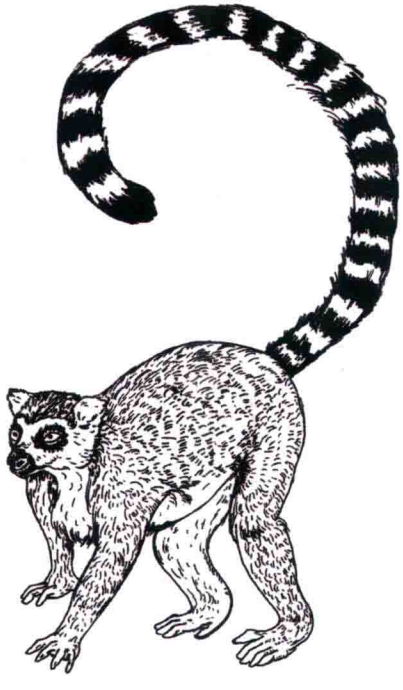
资深R语言技术专家和数据科学工程师撰写，多位专家推荐

17种流行R语言工具包、4个典型综合案例，指导零基础读者迅速掌握R语言并成为数据科学工程师

数据科学与工程技术丛书



Data Science in Action with R

R数据科学实战

工具详解与案例分析

刘健 邬书豪◎著

机械工业出版社
China Machine Press

数据科学与工程技术丛书

R数据科学实战

工具详解与案例分析

刘健 邬书豪◎著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

R 数据科学实战：工具详解与案例分析 / 刘健，邬书豪著 . —北京：机械工业出版社，2019.6
（数据科学与工程技术丛书）

ISBN 978-7-111-62994-8

I. R…　II. ①刘…　②邬…　III. 程序语言—程序设计　IV. TP312

中国版本图书馆 CIP 数据核字（2019）第 122646 号

R 数据科学实战：工具详解与案例分析

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：张锡鹏　　责任校对：张惠兰
印　　刷：北京文昌阁彩色印刷有限责任公司　　版　　次：2019 年 7 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：15.75
书　　号：ISBN 978-7-111-62994-8　　定　　价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　　读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

推荐语

本书不是晦涩难懂的学术教材，而是具备极高实践价值的R语言工具书，每章都针对R语言的核心应用问题进行讲解，对于任何想深度理解R语言及实践应用的爱好者来说，都是一本很好的参考学习书籍，值得推荐。

——黄小伟　有赞数据分析团队负责人
（R语言中文社区创始人、表哥有话讲公众号创始人）

本书是一本少见的深入浅出讲解R语言数据科学的著作。R语言作为基础的数据分析工具，对于数据分析师和数据挖掘工程师来说十分必要，相信任何一个有志于从事数据科学行业的读者，都能从本书中获益。

——张俊红《对比Excel，轻松学习Python数据分析》作者

数据科学的门槛可以很高，也可以很低，数据分析工具的熟练使用非常重要！本书通俗易懂地讲解了R语言当中常用的数据处理工具包的使用和R的核心应用，是R语言爱好者学习数据分析很好的入门教材。

——梁勇　天善智能CEO、Python爱好者社区公众号号主

本书前半部分详细有序地讲解了数据分析各个步骤所需工具包的使用方法，后半部分结合多个案例对前文所述工具包进行综合运用，同时切合案例实际情况对问题进行了逐步剖析和拆解，内容详尽，案例丰富，推荐给大家！

——崔庆才《Python3网络爬虫开发实战》作者、微软小冰工程师

前　言

为什么要写这本书

开始学习和使用R语言，初学者最开始往往会有各种困惑和纠结，可能会走过许多的弯路。和众多初学者一样，我们也深感R语言的学习道路荆棘密布。写这本书的初衷就是希望将我们的经历分享给大家，让学习R语言的道路变得平坦一些，降低初学者使用R语言的难度。

在我们学习交流R语言的过程中，发现最大的挑战是学习资料过剩却不精。另外，国内的技术社区关于R语言的问答内容相对较少。开源的R语言从来不缺免费的学习资料，这当然是好事一件。但凡事总有两面性，因为每个人学习R语言的目的和应用场景都略有不同，很多学习资料初看像是在介绍R语言不同方向的问题或者介绍一些新奇的R包和函数，但是我们发现初学者经常容易花费大量的时间重复阅读相同的概念性问题。比如说使用R语言进行数据清理，不同的数据来源和分析任务可能会让数据清理有上百种可行的方案。在耗费了很多时间尝试这些不同的方法却不得要领时，随之而来的挫败感往往让人心生怯意。所以，我们写下此书，系统性地讲解R语言最流行实用的不同数据运用主题的操作框架，核心是希望让读者能够快速上手并实际运用R语言。

R语言只是万千工具中的一种，熟练掌握工具的各种特性固然重要，但是更重要的是明确任务目标和处理问题的先后顺序。换句话说，使用R语言进行数据分析的首要任务是明确自己的目标，然后围绕该目标建立合理的流程图，其次才是寻找最合适的工具来帮助我们完成每一个具体的任务。所以，最后我们发现万变不离其宗的是清晰的数据分析逻辑。只有当有了自己的数据分析路线图之后，才不会被每天涌现的新的学习资料所淹没，反而是能高效地搜索和应用这些新内容。这也是本书希望传递给读者的信息，R语言则是传递信息的一种媒介。就如同在军事战争中，你有了高级武器，并不一定可以确保你能打败敌人，只有对这些武器有了系统性的认识后，才代表你真正拥有了这些武器。本书就是R语言这件武器的速成手册，希望读者在系统性地认识R语言在数据科学领域中的效力后，降低其在生产环境中的实际运用难度。

读者对象

- 使用 R 语言进行数据处理的 R 语言初学者
- 使用 R 语言进行大数据处理的 R 语言爱好者
- 数据分析师、数据挖掘工程师
- 转型的数据科学人员
- 大中专院校学生

本书特色

本书按照数据分析的一般流程，介绍和讨论了在各个流程中所需的常见的 R 函数，并对其中相对重要的函数做了较为详尽的参数解释和代码演示。相较于大部分 R 语言学习资料中粗略概况性地告知读者不同场景可能用到的 R 函数，本书更侧重于帮助读者建立自己的数据分析逻辑结构以及由一系列常见 R 函数组成的“工具箱”。特别是 tidyverse 系列工具箱和 data.table 包，目前的中文博客社区里很少有资料对这两者进行较为完整和系统的介绍。对于 R 语言初学者来说，tidyverse 系列是学习使用 R 的最佳起点，而 data.table 包则对中高级用户大有助益。另外，本书对重要的“工具”函数，例如循环和迭代，做了较为详尽的解释和代码演示，来帮助读者理解其运行机制。最后，书中提供了 5 个实战案例，结合书中介绍的各种“工具”，强化使用 R 语言进行数据分析的路线图。

如何阅读本书

本书共 11 章，前 6 章（工具包篇）主要介绍和讨论使用 R 语言的一般流程以及常用的 R 包；后 5 章（案例篇）包含了 5 个实战案例，通过与前 6 章的内容相结合，展示如何使用这些 R 包。复现书中的代码需要读者对 .Rproj 有一定的了解，建议读者参阅相关网络教程学会使用 .Rproj。使用 .Rproj 的原因在于其可以将每一次数据分析或练习都视为一个独立的项目（不必调用 setwd 函数重置工作路径），这样做不但可以减少代码出错的几率，而且还能更利于进行数据管理。

对于零基础的 R 语言初学者，建议按照章节顺序进行阅读，尤其是第 1～3 章，介绍了数据分析中相对重要的数据准备阶段。对于有一定基础的 R 语言用户，可以直接阅读自己感兴趣的部分。各章节的简要介绍如下所示。

第 1 章为数据读取，对比介绍不同格式数据读取所需的 R 包，着重介绍平面文档和 Excel 格式文件的读取。

第 2 章为数据清洗，主要介绍 tibble（版本号：1.4.2）和 tidyr（版本号：0.8.0）中常用的函数及其参数设置。

第 3 章为数据计算，主要介绍 dplyr（版本号：0.7.4）中常用的函数及使用技巧。

第 4 章为 R 中的迭代循环，主要介绍基础 for 和 while 循环及 apply 家族函数的运行机制。

第 5 章主要介绍 purrr 包（版本号：0.2.4）的关键函数和运行机制。

第 6 章着重讲解 data.table 包（版本号：1.11.4）的使用技巧。

第 7～11 章为 5 个实战案例，在 ggplot2（版本号：2.2.1）包的配合下，结合前 6 章中的常用函数完整地呈现了一般的数据分析流程和简单的探索性数据分析。5 个案例具体如下：

- 数据科学从业者调查数据集清洗及探索性分析。
- 共享单车数据集初级分析。
- 星巴克店面数量数据集初级分析。
- 学生成绩数据集初级分析。
- YouTube 视频观看数据集处理及初级分析。

本书中的代码内容是在 Rstudio 内完成的，环境参数如下：

```
R version 3.5.0 (2018-04-23)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows >= 8 x64 (build 9200)

Matrix products: default

locale:
[1] LC_COLLATE=Chinese (Simplified)_China.936  LC_CTYPE=Chinese (Simplified)_
      China.936
    LC_MONETARY=Chinese (Simplified)_China.936
[4] LC_NUMERIC=C                                LC_TIME=Chinese (Simplified)_
      China.936
    attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

勘误和支持

由于作者的水平有限，写作时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，特意创建一个在线支持的 GitHub 站点：https://github.com/frank0434/Data-Science-in-Action-R-Tools-and-Case-Studies。我们将尽力在线上为读者提

供最满意的解答。书中的全部源文件都可以从上面的 GitHub 站点下载，我们也会将相应的功能及时更新出来。如果你有更多的宝贵意见，也欢迎发送邮件至邮箱 gong0435@gmail.com，期待能够得到你们的真挚反馈。

致谢

刘健在此感谢我的同事及人生导师 Linley Jesson。是她带我进入 R 语言的世界，并一直鼓励我不断尝试突破自我。是她的耐心指导，让我能够在短时间内熟练掌握 R 语言并应用到工作中解决实际问题。感谢我的父母，将我培养成人。最后感谢我的女儿和妻子，是你们的理解和默默付出让我能够占用陪伴你们的时间来完成大部分书稿。

邬书豪在此感谢我的大学老师徐磊教授 7 年来一直对我的鼓励和支持，是您的引导和启迪让我敢于多多尝试，坚定自己的信念走上了数据科学这条路，您谦谦君子的人格魅力与意志信念给予我人生中巨大的精神力量，感谢您一直与我分享您的待人接物的理念，使我受益匪浅。感谢我的好朋友石楠女士，你对我在数据科学成长道路上的关心、引导，使我坚定地在数据科学道路上解决了安身立命之本，指导我以严谨认真的态度对待工作和生活。感谢我的父母对我的养育与坚定的支持，让我有机会为自己的人生理想打拼，感谢我的领导给予我成长和贡献自己产出的机会，感谢我那些优秀的同事们，与你们一起共事让我成长良多。

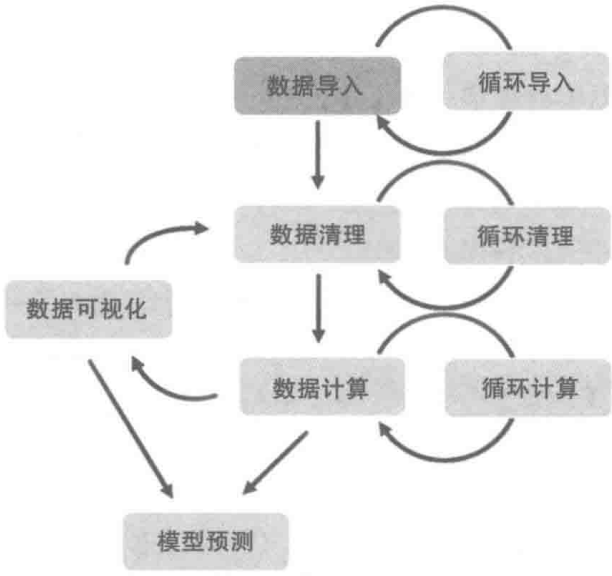
感谢机械工业出版社华章分社的编辑杨福川和张锡鹏，在这一年多的时间中始终支持我们的写作，是你们的理解和支持帮助我们顺利完成全部书稿。

谨以此书献给和我们一样在数据科学领域摸索前行的伙伴，以及众多热爱 R 语言的朋友们！

刘健　邬书豪

目　录

第一部分
Part 1
工具包篇



第 1 章
数据导入工具

无论数据分析的目的是什么，将数据导入 R 中的过程都是不可或缺的。毕竟巧妇难为无米之炊。所以本章主要介绍如何选择合适的包，将不同类型的数据文件导入 R 中。学习完本章的内容之后，读者将会获得以下技能。

1）掌握与数据文件类型相对应的 R 语言数据读取函数。

2）了解常用数据类型读取所需的 R 程序包。

3）了解不同 R 包中相似函数的优缺点。

4）清楚常用数据读取函数的参数设置。

5）能够处理规则及不规则原始数据文件的读取和初步检视。

在描述和讲解如何使用 R 语言各个包的基本方法的同时，本章还会介绍一些笔者曾经踩过的“坑”，以及从中学到的一些小知识点或技巧，希望能让读者在学习过程中避免重蹈覆辙。

1.1　utils——数据读取基本功

utils 包是 R 语言的基础包之一。这个包最重要的任务其实并不是进行数据导入，而是为编程和开发 R 包提供非常实用的工具函数。使用 utils 包来进行数据导入和初步的数据探索也许仅仅只是利用了 utils 包不到 1% 的功能，但这 1% 却足以让你在学习 R 语言时事半功倍。

1.1.1　read.csv/csv2——逗号分隔数据读取

.csv 可能是目前最常见的平面文件类型了。它代表的是 comma-separated values，简单来讲就是，文件里每一个单独的数据值都是用逗号进行分隔的。.csv 只是 text file（文本文件）的一种，文本文件在微软的 Windows 操作系统中常以拓展名为 .txt 的形式呈现。文本文件可以使用各种符号来分隔数据值，例如常见的 tab 和“;”（分号），或者其他任意符号。即便是以 .csv 为拓展名的文件也并非一定是以逗号进行分隔的，相关内容在本章后面的函数演示部分会有介绍。文件的拓展名并非必须，熟悉 Linux 系统的读者可能接触过很多无拓展名的文件。处理无拓展名的文本文件数据时，最简单的办法就是使用 data.table 包中的 fread 函数（相关内容请参见第 6 章）。

utils 里的 read.csv/csv2 是专门用于设置快速读取逗号分隔（read.csv）或是分号分隔（read.csv2）。也就是说，在事先了解数据值分隔符号的情况下，这两个函数对分隔符和其他一些参数的默认设置会使数据导入的部分更加简单和快捷。有一点需要特别注意，即这两个函数对小数点的处理：前者默认的小数点是“.”，后者默认的小数点是“,”。这只是因为不同国家技术人员对数据值分隔符的见解或者好恶不同而造成的。

万里长征第一步，我们先来看 read.csv 最简单的使用方式，代码如下：

```
> flights <- read.csv(file = "flights.csv")
```

此行代码可以解读为使用 read.csv 从工作空间读取文件 flights.csv，然后将数据集保存到 flights 中，其他所有参数都使用默认值。因为 flights.csv 文件已经在 R 的工作路径里，所以此处免去了设置 work directory。这里希望读者能够自

行探索使用 .rproj（R 项目——将每一次数据分析的过程都看作一个独立的项目）来对每一个独立的数据分析工作进行分类和归集。该方法不仅免去了设置路径的麻烦，也减少了因原始数据文件太多而可能导致的各种隐患。

小知识

函数在执行的时候可以依照其默认设置的参数位置来执行，也就是说，用户无须指定每一个参数的名称，只需按照位置顺序来设定参数值即可。比如，read.csv 中的 file 参数名就可以省略，只要第一位是读取文档的目标路径和文件名就可以。

数据文件被读取到 R 工作环境中的第一步通常为调用 str 函数来对该数据对象进行初步检视，下面的代码列出了该函数最简单的使用方式。

```
    > str(object = flights)
    'data.frame':   6 obs. of  6 variables:
$ carrier : Factor w/ 4 levels "AA","B6","DL",..: 4 4 1 2 3 4
$ flight  : int  1545 1714 1141 725 461 1696
$ tailnum : Factor w/ 6 levels "N14228","N24211",..: 1 2 4 6 5 3
$ origin  : Factor w/ 3 levels "EWR","JFK","LGA": 1 3 2 2 3 1
$ dest    : Factor w/ 5 levels "ATL","BQN","IAH",..: 3 3 4 2 1 5
$ air_time: int  227 227 160 183 116 150
```

str 函数可用于检视读取数据结构、变量名称等。这里同样也只指定了一个非默认参数，其他参数全部都为默认值。str 的输出结果由 5 个主要部分组成，具体说明如下。

1）data.frame 代表数据集在 R 中的呈现格式，这里指的是数据框格式，读者可以将其设想为常见的 Excel 格式。

2）6 obs. of 6 variables 代表这个数据集有 6 个变量，每个变量分别有 6 个观测值。

3）$ carrier 与其余带有“$”符号的函数均指变量名称。

4）变量名称冒号后面的 Factor 和 int 代表的是变量类型。这里分别是指因子型 Factor 和整数型 int 数据。另外还有字符型 chr、逻辑型 logi、浮点型 dbl（带有小数点的数字）、复杂型 complex 等。因子型变量的后面还列出了各个变量的因子水平，也就是拥有多少个不同的因子。比如，出发地 origin 后的 3 levels 就是表示其有 3 个因子水平。只是出发地是否属于因子类型的数据还有待商榷，而 read.csv 默认将所有的字符型数据都读成了因子型。

5）数据中的实际观测值。str 函数在默认情况下会显示 10 行数据。使用 str 函数浏

览导入的数据集可以让用户确定读取的数据是否正确、数据中是否有默认的部分、变量的种类等信息，进而确定下一步进行数据处理的方向。其他用来检视数据集的函数还有 head、tail、view 等，另外，Rstudio 中的 Environment 部分也可以用于查看目前工作环境中的数据框或其他类型的数据集。

前文提到过，.csv 并非一定是以逗号进行分隔。如果遇到以非逗号分隔数据值的情况，加之未指定分隔符（例如，运行 read.csv 读取以 Tab 分隔的文件），就会出现下面的情况：

```
    > flights1 <- read.csv(file = "flights1.csv")
    > str(object = flights1)
    'data.frame':   6 obs. of  1 variable:
$ carrier.flight.tailnum.origin.dest.air_time: Factor w/ 6 levels "AA\t1141\
    tN619AA\tJFK\tMIA\t160",..: 4 6 1 2 3 5
```

小技巧

指定（assgin）符号“<-”的快捷键是“alt”加“-”（短划线）。Rstudio 快捷键参照表可以通过“alt+K”来查看详细内容。

由代码可知，read.csv 函数将所有数据都读取到了一列中。因为按照默认的参数设置，函数会寻找逗号作为分隔列的标准，若找不到逗号，则只好将所有变量都放在一列中。指定分隔符参数可以解决这个问题。将 \t（tab 在 R 中的表达方式）指定给 sep 参数后再次运行 read.csv 读取以 Tab 分隔的 csv 文件，代码如下：

```
    > flights3 <- read.csv(file = "flights1.csv", sep = "\t")
    > str(flights3)
    'data.frame':   6 obs. of  6 variables:
$ carrier : Factor w/ 4 levels "AA","B6","DL",..: 4 4 1 2 3 4
$ flight  : int  1545 1714 1141 725 461 1696
$ tailnum : Factor w/ 6 levels "N14228","N24211",..: 1 2 4 6 5 3
$ origin  : Factor w/ 3 levels "EWR","JFK","LGA": 1 3 2 2 3 1
$ dest    : Factor w/ 5 levels "ATL","BQN","IAH",..: 3 3 4 2 1 5
$ air_time: int  227 227 160 183 116 150
```

根据实际情况不同，字符型数据有时会是因子，有时不会。如果使用 read.csv 默认的读取方式，那么字符型全因子化会对后续的处理分析带来很多麻烦。所以最好是将字符因子化关掉。stringsAsFactors 参数就是这个开关，示例代码如下：