.56	10.64	10.32	10.80	10.40	10.32	10.00
v11	淮稻 7 号	10.80	10.64	10.72	10.40	10.88	10.48	10.40	10.40
v12	华粳 3 号	9.76	9.52	9.68	9.20	9.76	9.44	9.36	10.00
v13	扬辐粳 7 号	10.16	10.00	10.08	9.76	10.24	9.84	9.76	9.84
v14	盐粳 2 号	8.96	8.64	8.88	8.48	8.96	8.56	8.72	9.28
v15	泗稻 8 号	9.36	9.12	9.28	8.88	9.36	8.96	8.88	10.00
v16	镇稻 1 号	9.12	8.80	9.04	8.64	9.12	8.72	8.80	9.28
v17	武育粳 3 号	10.16	10.00	10.08	9.76	10.24	9.84	9.76	9.68
v18	盐粳 4 号	10.24	10.00	10.08	9.76	10.24	9.84	9.76	9.76
v19	武育粳 4 号	10.00	9.76	9.84	9.52	10.00	9.60	9.52	10.24
v20	泗稻 10 号	9.76	9.52	9.68	9.36	9.76	9.44	9.36	9.52
v21	镇稻 4 号	9.76	9.52	9.68	9.36	9.76	9.44	9.36	9.20
v22	盐粳 5 号	9.60	9.36	9.52	9.20	9.60	9.28	9.28	9.12
v23	通粳 109	9.76	9.52	9.68	9.36	9.76	9.44	9.36	9.36
v24	香粳 111	9.68	9.44	9.60	9.20	9.68	9.36	9.28	10.40
v25	镇香粳 5 号	9.92	9.68	9.84	9.52	9.92	9.60	9.52	10.56

在 Excel 录入 4 列数据，y 表示产量，g 表示品种，e 表示地点，r 表示区组，将其保存为 Example5_6.csv(如图 5－9)。

	A	B	C	D
1	y	g	e	r
2	11.76	v1	u1	r1
3	10.24	v2	u1	r1
4	9.68	v3	u1	r1
5	11.04	v4	u1	r1
6	10.32	v5	u1	r1
7	10.16	v6	u1	r1
8	9.84	v7	u1	r1
9	10.00	v8	u1	r1
10	10.56	v9	u1	r1
11	10.72	v10	u1	r1
12	10.80	v11	u1	r1
13	9.76	v12	u1	r1
14	10.16	v13	u1	r1
15	8.96	v14	u1	r1
16	9.36	v15	u1	r1
17	9.12	v16	u1	r1
18	10.16	v17	u1	r1
19	10.24	v18	u1	r1
20	10.00	v19	u1	r1
21	9.76	v20	u1	r1
22	9.76	v21	u1	r1
23	9.60	v22	u1	r1
24	9.76	v23	u1	r1
25	9.68	v24	u1	r1
26	9.92	v25	u1	r1

	A	B	C	D
25	9.68	v24	u1	r1
26	9.92	v25	u1	r1
27	10.80	v1	u1	r2
28	9.84	v2	u1	r2
29	9.44	v3	u1	r2
30	10.64	v4	u1	r2
31	10.16	v5	u1	r2
32	10.00	v6	u1	r2
33	9.60	v7	u1	r2
34	9.84	v8	u1	r2
35	10.40	v9	u1	r2
36	10.56	v10	u1	r2
37	10.64	v11	u1	r2
38	9.52	v12	u1	r2
39	10.00	v13	u1	r2
40	8.64	v14	u1	r2
41	9.12	v15	u1	r2
42	8.80	v16	u1	r2
43	10.00	v17	u1	r2
44	10.00	v18	u1	r2
45	9.76	v19	u1	r2
46	9.52	v20	u1	r2
47	9.52	v21	u1	r2
48	9.36	v22	u1	r2
49	9.52	v23	u1	r2
50	9.44	v24	u1	r2
51	9.68	v25	u1	r2

图 5－9　例 5－6 的部分数据录入格式

代码 5－6　品种区域试验资料的方差分析

```
> example5_6 <- read.csv("E:/RinBio/Chapt5/Example5_6.csv", header = TRUE)
> library(agricolae)
> model <- AMMI(example5_6$e, example5_6$g, example5_6$r, example5_6$y, PC = TRUE)
> summary(model)
          Length    Class          Mode
ANOVA     5         anova          list
genXenv   100       by             numeric
analysis  7         data.frame     list
means     4         data.frame     list
biplot    5         data.frame     list
PC        7         princomp       list
> model$ANOVA
Analysis of Variance Table

Response: Y
           Df    Sum    Sq Mean    Sq F value    Pr(> F)
ENV        3     0.861  0.28715    0.2227        0.8763
REP(ENV)   4     5.158  1.28941    47.3908       < 2e - 16 ***
GEN        24    61.642 2.56840    94.3987       < 2e - 16 ***
ENV:GEN    72    1.894  0.02630    0.9667        0.5566
Residuals  96    2.612  0.02721
- - -
```

```
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> model$analysis
     percent     acum Df     Sum.Sq     Mean.Sq F.value     Pr.F
PC1     86.9     86.9     26     1.646079     0.063311     2.33     0.0016
PC2     9.0     95.9     24     0.170779     0.007116     0.26     0.9998
PC3     4.1     100.0     22     0.076902     0.003496     0.13     1.0000
> plot(model, 0, 1)
```

本例中，利用 agricolae 包中的 AMMI 函数进行品种区域试验资料的方差分析，以环境 example5_6$e、品种 example5_6$g、区组 example5_6$r 和产量 example5_6$y 作为参数，同时设置 PC＝TRUE 表示计算主成分，将分析结果存储在 model 中。通过调用 model$ANOVA，我们可以查看方差分析的结果。结果显示，不同品种间的主效差异极为显著（F＝94.398 7，p＜0.01），品种×地点互作效应不显著（F＝0.966 7，p＝0.556 6），地点主效不显著（F＝0.222 7，p＝0.876 3）。进一步查看 model$analysis，其中呈现了互作效应的主成分分析结果，PC1（第一主成分）达到了显著水平，且其解释了 86.9%的互作效应平方和。

为了更直观地展示分析结果，使用 plot（model，0，1）绘制 AMMI 分析的相关图形。图 5－10 显示的是第一主成分对产量的双标图，该图中横轴通常代表品种的观测值（如产量），而纵轴则代表第一主成分（PC1）的分数。在该图中，以 PC1 值为 0 作一条水平线，以所有品种产量的平均值作垂线，将图表划分为 4 个区域。对供试品种而言，在水平方向上横坐标值越大，代表产量越高；在垂直方向上距水平线越近，则稳产性越好。结果显示，丰产性最高的品种是 v1，即武运粳 8 号，可将其选为抗性改良对象。

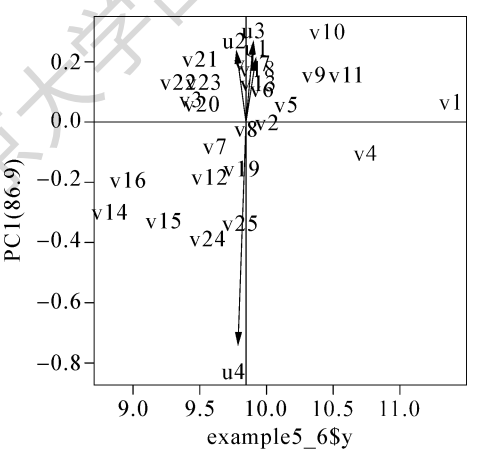

图 5－10　AMMI 模型双标图（第一主成分对产量）

5.8　裂区试验资料的方差分析

裂区试验（Split-Plot Experiment）是一种特殊的试验设计，用于研究两个或多个因素

之间的交互作用，其中一个因素被视为“主区”(Whole-Plot Factor)，另一个因素被视为“裂区”(Sub-Plot Factor)。主区因素通常代表难以控制或改变的环境因素(如土壤类型、地理位置等)，而裂区因素则代表容易改变或控制的处理因素(如施肥量、品种等)。设计这类试验时，首先将需要较大试验单元(这些单元叫主区)的处理随机地排入区组；然后将每一主区皆分成若干个亚试验单元(这些单元叫裂区)，把各裂区处理随机地排入裂区，处理数应等于每一主区的亚试验单元数，以使每一主区中都包含一套裂区处理。

例 5－7　以 4 个山芋品种做翻蔓试验。品种编号为 A_1、A_2(短蔓种)、A_3、A_4(长蔓种)，翻蔓次数编号为 B_1(不翻蔓)、B_2(翻蔓 3 次)、B_3(翻蔓 6 次)。以品种为主区因素、翻蔓为裂区因素，二裂式裂区设计，重复 3 次，裂区计产面积 40 平方米。其田间排列和裂区芋干产量(公斤)列于图 5－11，试进行方差分析。(莫惠栋著《农业试验统计(第二版)》250 页例 9.6)

	A_3	A_2	A_4	A_1
区组Ⅰ	B_2 20	B_3 20	B_3 18	B_1 24
	B_1 18	B_1 24	B_2 22	B_3 20
	B_3 18	B_2 24	B_1 21	B_2 22
	A_2	A_3	A_1	A_4
区组Ⅱ	B_3 22	B_3 19	B_1 23	B_1 20
	B_2 22	B_1 20	B_3 21	B_2 19
	B_1 26	B_2 20	B_2 20	B_3 16
	A_1	A_4	A_3	A_2
区组Ⅲ	B_1 24	B_2 19	B_1 18	B_3 23
	B_2 24	B_1 22	B_3 18	B_2 22
	B_3 23	B_3 18	B_2 18	B_1 25

图 5－11　山芋二裂式试验的田间排列和产量

	A	B	C	D
1	block	A	B	yield
2	r1	a1	b1	24
3	r1	a1	b2	22
4	r1	a1	b3	20
5	r1	a2	b1	24
6	r1	a2	b2	24
7	r1	a2	b3	20
8	r1	a3	b1	18
9	r1	a3	b2	20
10	r1	a3	b3	18
11	r1	a4	b1	21
12	r1	a4	b2	22
13	r1	a4	b3	18
14	r2	a1	b1	23
15	r2	a1	b2	20
16	r2	a1	b3	21
17	r2	a2	b1	26
18	r2	a2	b2	22
19	r2	a2	b3	22
20	r2	a3	b1	20
21	r2	a3	b2	20
22	r2	a3	b3	19
23	r2	a4	b1	20
24	r2	a4	b2	19
25	r2	a4	b3	16
26	r3	a1	b1	24
27	r3	a1	b2	24
28	r3	a1	b3	23
29	r3	a2	b1	25
30	r3	a2	b2	22
31	r3	a2	b3	23
32	r3	a3	b1	18
33	r3	a3	b2	18
34	r3	a3	b3	18
35	r3	a4	b1	22
36	r3	a4	b2	19
37	r3	a4	b3	18
38				

图 5－12　例 5－7 的数据录入格式

代码 5-7　裂区试验资料的方差分析

```
> example5_7 <- read.csv("E:/RinBio/Chapt5/Example5_7.csv", header = TRUE)
> library(agricolae)
> fit <- aov(yield~A*B+Error(block/A), data = example5_7)
> summary(fit)

Error: block
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  2    1.5    0.75

Error: block:A
            Df Sum Sq Mean Sq F value  Pr(>F)
A            3  122.1   40.69    13.2 0.00473 **
Residuals    6   18.5    3.08
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
          Df Sum Sq Mean Sq F value   Pr(>F)
B          2  35.17  17.583  14.552 0.000251 ***
A:B        6  14.17   2.361   1.954 0.133157
Residuals 16  19.33   1.208
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> mc_e5_7A <- with(example5_7, LSD.test(yield, A, DFerror = 6, MSerror = 3.08))
> mc_e5_7A$groups
      yield groups
a2 23.11111      a
a1 22.33333      a
a4 19.44444      b
a3 18.77778      b
> mc_e5_7B <- with(example5_7, LSD.test(yield, B, DFerror = 16, MSerror = 1.208))
> mc_e5_7B$groups
      yield groups
b1 22.08333      a
b2 21.00000      b
b3 19.66667      c
```

本例中，首先使用 read.csv()函数从"E:/RinBio/Chapt5/Example5_7.csv"文件读取数据到 example5_7 数据框中；加载 agricolae 包，使用 aov()函数构建方差分析模型 fit，以芋干产量 yield 为响应变量，品种 A 和翻蔓次数 B 及其交互作用 A*B 为因子，同时考

虑区组 block 和品种的嵌套关系 Error(block/A)。通过 summary(fit)查看方差分析模型的结果，计算结果中，“Error：block”反映区组的方差分析结果，“Error：block：A”反映主区的方差分析结果，包括品种间差异和主区误差，“Error：Within”反映裂区的方差分析结果，包括翻蔓次数间差异和裂区误差。结果表明，不同品种间山芋的产量存在显著差异（$F=13.2$，$p<0.01$），不同翻蔓次数间的山芋产量存在显著差异（$F=14.55$，$p<0.01$）；品种和翻蔓次数之间不存在交互作用（$F=1.95$，$p=0.133\ 16$）。在对品种进行多重比较时，采用主区误差的方差作为误差方差；而在对翻蔓次数进行多重比较时，则采用裂区误差项的方差作为误差方差。使用 with()函数结合 LSD.test()函数对品种 A 进行 LSD 检验，指定误差自由度 DFerror 为 6，误差均方 MSerror 为 3.08，结果存于 mc_e5_7A，并查看 mc_e5_7A $ group 获取品种分组信息；对翻蔓次数 B 进行 LSD 检验，指定误差自由度 DFerror 为 16，误差均方 MSerror 为 1.208，结果存于 mc_e5_7B，查看 mc_e5_7B $ group 获取翻蔓次数分组信息，以此完成对山芋品种翻蔓试验结果的方差分析和多重比较。结果说明，品种 A_2 产量最高，但与 A_1 无显著差异。不翻蔓的芋干产量最高，翻蔓 3 次和翻蔓 6 次的，芋干产量皆显著降低。

在进行裂区试验时，由于同一主区试验空间非处理因素的一致性通常要比不同主区更好，所以一般情况下品种×区组的方差要比误差项的方差为大，如果品种×区组的方差不比误差项的方差大，则可以按照二因素随机区组试验资料进行处理。

5.9　协方差分析

在进行一般方差分析时，要求除研究的因素外，应该保证其他条件的一致或者接近一致。然而在实际进行的试验中，很难对这一点进行控制，那么这时候就需要用到协方差分析。协方差分析是将回归分析与方差分析结合起来的一种统计分析方法，其主要功用是对试验误差进行统计控制。例如研究几种配合饲料对猪的增重效果，希望供试仔猪的初始体重都相同，这很难达到，仔猪的初始体重不同，将影响到猪的增重。这时若仔猪的初始体重与增重之间存在线性回归关系，就可利用协方差分析将增重矫正为初始体重相同时的增重，消除初始体重对增重的影响。

协方差分析从影响因素和变量的个数分为单因素协方差分析、多因素协方差分析和多变量协方差分析等；从试验设计的方式分为完全随机设计协方差分析、随机区组设计协方差分析和析因协方差分析等。我们仅采用实例对单因素协方差分析进行初步介绍。

R 语言中，HH 包中的函数 ancova()提供了协方差分析的计算，其一般的使用格式为：

```
ancova(formula, data = NULL, subset, na.action, ...)
```

formula：一个模型公式，用于指定协方差分析中的因变量、自变量（因子）和协变量。公式的形式通常为“因变量～自变量＋协变量”，如果有多个自变量或协变量，可以使用“＋”将它们连接起来。

data：一个可选的数据框，用于指定包含因变量、自变量和协变量的数据集。如果数

据已经以数据框的形式存在于 R 环境中，可以通过此参数指定。

subset：一个可选的逻辑表达式，用于从数据集中选择子集进行分析。只有当数据框通过 data 参数指定时，此参数才有效。

na.action：一个可选的函数，用于处理数据中的缺失值(NA)。默认情况下，ancova()函数会使用 na.omit()函数删除包含缺失值的观测。

...：其他可选参数，这些参数可能依赖于 HH 包的具体版本和底层实现，用于进一步定制协方差分析的过程。

ancova()函数返回值是一个列表对象，主要包括：anova 是方差分析表，包含各因素和协变量的自由度、平方和、均方、F 值及 p 值，可判断因素和协变量对响应变量的影响是否显著；coefficients 为模型的系数估计值，体现协变量和因素水平对响应变量的作用大小；residuals 是模型的残差，即观测值与模型预测值的差异，可用于评估模型拟合效果；fitted.values 是模型的拟合值，是根据模型对响应变量的预测结果；call 记录调用 ancova()函数时的具体语句，方便回顾分析过程；terms 展示协方差分析模型的公式结构，明确分析所涉及的因素和协变量；contrasts 给出因素的对比编码方式；xlevels 记录因素的水平信息。

例 5－8 为研究 A、B、C 三种肥料对于苹果的增产效果，选了 24 株同龄的苹果树，第一年记下各树的产量(X，千克)，第二年将每种肥料随机施于 8 株苹果上，再记下其产量(Y，千克)，得结果于表 5－8。试进行协方差分析。(莫惠栋著《农业试验统计(第二版)》365 页例 11.1)

表 5－8 施用三种肥料前后的苹果产量

肥 料	观察值(X_{ij}，Y_{ij})							
A	X_{1j}	47	58	53	46	49	56	54	44
	Y_{1j}	54	66	63	51	56	66	61	50
B	X_{2j}	52	53	64	58	59	61	63	66
	Y_{2j}	54	53	67	62	62	63	64	69
C	X_{3j}	44	48	46	50	59	57	58	53
	X_{3j}	52	58	54	61	70	64	68	66

	A	B	C
1	x	y	f
2	47	54	A
3	58	66	A
4	53	63	A
5	46	51	A
6	49	56	A
7	56	66	A
8	54	61	A
9	44	50	A
10	52	54	B
11	53	53	B
12	64	67	B
13	58	62	B

	A	B	C
14	59	62	B
15	61	63	B
16	63	64	B
17	66	69	B
18	44	52	C
19	48	58	C
20	46	54	C
21	50	61	C
22	59	70	C
23	57	64	C
24	48	69	C
25	53	66	C
26			

图 5－13 例 5－8 的数据录入格式

代码 5-8　协方差分析

```
> example5_8 <- read.csv("Example5_8.csv", header = TRUE)
> aggregate(example5_8$y, by = list(example5_8$f), FUN = mean)
Group.1       x
1         A 58.375
2         B 61.750
3         C 61.750
> example5_8[, 3] <- as.factor(example5_8[, 3])
> library(HH)
> library(effects)
> ancova_result <- ancova(y~x + f, data = example5_8)
> ancova_result
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value     Pr(> F)
x          1  482.83  482.83  57.643  2.59e-07 ***
f          2  241.27  120.64  14.402  0.0001336 ***
Residuals 20 167.52   8.38
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
> f_effect <- effect("f", ancova_result)
> f_effect

 f effect
f
       A         B         C
61.42772  55.37119  65.07609
```

本例中，首先使用 read.csv()函数从"Example5_8.csv"文件读取数据到 example5_8 数据框；使用 aggregate()函数按照肥料类型 example5_8$f 对苹果产量 example5_8$y 分组求均值，得到三种肥料处理下（A，B，C）苹果产量的均值，分别为 58.375、61.750 和 61.750，然后将数据框的第三列转换为因子类型，以便后续分析。然后，加载 HH 和 effects 包，前者提供 ancova()函数进行协方差分析，后者用于效应分析。使用 ancova()函数构建协方差分析模型，以第二年苹果产量 y 为响应变量，第一年产量 x 为协变量，肥料类型 f 为分类因子，分析结果存储在 ancova_result 中，结果显示，基础生产力（X）的测验达显著水平（$F=57.643$，$p<0.01$），表明果树产量（Y）和前一年的基础生产力（X）显著相关。肥料对果树产量有显著影响（$F=14.402$，$p<0.01$）。最后利用 effects 包中的 effect()函数可以获取去除协变量效应后的组均值（调整的组均值），三个处理（A，B，C）的反应变量 Y 的去除协变量效应后的矫正均值分别为 61.42.7，55.371，65.076。除显示协方差分析结果外，还生

成了如图 5－14 所示的施肥之后产量(y)依施肥之前产量(x)和肥料类别(f)变化图，直观地呈现了不同肥料处理对苹果产量的影响及其与前一年产量的关系。

$$y \sim x + f$$

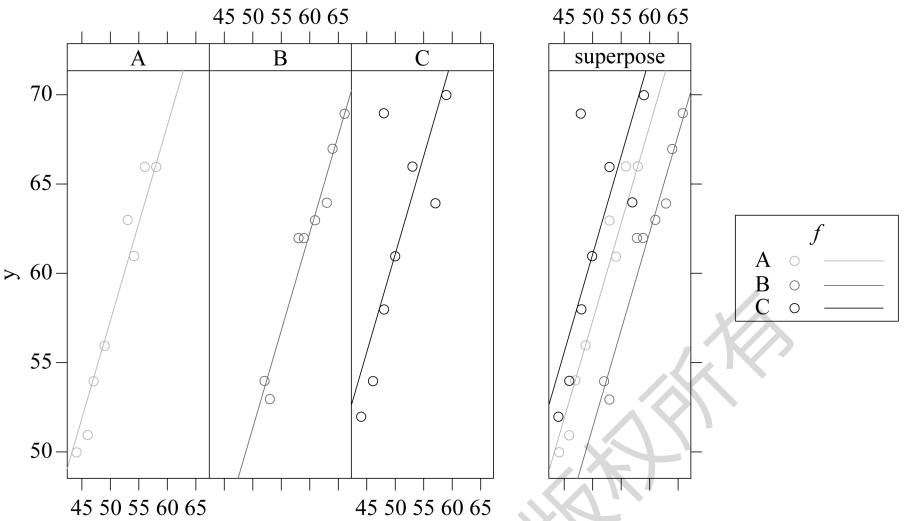

图 5－14　施肥之后产量(y)依施肥之前产量(x)和肥料类别(f)变化图

习 题

5－1　一般而言，两个均数的假设测验和多个平均数的假设测验，应分别用什么统计方法进行？

5－2　有 6 个不同水稻品种的纹枯病接种鉴定试验，每品种 5 个重复，纹枯病病级调查如下表，进行方差分析并给出多重比较结果。

品种名称	观察值				
品种 1	8.2	9.0	7.8	8.0	9.0
品种 2	7.5	6.2	6.6	7.5	5.6
品种 3	6.8	5.3	5.8	5.4	6.3
品种 4	4.9	4.7	5.8	5.1	4.6
品种 5	6.5	5.4	6.3	6.7	6.3
品种 6	3.6	3.9	4.0	3.2	3.4

5－3　将一种生长激素配成 M_1、M_2、M_3、M_4 四种浓度，处理 P_1、P_2、P_3 三种水稻种子，7 天后得各处理水稻的发芽率(%)如下表。试做方差分析并进行多重比较。

M_i	P_j		
	P_1	P_2	P_3
M_1	89.3	94.2	91.1
M_2	92.1	96.1	93.5
M_3	94.7	97.3	95.9
M_4	96.8	95.7	96.2

5－4　在温室内以 4 种培养液培养某植物，每种培养液培养 3 盆，每盆 4 株，全试验共有 12 盆按完全随机排列，管理条件相同，30 天后测定株高生长量(mm)，得结果列于下表，试做方差分析。

培养液(A)	盆号(B)	生长量/mm			
A_1	B_{11}	50	55	40	35
	B_{12}	35	35	30	40
	B_{13}	45	40	40	50
A_2	B_{21}	50	45	50	45
	B_{22}	55	60	50	50
	B_{23}	55	45	65	55
A_3	B_{31}	85	60	90	85
	B_{32}	65	70	80	65
	B_{33}	70	70	70	70
A_4	B_{41}	60	55	35	70
	B_{42}	60	85	45	75
	B_{43}	65	65	85	75

5－5　有一早稻二因素试验，A 因素为品种，分 A_1(早熟)、A_2(中熟)、A_3(迟熟)三个水平，B 因素为密度，分 B_1(16.5 cm×6.6 cm)，B_2(16.5 cm×9.9 cm)、B_3(16.5 cm×13.2 cm)三个水平，随机区组试验设计，重复 3 次，得结果如下表，试进行方差分析。

品种	密度	区组Ⅰ	区组Ⅱ	区组Ⅲ
A_1	B_1	8	8	8
	B_2	7	7	6
	B_3	6	5	6
A_2	B_1	9	9	8
	B_2	7	9	6
	B_3	8	7	6
A_3	B_1	7	7	6
	B_2	8	7	8
	B_3	10	9	9

5-6 在温室内以 4 种培养液培养某作物，每种 3 盆，每盆 5 株，一个月后测定其株高生长量(mm)，得结果于下表，试做方差分析。

品种	盆号	生长量/mm				
A	1	50	35	45	55	35
	2	55	35	40	40	30
	3	40	30	40	35	40
B	4	50	55	55	45	60
	5	45	60	45	50	50
	6	50	50	65	45	50
C	7	85	65	70	60	70
	8	60	70	70	90	80
	9	90	80	70	85	65
4	10	60	60	65	55	85
	11	55	85	65	35	45
	12	35	45	85	70	75

5-7 设一个水稻品种区域试验，包括对照种在内共有 5 个供试品种，在 4 个地点进行试验，每点每次试验均统一采用相同小区面积重复 3 次的随机区组设计，得产量($kg/33\ m^2$)数据结果列于下表。试做方差分析。

试点	品种	区组		
		Ⅰ	Ⅱ	Ⅲ
甲	A	19.7	31.4	29.6
	B	28.6	38.3	43.5
	C	20.3	27.5	32.6
	D	27.9	40.0	46.1
	E	22.3	30.8	31.1
乙	A	40.8	29.4	30.2
	B	44.4	34.9	33.9
	C	44.6	41.4	26.2
	D	39.8	39.2	29.1
	E	71.5	47.6	55.4

续　表

试点	品种	区组		
		Ⅰ	Ⅱ	Ⅲ
丙	A	34.7	29.1	35.1
	B	28.8	28.7	21.0
	C	29.8	38.4	28.0
	D	27.2	27.6	20.4
	E	43.0	32.7	32.0
丁	A	20.2	30.2	16.0
	B	13.2	20.5	9.6
	C	24.5	41.6	30.6
	D	19.0	18.4	24.6
	E	27.6	30.0	22.7

5－8　设有一小麦中耕次数(A)和施肥量(B)试验，主处理为 A，分 A_1、A_2、A_3 3 个水平，副处理为 B，分 B_1、B_2、B_3、B_4 4 个水平，裂区设计，重复 3 次($r=3$)，副区计产面积33 m^2，其田间排列和产量(kg/33 m^2)见下表，试做方差分析。

重　复　Ⅰ

A_1		A_3		A_2	
B_2	B_1	B_3	B_2	B_4	B_3
37	29	15	31	13	13
B_3	B_4	B_4	B_1	B_1	B_2
18	17	16	30	28	31

重　复　Ⅱ

A_3		A_2		A_1	
B_1	B_3	B_4	B_3	B_2	B_3
27	14	12	13	32	14
B_4	B_2	B_2	B_1	B_4	B_1
15	28	28	29	16	28

重　复　Ⅲ

A_1		A_3		A_2	
B_4	B_3	B_2	B_4	B_1	B_2
15	17	31	13	25	29
B_2	B_1	B_1	B_3	B_3	B_4
31	32	26	11	10	12

5－9　比较三种猪饲料 A_1、A_2 和 A_3 对猪催肥的效果，测得每头猪增加的重量(Y，kg)和初始重量(X，kg)，试测验这三种肥料对猪的催肥效果有无显著差异。

饲料	观察值(X_{ij}，Y_{ij})								
A_1	X_{1j}	15	13	11	12	12	16	14	17
	Y_{1j}	85	83	65	76	80	91	84	90
A_2	X_{2j}	17	16	18	18	21	22	19	18
	Y_{2j}	97	90	100	95	103	106	99	94
A_3	X_{3j}	22	24	20	23	25	27	30	32
	Y_{3j}	89	91	83	95	100	102	105	110

第6章

相关与回归分析

在科学研究和生产实践中，经常需要对两类变量之间的关系进行分析。例如作物产量和种植密度、害虫的发生量和气象因子、动物的体重和生长天数等，这些变量之间的关系分析即相关和回归分析。相关和回归分析是生物学研究中最为常用的统计分析方法之一。

相关分析是计算反映各个变量之间相关密切程度和性质的统计数。回归关系一般用反映因变量和自变量之间数量关系的回归方程表示，求解方法通常采用最小二乘法。

回归分析依自变量个数的多少分为一元回归和多元回归；依因变量和自变量之间的关系类型分为线性回归和非线性回归。另外，对于社会属性类数据，可以通过 Logistic 回归模型拟合线性模型。

6.1 线性相关分析

Pearson 积差相关系数衡量了两个定量变量之间的线性相关程度。它是说明有直线关系的两变量间相关关系密切程度和相关方向的统计指标，通常用 r 表示，取值范围为 $[-1,1]$，绝对值越接近 1，则线性相关关系越密切。

Spearman 等级相关系数则衡量分级定序变量之间的相关程度。

Kendall's Tau 相关系数也是一种非参数的等级相关度量。如欲考察几位老师对多篇作文的评分标准是否一致（又称评分者信度），就应该使用肯德尔系数。

对于定序数据而言，Spearman 系数与 Pearson 系数是等价的；如果一个变量为定量数据，一个变量为定序数据，应计算 Spearman 系数或将定量数据变为定序数据后使用 Pearson 系数。

肯德尔系数一个重要优点在于便于解释，如果肯达尔系数等于 1/3，意味着：一致情况的出现频率是不一致的两倍。

6.1.1 cor()函数

在 R 语言中，cor()函数用于计算两个变量或数据框中列之间的相关系数。其一般的使用格式为：

```
cor(x, y = NULL, use = "everything",
       method = c("pearson", "kendall", "spearman"))
```

x:一个向量或矩阵,如果是矩阵,则计算列之间的相关性。

y:一个向量,当 x 是向量时,y 也是向量,cor()将计算 x 和 y 之间的相关性。如果 x 是矩阵,则 y 可以省略。

use:一个表示如何处理缺失值的字符串。可能的值包括"all.obs"(所有观测值都必须完整),"complete.obs"(只使用完整的观测值对),"pairwise.complete.obs"(成对地使用完整的观测值),"everything"(默认,使用所有可能的观测值)。

method:字符串,指定计算相关系数的方法。"pearson"是皮尔逊相关系数,"kendall"是肯德尔 τ 相关系数,"spearman"是斯皮尔曼 ρ 相关系数。

其返回值根据输入不同而变化。若输入为两个向量,返回一个单一的相关系数值,范围在−1 到 1 之间,反映两向量线性相关程度,其中 1 表示完全正相关,−1 表示完全负相关,0 表示没有线性关系。当输入是矩阵或数据框时,返回一个相关系数矩阵,是方阵,行数和列数与输入的列数相同,对角元素为 1,非对角元素表示对应列间的相关系数。同时,可通过 method 参数指定计算方法(如"pearson"、"kendall"、"spearman"),但不改变返回值基本形式。

cov()函数用于计算两个变量或数据框中列之间的协方差。协方差是衡量两个变量共同变化的程度的统计量。如果两个变量一起增加或减少,它们的协方差是正的;如果一个增加而另一个减少,协方差是负的。cov()函数的语法结构与 cor()函数一致,在此不赘述。

例 6-1　许多害虫的发生都和气象条件有一定的关系。山东临沂地区测定 1964—1973 年(共 10 年)间 7 月下旬的温雨系数(雨量 mm/平均温度℃)和大豆第二代造桥虫发生量(每百株大豆上的虫数)的关系如表 6-1,试进行相关分析。(莫惠栋著《农业试验统计(第二版)》313 页例 10.1)

表 6-1　温雨系数和造桥虫虫口密度

温雨系数(X)	虫口密度(Y)	温雨系数(X)	虫口密度(Y)
1.58	180	2.41	175
9.98	28	11.01	40
9.42	25	1.85	160
1.25	117	6.04	120
0.30	165	5.92	80

代码 6-1　使用 cov()和 cor()函数计算相关系数

```
> example6_1 <- read.csv("E:/RinBio/Chapt6/Example6_1.csv", header = TRUE, sep = ",")
> cov_res <- cov(example6_1)
```

```
> print(cov_res)
             x          y
x   16.31763   - 230.2444
y   - 230.24444   3837.5556
> cor_res <- cor(example6_1)
> print(cor_res)
             x           y
x  1.0000000   - 0.9200964
y  - 0.9200964   1.0000000
```

本例中，首先使用 read.csv()函数从“E:/RinBio/Chapt6/Example6_1.csv”文件中读取数据；运用 cov()函数计算 example6_1 数据的协方差矩阵，结果存于 cov_res，并通过 print()函数输出，以此了解变量间的协变情况；之后，使用 cor()函数计算 example6_1 数据的相关系数矩阵，结果存储在 cor_res 中，并同样通过 print()函数输出，进而可分析温雨系数和造桥虫发生量之间的线性相关程度。

根据结果可知：x 的方差为 16.31763，y 的方差为 3837.5556，x 与 y 的协方差为 −230.24444。协方差为负值意味着当 x 高于其均值时，y 往往低于其均值，反之亦然。利用 cor()函数计算了相关系数，此处是利用默认值，计算的是 Pearson 相关系数，结果表明虫口密度与温雨系数相关系数为−0.9201。

6.1.2 cor.test()函数

cor()函数仅仅是计算了相关系数的数值，没有对求得的相关系数进行显著性测验。cor.test()函数不仅可以实现相关系数的计算，并且可以对其进行显著性测验。其一般格式为：

```
cor.test(x, y,
         alternative = c("two.sided", "less", "greater"),
         method = c("pearson", "kendall", "spearman"),
         exact = NULL, conf.level = 0.95, continuity = FALSE, ...)
```

x：必需的参数，表示第一个变量，可以是数值向量、矩阵的列或数据框的列。

y：必需的参数(除非 x 是矩阵且 use = "complete.obs")，表示第二个变量，类型与 x 相同。如果 x 是矩阵，则 y 可以省略，此时函数会计算 x 中各列之间的相关性。

alternative：指定备择假设的类型，可以是"two.sided"(双侧检验，默认)、"less"(单侧检验，备择假设为相关系数小于 0)或"greater"(单侧检验，备择假设为相关系数大于 0)。

method：指定计算相关系数的方法，可以是"pearson"(皮尔逊相关系数，默认，适用于连续变量且数据呈正态分布的情况)、"kendall"(肯德尔秩相关系数，适用于等级数据)或"spearman"(斯皮尔曼秩相关系数，适用于不满足正态分布假设的连续数据或等级数据)。

exact：对于小样本数据，可以指定是否进行精确计算。对于"pearson"方法，如果样本量小于或等于 50 且没有缺失值，可以设置为 TRUE 以进行精确计算。对于"kendall"和"spearman"方法，此参数的行为可能依赖于具体的实现。

conf.level：设置置信区间的置信水平，默认值为 0.95，即 95%置信水平。

continuity：此参数通常与肯德尔和斯皮尔曼秩相关系数相关，用于控制连续性校正的应用。当设置为 TRUE 时，会应用某种形式的连续性校正来调整 p 值。

...：其他参数，可能用于传递给特定的方法或控制函数的其他行为。

cor.test()函数返回值是一个列表，包含多项关键信息。Statistic 为检验统计量的值；p.value 用于判断相关性是否显著，低于设定阈值则显著；parameter 是检验的自由度；estimate 给出相关系数估计值，体现变量间线性相关程度与方向；null.value 是原假设下的相关系数，默认是 0；alternative 指明备择假设类型，有双侧、左侧或右侧检验；还可能有 conf.int，即相关系数的置信区间，用于估计总体相关系数范围。

例 6-2　对例 6-1 数据求取相关系数并进行显著性测验。

代码 6-2　使用 cor.test()函数计算例 6-1 的相关系数并进行显著性测验

```
> example6_2 <- read.csv("E:/RinBio/Chapt6/Example6_1.csv", header = TRUE, sep = ",")
> cor_res <- cor.test(example6_2$x,example6_2$y)
> print(cor_res)

    Pearson's product-moment correlation

data:  example6_2$x and example6_2$y
t = -6.6441, df = 8, p-value = 0.0001618
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.9812614 -0.6904724
sample estimates:
       cor
-0.9200964
```

本例中，首先使用 read.csv()函数从指定路径“E:/RinBio/Chapt6/Example6_1.csv”读取数据文件，读取的数据被存储在数据框 example6_2 中；调用 cor.test()函数对 example6_2 数据框里的 x 变量和 y 变量进行相关性检验，检验结果保存在 cor_res 里；最后，使用 print()函数输出 cor_res，即输出相关性检验的结果，其中包含相关系数、值等信息，以此来判断 x 和 y 之间的相关性及其显著性。

根据计算结果，x 和 y 变量的相关系数与 cor()计算结果一致，显著性测验的概率值小于 0.01，表明虫口密度与温雨系数呈极显著负相关，即虫口密度随着温雨系数的增大而减小。

6.1.3　psych 包的 corr.test()函数

cor.test()只能用于计算两个变量的相关性检验，如果对多个变量或者矩阵数据的话，则可以使用 psych 包的 corr.test()函数。其一般格式为：

```
corr.test(x, y = NULL, use = "pairwise",method ="pearson",adjust ="holm",
    alpha =.05,ci = TRUE,minlength = 5,normal = TRUE)
```

x:必需参数,一个矩阵或数据框架,包含了要进行相关性分析的变量。

y:可选参数,另一个矩阵或数据框架,与 x 具有相同的行数。如果 y 为 NULL(默认),则函数将计算 x 中各列之间的相关性。如果提供了 y,则函数将计算 x 和 y 之间变量的相关性。

use:指定如何处理缺失值。可选值有"pairwise"(默认值,成对删除缺失值)和"complete"(仅使用完整案例)。

method:指定计算相关系数的方法。可选值有"pearson"(默认,皮尔逊相关系数)、"spearman"(斯皮尔曼秩相关系数)和"kendall"(肯德尔秩相关系数)。

adjust:指定多重比较校正的方法。可选值有多种,如"holm"(Holm 校正,默认值)、"bonferroni"、"hochberg"等。如果不进行校正,可以设置为"none"。

alpha:显著性水平,用于判断相关性的显著性。默认值为 0.05。

ci:是否计算置信区间。默认值为 TRUE。

minlength:置信区间计算中的最小缩写长度。默认值为 5。

normal:指定是否假设数据服从正态分布。默认值为 TRUE。如果设置为 FALSE,则对于非正态分布数据可能会得到更准确的 p 值,但计算速度会变慢。

corr.test()函数返回值是一个包含多种信息的列表。其中,r 为相关系数矩阵,呈现各变量两两之间的相关程度;p 是对应的 p 值矩阵,用于判断各变量间相关性是否显著;n 是样本量矩阵,显示每个相关系数计算所基于的样本数量;se 是标准误矩阵,反映相关系数估计的误差情况;z 是经过费舍尔 Z 转换后的矩阵;adjust 说明了对 p 值进行多重比较调整时所采用的方法,这些信息共同为变量间相关性的分析提供了全面的依据。

例 6-3 玉米籽粒淀粉 RVA 黏性指标是籽粒品质性状的重要指标,测定了 30 个玉米品种的 RVA 黏性指标,包括峰值黏度(PV)、谷值黏度(TV)、崩解值(BD)、终值黏度(FV)、回复值(SB)、糊化时间(PT)和糊化温度(PTP)。试进行相关分析。

表 6-2 30 个玉米品种的 RVA 指标

品种编号	PV	TV	BD	FV	SB	PT	PTP
1	1 147	919	228	2 834	1 687	5.80	73.85
2	1 172	1 019	153	2 850	1 678	6.00	74.65
3	1 302	938	364	2 948	1 646	5.13	74.60
4	1 285	936	349	2 929	1 644	5.07	74.00
5	1 156	713	443	2 025	869	4.87	72.25
6	1 156	713	443	2 025	869	4.87	72.25
7	913	724	189	2 258	1 345	5.20	73.15
8	892	702	190	2 171	1 279	5.27	73.90

续　表

品种编号	PV	TV	BD	FV	SB	PT	PTP
9	1 101	980	121	1 824	723	6.20	73.90
10	1 187	1 058	129	2 023	836	6.00	73.05
11	892	719	173	2 231	1 339	5.47	75.55
12	906	716	190	2 343	1 437	5.33	74.65
13	1 353	973	380	2 797	1 444	5.13	72.70
14	1 360	994	366	2 727	1 367	5.13	72.30
15	1 075	897	178	1 972	897	5.53	73.10
16	1 069	881	188	1 992	923	5.47	73.05
17	1 918	1 383	535	1 773	−145	4.93	70.75
18	1 865	1 370	495	1 750	−115	5.00	71.50
19	1 936	1 244	692	3 396	1 460	5.27	73.10
20	1 843	1 169	674	3 288	1 445	5.07	73.00
21	1 286	1 103	183	2 566	1 280	6.07	73.90
22	1 347	1 181	166	2 587	1 240	6.00	74.80
23	1 882	1 303	579	4 100	2 218	5.53	73.10
24	1 932	1 321	611	4 032	2 100	5.47	72.45
25	1 139	919	220	2 832	1 693	5.60	73.85
26	1 162	921	241	2 798	1 636	5.73	73.95
27	586	564	22	813	227	5.40	69.85
28	756	711	45	985	229	5.00	69.95
29	2 337	1 442	895	3 830	1 493	5.13	70.70
30	2 149	1 371	778	3 881	1 732	5.07	70.80

代码 6－3　使用 psych 包中的 corr.test()函数对例 6－3 中的多个相关系数进行计算

```
> example6_3 <- read.csv("E:/RinBio/Chapt6/Example6_3.csv", header = TRUE)
> library(psych)
> cor_res <- corr.test(example6_3, adjust = "none")
> print(cor_res)
Call:corr.test(x = example6_3, adjust = "none")
Correlation matrix
```

```
PV    TV    BD    FV    SB    PT    PTP
PV    1.00  0.93  0.92  0.71  0.23  - 0.24 - 0.29
TV    0.93  1.00  0.73  0.62  0.16  0.03 - 0.20
BD    0.92  0.73  1.00  0.70  0.27  - 0.50 - 0.35
FV    0.71  0.62  0.70  1.00  0.85  0.030.21
SB    0.23  0.16  0.27  0.85  1.00  0.220.51
PT    - 0.24 0.03  - 0.50 0.03  0.22  1.000.53
PTP   - 0.29 - 0.20 - 0.35 0.21  0.51  0.531.00
Sample Size
[1] 30
Probability values (Entries above the diagonal are adjusted for multiple tests.)
      PV    TV    BD    FV    SB    PT    PTP
PV    0.00  0.00  0.00  0.00  0.22  0.19  0.12
TV    0.00  0.00  0.00  0.00  0.41  0.89  0.28
BD    0.00  0.00  0.00  0.00  0.14  0.00  0.06
FV    0.00  0.00  0.00  0.00  0.00  0.89  0.26
SB    0.22  0.41  0.14  0.00  0.00  0.24  0.00
PT    0.19  0.89  0.00  0.89  0.24  0.00  0.00
PTP   0.12  0.28  0.06  0.26  0.00  0.00  0.00

 To see confidence intervals of the correlations, print with the short = FALSE option
```

本例中，首先使用 read.csv()函数从“E:/RinBio/Chapt6/Example6_3.csv”文件读取数据，读取的数据存储在数据框 example6_3 中；使用 library(psych)加载 psych 包，该包提供了用于进行相关性分析的函数；然后，调用 corr.test()函数对 example6_3 数据框中的各变量进行相关性检验，设置 adjust = "none"表示不进行多重比较调整，将检验结果存储在 cor_res 中；最后，使用 print()函数输出 cor_res，即输出相关性分析的结果，其中包含各变量间的相关系数、p 值等信息，从而可了解各 RVA 黏性指标之间的相关性。

数据结果包括了三部分，第一部分为相关系数矩阵(Correlation Matrix)，第二部分为样本容量(Sample Size)，第三部分是对相关系数进行显著性测验的结果(Probability Values)。结果表明 PV、TV、BD、FV 两两间均存在显著相关，SB 与 FV、PTP 之间存在显著相关，PT 与 BD、PTP 之间存在极显著相关。

6.1.4 corrplot 包

corrplot 是一个绘制相关矩阵和置信区间的包，它也包含了一些矩阵排序的算法。比较常用的函数包括 corrplot()函数和 corrplot.mixed()函数。corrplot()函数可以将相关系数矩阵以多种图形形式展示出来，如圆形、方形、椭圆等，还能通过颜色、阴影等方式来表示相关系数的大小和正负，帮助用户快速了解变量间的相关关系，发现数据中的模式和趋势，其一般格式为：

```
corrplot(corr, method = "circle", type = "full", order = "original",
         add = FALSE, col = NULL, bg = "white", title = "",
         is.corr = TRUE, diag = TRUE,...)
```

corr:这是必须提供的参数,为一个相关系数矩阵,通常是由 cor()函数计算得到的矩阵。

method:用于指定绘制相关系数矩阵的方法,可选值有"circle"(圆形)、"square"(方形)、"ellipse"(椭圆)、"number"(数字)、"shade"(阴影)、"color"(颜色块)和"pie"(饼图)等,默认是"circle"。

type:控制相关系数矩阵显示的类型,取值为"full"(显示完整矩阵)、"lower"(只显示下三角部分)、"upper"(只显示上三角部分),默认是"full"。

order:设置矩阵中变量的排序方式,可选值有"original"(按原始顺序)、"AOE"(按特征值的绝对值排序)、"FPC"(按第一主成分排序)、"hclust"(按层次聚类结果排序)等,默认是"original"。

col:指定用于表示相关系数的颜色向量。如果为 NULL,则会根据 method 参数的不同使用默认的颜色方案。

add:逻辑值,若为 TRUE,则在现有图形上添加相关系数图,默认为 FALSE。

bg:指定图形的背景颜色,默认为"white"。

title:为图形添加标题,默认为空字符串。

is.corr:逻辑值,指示输入的 corr 矩阵是否为相关系数矩阵。如果为 FALSE,则函数会假设 corr 矩阵中的值是协方差。默认为 TRUE。

diag:逻辑值,用于决定是否显示矩阵的对角线元素。默认值为 TRUE。

corrplot()函数还有许多其他可选参数,如 addrect、tl.pos 等,可以进一步对图形的细节进行调整和美化。

若绘制图形和数值混合矩阵,则使用 corrplot.mixed()函数,该函数可以对上三角和下三角部分采用不同的绘图方法来展示相关系数矩阵,为用户提供更灵活和个性化的相关系数矩阵可视化方案,有助于更细致地分析变量之间的相关性,其一般格式为:

```
corrplot.mixed(corr, lower = c("circle", "square", "ellipse", "number", "shade", "color",
"pie"),
               upper = c("circle", "square", "ellipse", "number", "shade", "color", "pie"),
               diag = c("n", "l", "u"), order = "original", hclust.method = "complete",
               reorder = TRUE, type = "full", tl.pos = NULL, tl.cex = 1, tl.col = "black",
               number.cex = 0.7, ...)
```

corr:这是一个必需的参数,代表相关系数矩阵,通常通过 cor()函数计算得出。该矩阵包含了各个变量之间的相关系数,是绘制相关系数图的基础数据。

lower:指定相关系数矩阵下三角部分的绘图方法。可选值有"circle"(圆形)、"square"(方形)、"ellipse"(椭圆)、"number"(直接显示数字)、"shade"(阴影)、"color"(颜色块)和"pie"(饼图)等。默认值为"circle"。

upper:指定相关系数矩阵上三角部分的绘图方法,可选值与 lower 相同。默认值为"number"。

diag:控制对角线上元素的显示方式。取值为"n"时,不显示对角线元素;取值为"l"时,对角线元素使用下三角的绘图方法;取值为"u"时,对角线元素使用上三角的绘图方法。

order:设置矩阵中变量的排序方式。可选值有"original"(按原始顺序)、"AOE"(按特征值的绝对值排序)、"FPC"(按第一主成分排序)、"hclust"(按层次聚类结果排序)等。默认值为"original"。

hclust.method:当 order = "hclust"时,该参数指定层次聚类所使用的方法。默认值为"complete"。

reorder:逻辑值,决定是否对矩阵进行重新排序。默认值为 TRUE。

type:控制相关系数矩阵显示的类型,可选值为"full"(显示完整矩阵)、"lower"(只显示下三角部分)、"upper"(只显示上三角部分)。默认值为"full"。

tl.pos:指定变量标签的位置。可以取值为"n"(不显示标签)、"l"(仅显示下三角标签)、"u"(仅显示上三角标签)、"d"(仅显示对角线标签)、"lt"(显示左和上标签)、"ld"(显示左和对角线标签)、"ud"(显示上和对角线标签)、"ldu"(显示左、上和对角线标签)等。如果为 NULL,则根据 type 参数自动选择合适的位置。

tl.cex:指定变量标签的字体大小,默认值为 1。

tl.col:指定变量标签的颜色,默认值为"black"。

number.cex:当绘图方法为"number"时,该参数指定显示数字的字体大小,默认值为 0.7。

...:可以传递给 corrplot()函数的其他参数。

例 6-4 利用 corrplot 包实现对例 6-3 数据相关分析结果的可视化。

代码 6-4 利用 corrplot 包实现对例 6-3 数据相关分析结果的可视化

```
> example6_4 <- read.csv("E:/RinBio/Chapt6/Example6_3.csv", header = TRUE, sep = ",")
> corr_res <- cor(example6_4)
> print(corr_res)
             PV         TV         BD         FV         SB          PT          PTP
PV           1.0000000  0.93457707 0.9228918  0.70892518 0.2284310  - 0.24419516 - 0.2930846
TV           0.9345771  1.00000000 0.7255245  0.62069314 0.1557145  0.02743926   - 0.2035564
BD           0.9228918  0.72552449 1.0000000  0.69956004 0.2733467  - 0.50207892 - 0.3466331
FV           0.7089252  0.62069314 0.6995600  1.00000000 0.8485765  0.02661835   0.2132187
SB           0.2284310  0.15571453 0.2733467  0.84857653 1.0000000  0.21992788   0.5141821
PT           - 0.2441952 0.02743926 - 0.5020789 0.02661835 0.2199279 1.00000000   0.5278595
PTP          - 0.2930846 - 0.20355641- 0.3466331 0.21321871 0.5141821 0.52785949   1.0000000
> library(corrplot)
```

```
> corrplot(corr_res, method ="color", addCoef.col = 'grey')
> corrplot(corr_res, method ="circle", type = "upper",addCoef.col = 'grey')
> corrplot.mixed(corr_res,lower ="number", upper ="ellipse")
> corrplot.mixed(corr_res,lower ="circle", upper ="number")
```

本例中，首先运用 read.csv()函数从“E:/RinBio/Chapt6/Example6_3.csv”文件读取数据，header＝TRUE 表明首行为列名；使用 cor()函数计算 example6_4 数据框中各变量间的相关系数矩阵，结果存于 corr_res 并通过 print()函数输出。之后，使用 library(corrplot)加载 corrplot 包，利用其功能进行相关矩阵的可视化，先以颜色块方式展示相关矩阵，相关系数颜色设为灰色；再用圆形展示上三角相关矩阵，同时标注灰色的相关系数；通过 corrplot.mixed()函数分别以对角线下为数值、对角线上为椭圆，以及对角线下为圆形、对角线上为数值这两种混合方式展示相关矩阵，以多维度直观呈现各 RVA 黏性指标间的相关性。

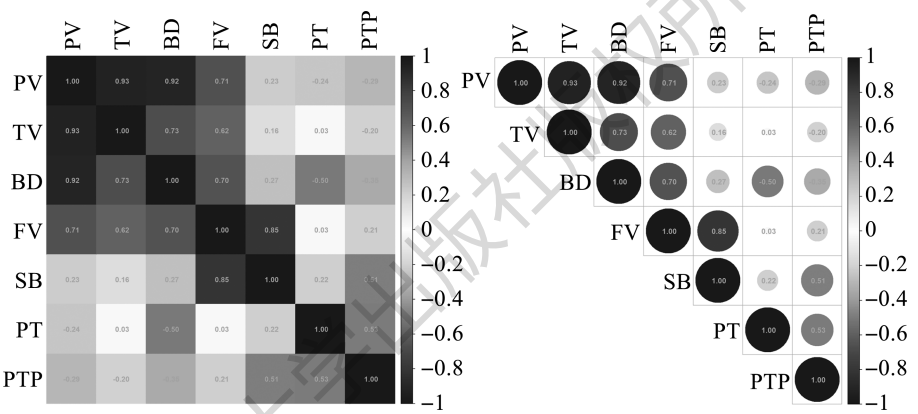

图 6－1　corrplot()函数绘制的相关系数可视化矩阵

图 6－1 显示的是 corrplot()函数绘制的相关系数可视化矩阵，左侧的图形是利用 method ="color"指定使用颜色深浅表示相关系数的大小，用 addCoef.col = 'grey'指定在图形上添加相关系数数值，数值颜色为灰色。右侧的图形是利用 method ="circle"指定使用圆形表示相关系数的大小，用 type = "upper"指定仅绘制上三角部分。

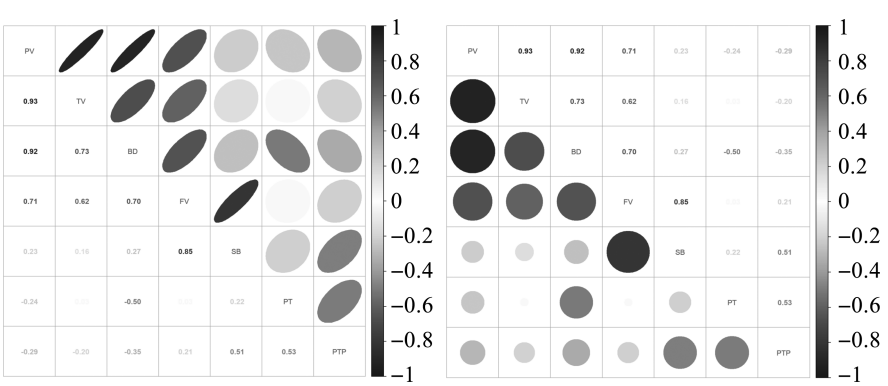

图 6－2　corrplot.mixed()函数绘制的相关系数可视化矩阵

图 6－2 显示的是 corrplot.mixed()函数绘制的相关系数可视化矩阵。左侧的图形用 lower ="number"指定下三角部分使用数字表示相关系数，用 upper ="ellipse"指定上三角部分使用椭圆表示相关系数的大小和方向。右侧的图形用 lower ="circle"指定下三角部分使用圆形面积表示相关系数的大小，用 upper ="number"指定上三角部分使用数字表示相关系数。

6.1.5 PerformanceAnalytics 包的 chart.Correlation()函数

R 语言的 PerformanceAnalytics 包是用于业绩和风险分析的计量经济学工具，该包中的 chart.Correlation()函数可以实现相关系数的计算、显著性测验和可视化。

```
chart.Correlation(R, histogram = TRUE, pch = 16,
                  method = c("pearson", "kendall", "spearman"), ...)
```

R：这是一个必需的参数，它应该是一个包含数值数据的矩阵或数据框。矩阵或数据框的每一列代表一个变量，函数会计算这些变量之间的相关性并进行可视化展示。

histogram：逻辑值，默认为 TRUE。当设置为 TRUE 时，在散点图矩阵的对角线上绘制每个变量的直方图，用于展示变量的分布情况；当设置为 FALSE 时，对角线上不绘制直方图。

pch：指定散点图中数据点的符号样式。它可以是一个整数或字符，不同的值对应不同的符号形状，例如 16 表示实心圆，默认值为 16。

method：指定计算相关系数所使用的方法，可选值为"pearson"（皮尔逊相关系数，用于衡量线性相关性）、"kendall"（肯德尔秩相关系数，适用于非正态分布数据）和"spearman"（斯皮尔曼秩相关系数，对数据分布没有严格要求），默认使用"pearson"方法。

...：其他可选参数，这些参数允许用户自定义热力图的外观和行为。例如，可以通过设置 col 参数来指定颜色方案，通过 main 参数来添加标题等。

例 6－5 利用 PerformanceAnalytics 包的 chart.Correlation()函数实现对例 6－3 数据相关分析结果的可视化。

代码 6－5 利用 PerformanceAnalytics 包的 chart.Correlation()函数实现对例 6－3 数据相关分析结果的可视化

```
> example6_5 <- read.csv("E:/RinBio/Chapt6/Example6_3.csv", header = TRUE, sep = ",")
> library(PerformanceAnalytics)
> chart.Correlation(example6_5)
```

本例中，首先使用 read.csv 函数从“E:/RinBio/Chapt6/Example6_3.csv”文件中读取数据，读取后的数据存储在数据框 example6_5 中；使用 library(PerformanceAnalytics)加载 PerformanceAnalytics 包，该包提供了用于金融和经济数据可视化与分析的工具；最后，调用 chart.Correlation 函数对 example6_5 数据框中的各变量进行相关性分析，并将分析结果以直观的图形方式展示出来，方便用户观察各变量之间的相关性。

执行上述操作后，将会返回如图 6－3 所示的计算结果。该图的上三角为数值化表示

的相关系数的数值，* 表示在 0.05 的水平上显著，** 表示在 0.01 的水平上显著，*** 表示在 0.001 的水平上显著。主对角线显示的变量名称及各变量的直方图。下三角表示两个变量的散点图。

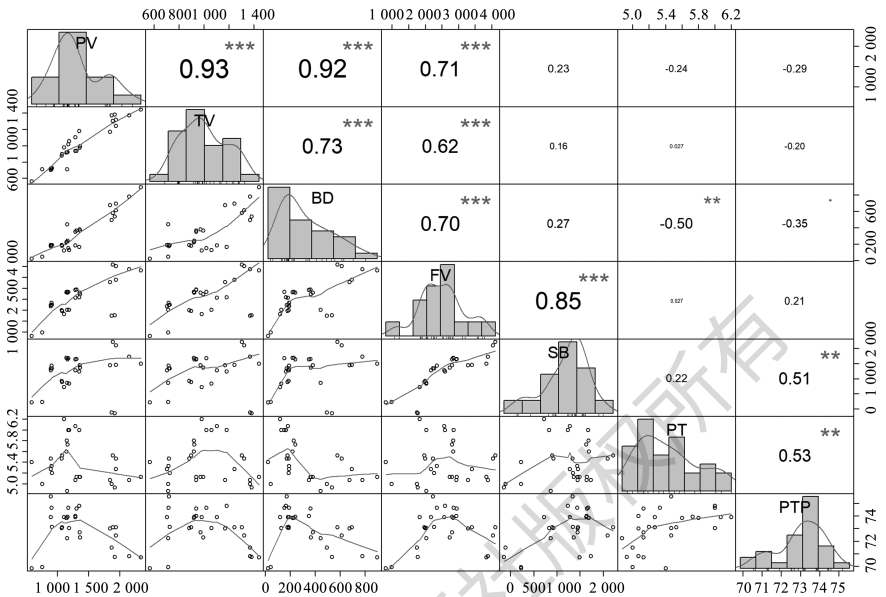

图 6－3　chart.Correlation()函数绘制的相关系数可视化矩阵

6.2　一元线性回归分析

如果两个变量在散点图上呈线性，我们就可设想其数量关系是可能用一个线性方程式来表示的。根据解析几何学原理，当 $\hat{Y}=f(X)$ 为线性方程时，其一般形式为：$\hat{Y}=a+bX$。此式读作：Y 依 X 的线性回归方程。其中，a 是 $X=0$ 时的 $\hat{Y}$ 值，即直线在 Y 轴上的截距，叫作回归截距；b 是 X 每增加 1 个单位，Y 平均地将要增加（$b>0$）或减小（$b<0$）的单位数，叫作回归系数或斜率。

利用最小二乘法，可以得到 a 和 b 的估值：

$$b=\frac{\sum(X-\bar{x})(Y-\bar{y})}{\sum(X-\bar{x})^2},a=\bar{y}-b\bar{x}$$

一般来说，对于任意一组观察值（X_i，Y_i），（$i=1,2,\cdots,N$），当 $\Sigma(X-\bar{x})\neq 0$ 时，由其计算公式总可以求出回归直线。但是这样建立的线性回归方程是否有意义，即 X 对 Y 是否有影响，且两者是否为线性关系，则必须进行假设测验。

lm()函数只是 R 中众多回归的函数之一。lm()函数是用于一般目的回归分析的函数，而其他函数则具有各自特殊的用途。lm 函数的基本格式为：

```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

formula:这是一个必需的参数,用于指定线性模型的形式。它使用公式表示法,例如 y～x1＋x2,其中 y 是因变量,x1 和 x2 是自变量,～表示“由…… 决定”或“依赖于”的关系。可以使用更复杂的公式,如包含交互项(y～x1 * x2)、多项式项(y～poly(x1, 2))等。

data:同样是必需的参数,通常是一个数据框(data.frame)。数据框中应包含公式中所涉及的所有变量,lm() 函数会从这个数据框中提取相应的数据来拟合模型。

subset:可选参数,用于指定一个向量或表达式,以从数据中选择特定的观测值来拟合模型。例如,可以使用逻辑向量 subset = data$x1 > 10 来选择 x1 大于 10 的观测值。

weights:可选参数,是一个与数据集中观测值数量相同的向量,用于为每个观测值指定权重。在加权最小二乘法中,权重用于调整每个观测值对回归系数估计的影响程度。

na.action:指定处理数据中缺失值(NA)的方法。默认值为 getOption("na.action"),常见的取值有 na.omit(删除包含缺失值的观测值)、na.exclude(在计算中排除缺失值,但保留观测值的索引)等。

method:指定拟合模型所使用的方法,默认值为"qr",表示使用 QR 分解方法来求解线性回归问题。一般情况下,使用默认方法即可。

model:逻辑值,默认为 TRUE。如果为 TRUE,则返回的模型对象中会包含模型矩阵和响应变量;如果为 FALSE,则不包含这些信息,可节省内存。

x:逻辑值,默认为 FALSE。如果为 TRUE,则返回的模型对象中会包含模型矩阵;如果为 FALSE,则不包含。

y:逻辑值,默认为 FALSE。如果为 TRUE,则返回的模型对象中会包含响应变量;如果为 FALSE,则不包含。

qr:逻辑值,默认为 TRUE。如果为 TRUE,则返回的模型对象中会包含 QR 分解的结果;如果为 FALSE,则不包含。

singular.ok:逻辑值,默认为 TRUE。当设计矩阵存在多重共线性(即矩阵不满秩)时,如果 singular.ok 为 TRUE,函数会继续执行并给出警告;如果为 FALSE,则会报错。

contrasts:用于指定分类变量的对比矩阵,是一个列表,列表中的每个元素对应一个分类变量的对比方式。一般情况下,使用默认设置即可。

offset:一个与响应变量长度相同的向量,用于指定模型中的偏移项。偏移项是一个已知的线性组合,会直接加到线性预测器中。

...:可以传递给其他底层函数的额外参数。

lm()函数返回一个 lm 类对象。该对象包含诸多重要信息,coefficients 给出模型中自变量和截距的估计系数,展示各变量对响应变量的影响;residuals 是观测值与预测值的差值,反映模型拟合偏差;fitted.values 是模型对响应变量的预测值;model 存储了拟合所用的数据;还包含如 df.residual(残差自由度)、sigma(残差标准差)等统计量,可辅助评估

模型质量。可通过 summary()等函数进一步剖析该对象,以全面了解模型表现。为了提取更多的信息,可以使用对 lm()类对象有特殊操作的通用函数,如表 6－3 所示。

表 6－3　R 中对拟合线性模型相关的其他函数

函数	功能
coef()	提取系数向量的估计值
resid()/residuals()	提取残差向量
fitted()/predict()	提取拟合值向量
vcov()	提取 β 的普通最小二乘估计量条件方差阵的估计值
deviance()	提取残差平方和
formula()	提取模型公式
df.residual()	提取残差的自由度
nobs()	提取模型中样本容量 n
AIC()	提取模型中 AIC 信息准则
BIC()	提取模型中 BIC 信息准则
logLik()	提取模型的对数似然函数值

例 6－6　lm()函数对例 6－1 数据的一元线性回归分析。

代码 6－6　利用 lm()函数对例 6－1 数据的一元线性回归分析

```
> example6_6 <- read.csv("E:/RinBio/Chapt6/Example6_1.csv", header = TRUE, sep = ",")
> plot(example6_6$x,example6_6$y)
> fit <- lm(example6_6$y~example6_6$x)
> anova(fit)
Analysis of Variance Table

Response: example6_6$y
              Df  Sum Sq Mean Sq F value    Pr(>F)
example6_6$x   1  29239.1  29239.1  44.144  0.0001618  ***
Residuals      8   5298.9    662.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(fit)

Call:
lm(formula = example6_6$y~example6_6$x)

Residuals:
```

```
    Min       1Q  Median       3Q      Max
- 44.574  - 14.358  - 1.544  21.347  29.793
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    179.212     13.338    13.436 9.02e - 07 ***
example6_6$x    - 14.110      2.124     - 6.644     0.000162 ***
- - -
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1

Residual standard error: 25.74 on 8 degrees of freedom
Multiple R - squared:  0.8466,Adjusted R - squared:  0.8274
F - statistic: 44.14 on 1 and 8 DF,  p - value: 0.0001618

> confint(fit)
                 2.5 %      97.5 %
(Intercept)    148.45388     209.970509
example6_6$x - 19.00749   - 9.212846
> abline(fit)
```

本例中，首先使用 read.csv()函数从“E:/RinBio/Chapt6/Example6_1.csv”文件读取数据，读取的数据存于 example6_6 数据框；通过 plot()函数绘制 example6_6 数据框中 x 变量和 y 变量的散点图(该函数的具体使用方法可参考本书第十章的内容)，以直观呈现二者关系；然后，使用 lm()函数构建线性回归模型，将 y 设为因变量，x 设为自变量，模型拟合结果存储在 fit 中；使用 anova()函数对拟合的线性回归模型进行方差分析，以评估模型的显著性和各因素对响应变量的影响程度，之后使用 summary()函数输出回归模型的详细统计信息，如系数估计值、标准误差、t 值、p 值等；使用 confint()函数计算回归模型系数的置信区间；最后使用 abline()函数在之前绘制的散点图上添加回归直线，从而完成对例 6-1 数据的一元线性回归分析及结果展示。

如图 6-4 所示，两个变量间表现出明显的线性关系，说明温雨系数和大豆第二代造桥虫发生量存在一定的线性相关。通过 lm()函数拟合回归方程，由 anova()函数得到方差分析表，结果显示模型拟合在 0.001 的水平上达到了显著($F=44.144$，$p=0.00016$)。由 summary()函数获取回归方程的相关信息，Coefficients 给出了截距项和 x 的估计值(Estimate)，标准误(Std Error)，t 值(t value)以及 p 值(Pr(v>|t|))，本例中截距项的估计值为 179.212，回归系数的估计值为 −14.110，因此线性回归方程为：$\hat{y}=179.212-14.110x$，并且截距项和回归系数的显著性测验在 0.001 的水平上均达到了显著。Multiple R-squared 给出了回归模型的拟合度，称为决定系数，其值取值范围为 0 到 1，越大表示两变量之间密切程度越高。Adjusted R-squared 表示校正后的决定系数。F-statistic对整个模型进行检验，此处 p 值为 0.000 161 8，说明回归模型具有极显著的统计意义。confint()函数给出了自变量 x 的 95% 的置信区间，可知 x 的置信区间为

[－19.007，9.212 8]。最后，可以调用 abline()函数在散点图中添加回归直线。

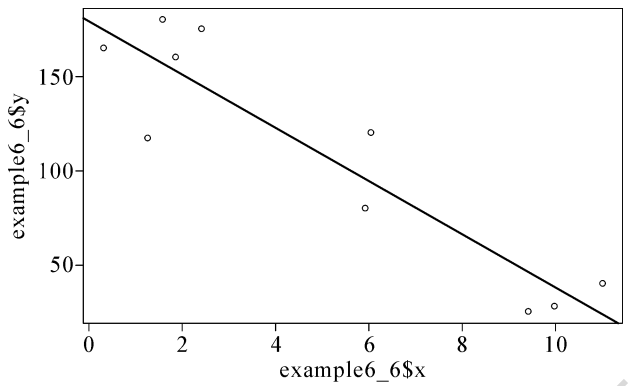

图 6－4　温雨系数和大豆第二代造桥虫发生量的关系

6.3　多元线性回归分析

对比一元线性回归，多元线性回归是用来确定 2 个或 2 个以上变量间关系的统计分析方法。多元线性回归基本的分析方法与一元线性回归方法是类似的，我们首先需要选取多元数据集并定义数学模型，然后进行参数估计，对估计出来的参数进行显著性检验、残差分析、异常点检测，最后确定回归方程进行模型预测。

多元线性回归分析的模型为 $\hat{Y}=b_0+b_1X_1+b_2X_2+\cdots+b_nX_n$。其中 $\hat{Y}$ 为根据所有自变量 X 计算出的估计值，b_0 为常数项，b_1，b_2，…，b_n 称为 Y 对应于 X_1，X_2，…，X_n 的偏回归系数。偏回归系数表示假设在其他所有自变量不变的情况下，某一个自变量变化引起因变量变化的比率。多元线性回归方程中的回归系数和常数项同样可以利用最小二乘法来计算出来，但是这个模型是否恰当，也需要进行假设测验。

由于多元回归方程有多个自变量，区别于一元回归方程，有一项很重要的操作就是自变量的优化，挑选出相关性最显著的自变量，同时去除不显著的自变量。在 R 语言中，优化函数可以很好地帮助我们来改进回归模型。

在 R 中，常用的优化函数是 step()。其一般格式为：

```
step(object, scope, scale = 0,
     direction = c("both", "backward", "forward"),
     trace = 1, keep = NULL, steps = 1000, k = 2, ...)
```

object：这是一个必需的参数，它是一个已经拟合好的线性模型对象，通常是通过 lm()函数拟合得到的。step() 函数会基于这个初始模型开始逐步回归过程。

scope：用于指定逐步回归过程中模型选择的范围。它可以是一个公式或者一个包含上界和下界模型的列表。如果是公式，它定义了所有可能包含在模型中的自变量；如果是列表，scope 应该包含两个元素：upper 和 lower，分别表示最大模型和最小模型。例如，

scope=list(upper=y~x1+x2+x3, lower=y~1) 表示最大模型包含自变量 x1、x2 和 x3,最小模型只包含截距项。

scale:该参数用于计算 AIC 时的尺度因子。在某些情况下,当模型有特定的分布假设时,需要指定这个尺度参数。默认值为 0,表示根据模型自动计算尺度。

direction:指定逐步回归的方向,有三种可选值。"both"表示双向逐步回归,即既可以添加自变量,也可以删除自变量。这是最常用的方法,它结合了前向选择和后向消除的优点。"backward"表示后向消除法,从全模型开始,逐步删除对模型贡献最小的自变量,直到无法再通过删除自变量降低准则值为止。"forward"表示前向选择法,从只包含截距项的模型开始,逐步添加对模型贡献最大的自变量,直到无法再通过添加自变量降低准则值为止。默认值为"both"。

trace:控制是否显示逐步回归过程中的详细信息。如果 trace=0,则不显示任何信息;如果 trace=1(默认值),则显示每一步的模型信息和准则值;如果 trace>1,则显示更详细的信息。

keep:这是一个可选的函数,用于指定在逐步回归过程中需要保留的信息。例如,可以定义一个函数来提取每一步模型的某些统计量,如残差平方和等。默认值为 NULL,表示不保留额外信息。

steps:指定逐步回归过程的最大步数。默认值为 1 000,通常这个值足够大,以确保算法能够收敛到一个合适的模型。

k:用于计算信息准则的惩罚因子。当 k=2 时,使用的是赤池信息准则(AIC);当 k=log(n)(其中 n 是样本量)时,使用的是贝叶斯信息准则(BIC)。默认值为 2,即使用 AIC。

step()函数返回值是一个经过逐步选择变量后得到的优化线性回归模型对象,此对象同样属于 lm 类(若初始模型为线性回归模型)。该返回对象里包含了在逐步选择过程中确定的最优变量组合所对应的模型系数、残差、拟合值等信息,与 lm()函数返回对象结构类似。同时,借助此返回对象,能够进一步对经过变量筛选后的模型进行评估、预测等操作,例如使用 summary()函数查看模型概要,使用 predict()函数进行预测。

例 6-7 测定"丰产 3 号"小麦的每株穗数(X_1)、每穗结实小穗数(X_2)、百粒重(X_3,克)、株高(X_4,厘米)和每株籽粒产量(Y,克)的关系,得结果于表 6-4。试进行最优多元线性回归分析。

表 6-4 例 6-7 的数据

X_1	X_2	X_3	X_4	Y
10	23	3.6	113	15.7
9	20	3.6	106	14.5
10	22	3.7	111	17.5
13	21	3.7	109	22.5
10	22	3.6	110	15.5
10	23	3.5	103	16.9
8	23	3.3	100	8.6

续　表

X_1	X_2	X_3	X_4	Y
10	24	3.4	114	17
10	20	3.4	104	13.7
10	21	3.4	110	13.4
10	23	3.9	104	20.3
8	21	3.5	109	10.2
6	23	3.2	114	7.4
8	21	3.7	113	11.6
9	22	3.6	105	12.3

代码 6－7　多元线性回归分析

```
> example6_7 <- read.csv("E:/RinBio/Chapt6/Example6_7.csv", header = TRUE, sep = ",")
> fit1 <- lm(y~x1 + x2 + x3 + x4,data = example6_7)
> summary(fit1)

Call:
lm(formula = y~x1 + x2 + x3 + x4, data = example6_7)

Residuals:
      Min       1Q       Median       3Q       Max
      -1.70862 -0.84450  0.06217    0.92980    1.74968

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    -51.9021    13.3518    -3.887    0.00302 **
x1             2.0262      0.2720     7.448     2.19e-05 ***
x2             0.6540      0.3027     2.161     0.05606
x3             7.7969      2.3328     3.342     0.00746 **
x4             0.0497      0.0830     0.599     0.56264
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ''1

Residual standard error: 1.357 on 10 degrees of freedom
Multiple R-squared:  0.9232,Adjusted R-squared:  0.8925
F-statistic: 30.06 on 4 and 10 DF,  p-value: 1.498e-05

> fit2 <- step(fit1)
Start:  AIC = 13.08
y~x1 + x2 + x3 + x4
```

```
       Df Sum of Sq     RSS     AIC
- x4     1     0.660    19.078    11.607
<none>                  18.418     13.079
- x2     1     8.597    27.015    16.825
- x3     1     20.574   38.992    22.329
- x1     1     102.168  120.586   39.265

Step:  AIC = 11.61
y~x1 + x2 + x3

       Df Sum of Sq     RSS     AIC
<none>                  19.078   11.607
- x2    1     9.269     28.347   15.547
- x3    1     20.762    39.840   20.652
- x1    1     101.508   120.586 37.265
> summary(fit2)

Call:
lm(formula = y~x1 + x2 + x3, data = example6_7)

Residuals:
    Min       1Q         Median       3Q         Max
    - 1.8953  - 0.6871   0.1562       0.9166     1.7140

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     - 46.9664    10.1926     - 4.608     0.000755 ***
x1              2.0131       0.2631      7.650       9.97e - 06 ***
x2              0.6746       0.2918      2.312       0.041170 *
x3              7.8302       2.2631      3.460       0.005334 **
- - -
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ''1

Residual standard error: 1.317 on 11 degrees of freedom
Multiple R - squared:  0.9205,Adjusted R - squared:  0.8988
F - statistic: 42.44 on 3 and 11 DF,  p - value: 2.445e - 06
```

本例中，首先使用 read.csv 函数从“E:/RinBio/Chapt6/Example6_7.csv”文件读取数据，读取结果存入 example6_7 数据框；运用 lm()函数构建一个包含所有自变量的初始多元线性回归模型 fit1，因变量为每株籽粒产量 y，自变量有每株穗数 x1、每穗结实小穗数 x2、百粒重 x3 和株高 x4；随后用 summary()函数输出 fit1 的详细信息，以初步评估该模

型；使用 step()函数对 fit1 进行逐步回归，筛选出最优变量组合，得到优化后的模型 fit2；最后，再次使用 summary()函数输出 fit2 的详细摘要，以此分析最优多元线性回归模型的拟合效果和各变量影响。

逐步回归分析是以 AIC 信息统计量为准则，通过选择最小的 AIC 信息统计量，来达到删除或增加变量的目的。结果显示，当用 x1、x2、x3、x4 作为回归方程的系数时，AIC 的值为 13.08，去掉 x4 回归方程的 AIC 值为 11.61，去掉 x4 之后所有的自变量在回归方程中都达到了显著水平，逐步回归分析终止。根据线性回归分析结果，则最优多元线性回归方程为：$\hat{y}=-46.966\,4+2.013\,1x_1+0.674\,6x_2+7.830\,2x_3$，并且常数项和回归系数的显著性测验均达到了显著水平，表明小麦的每株穗数、每穗结实小穗数、百粒重均对每株籽粒产量有显著影响。此线性回归方程的决定系数为 0.920 5。回归模型的显著性 $p=2.445\text{e}-06<0.01$，说明回归模型具有极显著的统计意义。

6.4　非线性回归

非线性回归是指在因变量与一组自变量之间建立非线性模型。这里的“线性”和“非线性”并非指因变量与自变量之间的直线关系和曲线关系，而是指因变量能否表示为自变量的线性组合。如果变量关系经过转换可以表达为线性关系，则可以应用线性回归的方法，而如果变量关系经过转换后不能表达为线性关系，就需要用到非线性回归分析方法。在 R 中，nls()函数可用于非线性拟合，其作用是根据给定的数据，找到非线性函数中参数的最优估计值，使得观测值与模型预测值之间的残差平方和最小，从而实现对非线性关系的建模和分析。其一般格式为：

```
nls(formula, data = parent.frame(), start, control = nls.control(),
    algorithm = c("default", "plinear", "port"), trace = FALSE, subset,
    weights, na.action, model = FALSE, lower = - Inf, upper = Inf)
```

formula：指定非线性模型的公式，其形式与线性模型公式类似，但使用的是非线性函数。例如，y～a * exp(b * x) 表示一个指数形式的非线性模型，其中 y 是因变量，x 是自变量，a 和 b 是待估计的参数。

start：是一个命名的数值向量，为模型中的参数提供初始值。由于非线性最小二乘估计是一个迭代过程，需要从初始值开始逐步搜索最优解，合适的初始值对于算法的收敛至关重要。初始值的选择通常需要根据数据的特点和对模型的先验知识来确定。

data：一个数据框(data.frame)，包含公式中所涉及的变量的数据。默认值为 parent.frame()，即从调用函数的环境中查找数据。

control：用于设置 nls() 函数的控制参数，通常使用 nls.control() 函数生成。nls.control() 可以设置诸如最大迭代次数(maxiter)、收敛阈值(tol)等参数，以控制迭代过程的行为。例如，control = nls.control(maxiter = 100) 表示将最大迭代次数设置为 100。

algorithm：指定拟合模型所使用的算法，有三种可选值："default"为默认算法，使用

高斯-牛顿法和列文伯格-马夸尔特法的组合，适用于大多数情况；"plinear"为部分线性算法，用于处理模型中部分参数是线性的情况，可以提高计算效率；"port"为 PORT 库中的算法，该算法对于一些复杂的模型可能更稳定。默认值为"default"。

trace：逻辑值，默认为 FALSE。如果设置为 TRUE，则在迭代过程中会输出每次迭代的详细信息，包括参数值、残差平方和等，有助于调试和监控算法的收敛情况。

subset：一个向量或表达式，用于指定从数据中选择特定的观测值来拟合模型。例如，subset＝x＞0 表示只使用自变量 x 大于 0 的观测值进行拟合。

weights：一个与数据集中观测值数量相同的向量，用于为每个观测值指定权重。加权最小二乘法中，权重用于调整每个观测值对参数估计的影响程度。

na.action：指定处理数据中缺失值(NA)的方法。

model：逻辑值，默认为 FALSE。如果设置为 TRUE，则返回的模型对象中会包含模型矩阵和响应变量；如果为 FALSE，则不包含这些信息。

lower 和 upper：分别指定参数的下界和上界。可以是数值向量，用于对参数的取值范围进行约束。默认值分别为 －Inf 和 Inf，表示参数没有取值范围限制。

nls()函数返回值是一个 nls 类的对象。该对象包含了拟合模型的关键信息，如模型参数的估计值，展示了使模型与数据达到最佳拟合时各参数的取值；残差，即观测值和模型预测值之间的差异，可用于衡量模型拟合的好坏；还有模型公式、数据等信息。可以通过多种方法进一步处理该返回对象，例如使用 summary()函数获取模型的详细统计信息，用 predict()函数基于模型进行预测。

例 6－8　江苏东台测定 1972 年越冬代棉红铃虫随时间(X，天，以 5 月 31 日为 0)变化的化蛹进度(Y，%)，结果见表 6－5，试用 Logistic 生长曲线($\hat{Y}=\frac{K}{1+a\,e^{-bX}}$)来描述 X 和 Y 之间的关系。

表 6－5　越冬代棉红铃虫随时间变化的化蛹进度

时间(X)	化蛹进度(Y，%)	时间(X)	化蛹进度(Y，%)
5	3.5	30	60.4
10	6.4	35	75.2
15	14.6	40	90.2
20	31.4	45	95.4
25	45.6	50	97.5

代码 6－8　非线性回归分析

```
> example6_8 <- read.csv("E:/RinBio/Chapt6/Example6_8.csv", header = TRUE)
> fit <- nls(y~k/(1 + a *exp(- b *x)),start = list(a = 1, b = 0.1,k = 100), data = example6_8)
> summary(fit)

Formula: y~k/(1 + a *exp(- b *x))
```

```
Parameters:
   Estimate Std. Error t value Pr(>|t|)
a    44.14367    8.33833    5.294    0.00113 **
b    0.14100     0.00848    16.628   6.95e-07 ***
k    101.99295   2.40785    42.358   1.07e-09 ***
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1''1

Residual standard error: 2.203 on 7 degrees of freedom

Number of iterations to convergence: 10
Achieved convergence tolerance: 1.925e-06

> plot(example6_8$x,example6_8$y)
> lines(example6_8$x,y=predict(fit),col="blue")
```

本例中，首先使用 read.csv()函数从“E:/RinBio/Chapt6/Example6_8.csv”文件读取数据，读取的数据存入 example6_8 数据框；利用 nls()函数进行非线性最小二乘拟合，构建 Logistic 生长曲线模型 fit，模型公式为 $y=\frac{k}{1+a\times\exp(-bx)}$，并给定初始参数值 $a=1$，$b=0.1$，$k=100$，数据来源于 example6_8；然后，用 summary()函数输出模型 fit 的详细信息，评估模型拟合情况；之后，使用 plot()函数绘制 example6_8 中时间 x 和化蛹进度 y 的散点图；最后，使用 lines()函数在散点图上绘制蓝色的拟合曲线，曲线数据通过 predict(fit)得到，直观展示 Logistic 生长曲线对数据的拟合效果。

此例中，由 summary()函数展示分析结果，Formula 项展示了拟合公式，Parameters 项给出了非线性回归模型中参数的估计值。具体给出了各个参数的估值(Estimate)、标准误(Std. Error)、t 值(t value)和显著性 p 值(Pr(>|t|))。三个参数 a、b、k 的估计值分别为 44.14、0.14 和 102.00，各参数的显著性测验均达到了显著水平，可以得出非线性方程的拟合结果为 $\hat{y}=\frac{102}{1+44.143\,7\mathrm{e}^{-0.141x}}$。此外，由 plot()函数绘制散点图，利用 lines()函数添加回归曲线，非线性回归方程的拟合结果如图 6-5 所示。

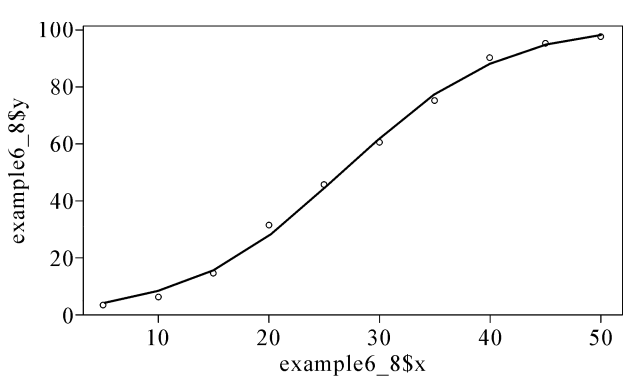

图 6-5　越冬代棉红铃虫的化蛹进度(Y)随时间(X)变化的关系

6.5 二分类变量 Logistic 回归

在实际科研实践中，人们常常要研究某一随机事件 A 发生的概率与某些因素之间的关系。例如在农业生产实践中，要研究农药的使用剂量与某种害虫的死亡率之间的关系；在医学研究中，要考察人们的某些生活习惯、生存环境等因素与某种疾病的发病率之间的关系。然后我们在考察害虫的死亡时，其只表现出两种类型，要么害虫死亡（计作 1），要么还存活（计作 0）；而发病与否也仅仅是两种类型。这类数据被称为二分类变量。若依变量为二分类变量，则在进行回归分析的时候需要采用二分类 Logistic 回归模型加以分析。

对于二分类变量的数据往往可以用 0 和 1 两个数值表示。二分类 Logistc 回归模型要解决的问题是对依变量中二值取一的概率建模而不是直接预测其取值。在一般的多元回归中，若以 P（概率）为依变量，则方程为 $P=b_0+b_1X_1+b_2X_2+\cdots+b_kX_k$，但使用该方程计算时，常会出现概率值 $P>1$ 或 $P<0$ 的不合理情况，为此，对 P 做对数单位转换，令 $\text{logit}P=\ln(P/(1-P))$，可得 Logistic 回归方程为

$$P=\frac{e^{b_0+b_1X_1+\cdots+b_kX_k}}{1+e^{b_0+b_1X_1+\cdots+b_kX_k}}$$

拟合二分类变量的 Logistic 回归模型的参数实际上就是拟合线性模型的参数。通常采用最大似然法来估计参数。

在 R 中，logistic 回归分析可以通过调用广义线性回归模型函数 glm()来实现。其一般格式为：

```
glm(formula, family = gaussian, data, weights, subset,
    na.action, start = NULL, etastart, mustart, offset,
    control = list(...), model = TRUE, method = "glm.fit",
    x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, ...)
```

formula：指定用于拟合的模型公式，类似于 lm()中的用法；

family：指定描述干扰项的概率分布和模型的连接函数，默认值为 gaussian，若需进行 logistic 回归，则需设置为 binomial(link = "logit")；

data：指定用于回归的数据对象，可以是数据框、列表或能被强制转换为数据框的数据对象；

weights：一个向量，用于指定每个观测值的权重；

subset：一个向量，指定数据中需要包含在模型中的观测值；

na.action：一个函数，指定当数据中存在缺失值时的处理办法；

start：一个数值型向量，用于指定线性预测器中参数的初始值；

etastart：一个数值型向量，用于指定线性预测器的初始值；

mustart：一个数值型向量，用于指定均值向量的初始值；

offset：指定用于添加到线性项中的一组系数恒为 1 的项；

control：指定控制拟合过程的参数列表，其中 epsilon 表示收敛的容忍度，maxit 表示迭代的最大次数，trace 表示每次迭代是否打印具体信息；

model：逻辑值，指定是否返回模型框架，默认值为 TRUE；

method：指定用于拟合的方法，“glm.fit”表示用于拟合，"model.frame"表示可以返回模型框架；

x：逻辑值，指定是否返回模型矩阵，默认值为 FALSE；

y：逻辑值，指定是否能够返回响应变量，默认值为 TRUE；

contrasts：模型中因子对照的列表。

glm()函数返回值是一个 glm 类的对象。此对象涵盖了模型拟合的众多关键信息，包含模型系数估计值，体现各自变量对响应变量的影响；残差，反映观测值与模型预测值的差异，用于评估拟合优度；拟合值，是模型依据自变量计算得出的响应变量预测结果；还存有模型公式、所用数据、分布族信息等。借助通用函数如 summary()可查看模型详细统计信息，predict()能基于模型进行预测，从而深入分析和利用该模型。

例 6－9　现有 34 名肺癌病人的生存资料（表 6－6）。其中 $x1$：生活行动能力评分（1～100）；$x2$：病人年龄；$x3$：由诊断到进入研究时间（月）；$x4$：肿瘤类型（“0”表示鳞癌、“1”表示小型细胞癌、“2”表示腺癌、“3”表示大型细胞癌）；$x5$：两种化疗方法（“1”表示常规方法、“0”表示试验新法）；y：病人的生存时间（“0”表示生存时间短，即生存时间小于 200 天；“1”表示生存时间长，即生存时间大于或等于 200 天）。具体试验数据如表 6－6 中所示，试用 Logistic 分析病人的生存时间长短的概率与 $x1$、$x2$、$x3$、$x4$、$x5$ 之间的关系。

表 6－6　34 名肺癌病人的生存资料

$x1$	$x2$	$x3$	$x4$	$x5$	y	$x1$	$x2$	$x3$	$x4$	$x5$	y
40	69	10	1	0	0	60	37	13	1	1	0
40	63	58	2	1	0	90	54	12	1	0	1
70	48	9	2	1	0	50	52	8	1	0	1
70	48	11	2	1	0	70	50	7	1	0	1
80	63	4	2	1	0	20	65	21	1	0	0
60	63	14	2	1	0	80	52	28	1	0	1
30	53	4	2	1	0	60	70	13	1	0	0
80	43	12	2	1	0	50	40	13	1	0	0
40	55	2	2	1	0	70	36	22	2	0	0
60	66	25	2	1	1	40	44	36	2	0	0
40	67	23	2	1	0	60	50	22	3	0	0
20	61	19	3	1	0	80	62	4	3	0	0

续 表

$x1$	$x2$	$x3$	$x4$	$x5$	y	$x1$	$x2$	$x3$	$x4$	$x5$	y
50	63	4	3	1	0	70	68	15	0	0	0
50	66	16	3	1	0	30	39	4	0	0	0
40	68	12	3	1	0	60	49	11	0	0	0
80	41	12	0	1	1	80	61	10	0	0	1
70	53	8	0	1	1	70	67	18	0	0	1

代码 6-9 二分类变量 Logistic 回归分析

```
> example6_9 <- read.csv("E:/RinBio/Chapt6/Example6_9.csv", header = TRUE)
> fit1 <- glm(y~x1 + x2 + x3 + x4 + x5, binomial(link = "logit"),data = example6_9)
> summary(fit1)

Call:
glm(formula = y~x1 + x2 + x3 + x4 + x5, family = binomial(link = "logit"),
    data = example6_9)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -6.36122    4.38550  -1.451   0.1469
x1             0.08735    0.04289   2.037   0.0417 *
x2             0.01593    0.05182   0.307   0.7586
x3             0.04510    0.05894   0.765   0.4442
x4            -1.37780    0.66401  -2.075   0.0380 *
x5            -0.09662    1.14889  -0.084   0.9330
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 39.299  on 33  degrees of freedom
Residual deviance: 22.641  on 28  degrees of freedom
AIC: 34.641

Number of Fisher Scoring iterations: 6

> fit2 <- step(fit1)
```

```
Start:  AIC = 34.64
y~x1 + x2 + x3 + x4 + x5

       Df Deviance    AIC
- x5   1    22.648    32.648
- x2   1    22.736    32.736
- x3   1    23.149    33.149
<none>     22.641    34.641
- x4   1    28.823    38.823
- x1   1    29.719    39.719

Step:  AIC = 32.65
y~x1 + x2 + x3 + x4
        Df Deviance    AIC
- x2    1   22.747     30.747
- x3    1   23.175     31.175
<none>     22.648     32.648
- x4    1   29.438     37.438
- x1    1   29.728     37.728

Step:  AIC = 30.75
y~x1 + x3 + x4

        Df Deviance    AIC
- x3    1   23.306     29.306
<none>     22.747     30.747
- x4    1   29.654     35.654
- x1    1   29.788     35.788

Step:  AIC = 29.31
y~x1 + x4

        Df Deviance    AIC
<none>     23.306     29.306
- x1    1   30.027     34.027
- x4    1   30.038     34.038
> summary(fit2)

Call:
glm(formula = y~x1 + x4, family = binomial(link = "logit"),
    data = example6_9)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.65640   2.76580  -1.684  0.0923.
x1            0.08271   0.04027   2.054  0.0400 *
x4           -1.35835   0.60922  -2.230  0.0258 *
---
Signif. codes:  0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 39.299  on 33  degrees of freedom
Residual deviance: 23.306  on 31  degrees of freedom
AIC: 29.306

Number of Fisher Scoring iterations: 6
```

本例中，首先使用 read.csv()函数从“E:/RinBio/Chapt6/Example6_9.csv”文件读取数据，读取的数据存于 example6_9 数据框；利用 glm()函数构建广义线性模型 fit1，以病人生存时间 y 为因变量，生活行动能力评分 $x1$、病人年龄 $x2$、由诊断到进入研究时间 $x3$、肿瘤类型 $x4$ 和化疗方法 $x5$ 为自变量，指定二项分布族并使用对数链接函数，数据源自 example6_9；然后，用 summary 函数输出 fit1 的详细信息，初步评估该模型；之后，通过 step()函数对 fit1 进行逐步回归，筛选出最优变量组合得到优化后的模型 fit2；最后，再次使用 summary()函数输出 fit2 的详细摘要，以分析最优 Logistic 回归模型下各变量对病人生存时间长短概率的影响。

summary()函数展示的分析结果显示，Coefficients 项给出了回归模型中各自变量的估值(Estimate)、标准误(Std. Error)、z 值(z value)和显著性 p 值(Pr(v>|z|))。该模型中 $x2$(病人年龄)、$x3$(诊断到进入研究时间)和 $x5$(化疗方法)三个自变量对于模型贡献未达显著水平，因此利用逐步回归优化模型，新模型的 AIC 值由原来的34.641降为 29.306，最终的模型包含 $x1$(生活行动能力评分)和 $x4$(肿瘤类型)两个自变量，各自回归系数显著性测验的概率值均小于 0.05，达显著水平。由此可以得出 Logistic 方程为

$$P=\frac{e^{(-4.656\,40+0.082\,71X_1-1.358\,35X_4)}}{1+e^{(-4.656\,40+0.082\,71X_1-1.358\,35X_4)}}$$

习 题

6－1　施化肥量对水稻产量影响的试验数据见下表，试进行线性相关和回归分析。

施化肥量 x(kg)	15	20	25	30	35	40	45
水稻产量 y(kg/亩)	330	345	365	405	445	450	455

6－2　在四川白鹅的生产性能研究中，得到如下一组关于雏鹅重(g)与 70 日龄重(g)的数据，试对其进行线性相关和回归分析。

编号	1	2	3	4	5	6	7	8	9	10
雏鹅重(X)	80	86	98	90	120	102	95	83	113	105
70 日龄重(X)	2 350	2 400	2 750	2 500	3 150	2 680	2 630	2 400	3 080	2 920

6－3　某学校 20 名一年级女大学生体重(kg)、胸围(cm)、肩宽(cm)及肺活量(L)实测值如下表所示，试对影响女大学生肺活量的有关因素做多元回归分析。

编号	体重(kg)	胸围(cm)	肩宽(cm)	肺活量(L)
1	51.3	73.6	36.4	2.99
2	48.9	83.9	34.0	3.11
3	42.8	78.3	31.0	1.91
4	55.0	77.1	31.0	2.63
5	45.3	81.7	30.0	2.86
6	45.3	74.8	32.0	1.91
7	51.4	73.7	36.5	2.98
8	53.8	79.4	37.0	3.28
9	49.0	72.6	30.0	2.52
10	53.9	79.5	37.1	3.27
11	48.8	83.8	33.9	3.10
12	52.6	88.4	38.0	3.28
13	42.7	78.2	30.9	1.92
14	52.5	88.3	38.1	3.27
15	55.1	77.2	31.1	2.64
16	45.2	81.6	30.2	2.85
17	51.4	78.3	36.5	3.16
18	48.7	72.5	30.0	2.51

续 表

编号	体重(kg)	胸围(cm)	肩宽(cm)	肺活量(L)
19	51.3	78.2	36.4	3.15
20	45.2	74.7	32.1	1.92

6-4 测定水稻品种 IR72 籽粒开花后不同天数(X,d)下的平均单粒重(Y,mg),得结果见下表,试用 Logistic 生长曲线方程($\hat{Y}=\frac{K}{1+a\cdot e^{-bX}}$)描述 X 和 Y 之间的关系。

开花后天数(d)	平均单粒重(mg)
0	0.30
3	0.82
6	4.31
9	9.82
12	14.08
15	17.64
18	18.68
21	19.34
24	18.96

6-5 40 位急性淋巴细胞白血病病人,在入院治疗时取得了外周血中的细胞数 X_1(千个/mm^3);淋巴结浸润等级 X_2(分为 0,1,2 级);出院后有无巩固治疗 X_3(1 表示有巩固治疗,0 表示无巩固治疗)。通过随访取得了病人的生存时间,并以变量 Y 表示生存时间(0 表示生存时间在 1 年以内,1 表示生存时间在一年或一年以上)。数据如下表所示,试用 Logistic 模型分析病人生存时间长短的概率与 X_1、X_2、X_3 的关系。

序号	X_1	X_2	X_3	Y	序号	X_1	X_2	X_3	Y
1	2.5	0	0	0	10	6.6	0	0	0
2	173	2	0	0	11	21.4	2	1	0
3	119	2	0	0	12	2.8	0	0	0
4	10	2	0	0	13	5.1	0	1	1
5	502.2	2	0	0	14	2.4	0	0	1
6	4	0	0	0	15	1.7	0	1	1
7	14.4	0	1	0	16	1.1	0	1	1
8	2	2	0	0	17	12.8	0	1	1
9	40	2	0	0	18	21.6	0	1	1

续　表

序号	X_1	X_2	X_3	Y	序号	X_1	X_2	X_3	Y
19	2	0	1	1	30	5.8	0	1	0
20	3.4	2	1	1	31	6.1	0	1	0
21	1.2	2	0	0	32	2.7	2	1	0
22	4	0	1	1	33	4.7	0	0	0
23	5.1	0	1	1	34	128	2	1	0
24	32	0	1	1	35	35	0	0	0
25	1.4	0	1	1	36	2	0	0	1
26	34.7	0	0	0	37	8.5	0	1	1
27	28.4	2	0	0	38	2	2	1	1
28	0.9	0	1	0	39	2	0	1	1
29	30.6	2	0	0	40	4.3	0	1	1