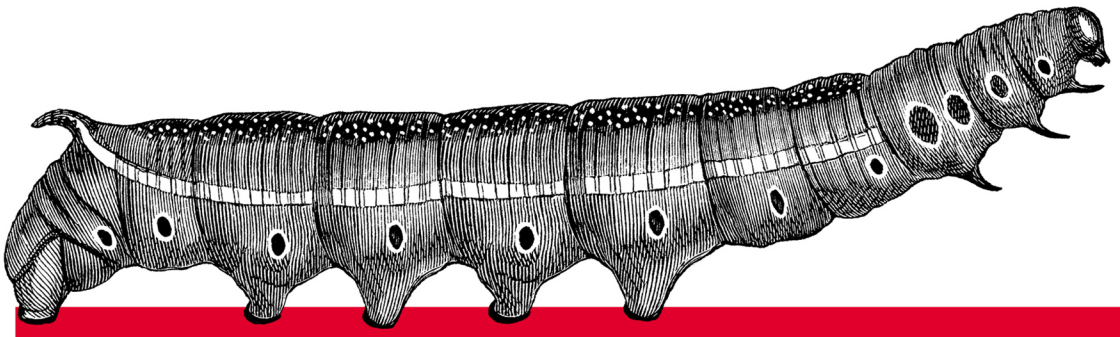
O'REILLY®

TURING 图灵程序设计丛书



Python 数据分析基础

Foundations for Analytics with Python

零编程经验也可学会用最火的Python语言进行数据分析

[美] Clinton W. Brownley 著

陈光欣 译

中国工信出版集团

人民邮电出版社 POSTS & TELECOM PRESS

数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

译者介绍

陈光欣

毕业于清华大学并留校工作，主要兴趣为数据分析与数据挖掘。

TURING 图灵程序设计丛书

Python数据分析基础

Foundations for Analytics with Python

[美] Clinton W. Brownley 著
陈光欣 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo
O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北　京

图书在版编目（CIP）数据

Python数据分析基础 /（美）克林顿·布朗利（Clinton W. Brownley）著 ; 陈光欣译. -- 北京 : 人民邮电出版社, 2017.8
（图灵程序设计丛书）
ISBN 978-7-115-46335-7

Ⅰ. ①P… Ⅱ. ①克… ②陈… Ⅲ. ①软件工具—程序设计 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第165176号

内 容 提 要

本书展示如何用 Python 程序将不同格式的数据处理和分析任务规模化和自动化。主要内容包括：Python 基础知识介绍、CSV 文件和 Excel 文件读写、数据库的操作、示例程序演示、图表的创建，等等。

本书适合数据分析与处理工作相关人员。

◆ 著 [美] Clinton W. Brownley
译 陈光欣
责任编辑 朱 巍
执行编辑 张海艳
责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京 印刷

◆ 开本：800×1000 1/16
印张：17
字数：402 千字 2017年 8 月第 1 版
印数：1－4 000册 2017年 8 月北京第 1 次印刷
著作权合同登记号 图字：01-2017-4510号

定价：69.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

版权声明

O’Reilly Media, Inc.介绍

O’Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O’Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过图书出版、在线服务或者面授课程，每一项 O’Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O’Reilly Radar 博客有口皆碑。”

——*Wired*

“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

献给 Aisha 和 Amaya，

“教育不是把桶灌满，而是将火点燃。”——苏格拉底

愿你们心中之火永远不熄。

目录

前言……xi

第 1 章　Python 基础……1

1.1　创建 Python 脚本……1
1.2　运行 Python 脚本……3
1.3　与命令行进行交互的几项技巧……6
1.4　Python 语言基础要素……10
1.4.1　数值……10
1.4.2　字符串……12
1.4.3　正则表达式与模式匹配……16
1.4.4　日期……19
1.4.5　列表……21
1.4.6　元组……26
1.4.7　字典……27
1.4.8　控制流……30
1.5　读取文本文件……35
1.5.1　创建文本文件……36
1.5.2　脚本和输入文件在同一位置……38
1.5.3　读取文件的新型语法……38
1.6　使用 glob 读取多个文本文件……39
1.7　写入文本文件……42
1.7.1　向 first_script.py 添加代码……42
1.7.2　写入 CSV 文件……45
1.8　print 语句……46
1.9　本章练习……47

第 2 章　CSV 文件 ····· 48
2.1　基础 Python 与 pandas ····· 50
2.1.1　读写 CSV 文件（第 1 部分）····· 50
2.1.2　基本字符串分析是如何失败的 ····· 56
2.1.3　读写 CSV 文件（第 2 部分）····· 57
2.2　筛选特定的行 ····· 58
2.2.1　行中的值满足某个条件 ····· 59
2.2.2　行中的值属于某个集合 ····· 60
2.2.3　行中的值匹配于某个模式 / 正则表达式 ····· 62
2.3　选取特定的列 ····· 64
2.3.1　列索引值 ····· 64
2.3.2　列标题 ····· 65
2.4　选取连续的行 ····· 67
2.5　添加标题行 ····· 69
2.6　读取多个 CSV 文件 ····· 71
2.7　从多个文件中连接数据 ····· 75
2.8　计算每个文件中值的总和与均值 ····· 78
2.9　本章练习 ····· 81
第 3 章　Excel 文件 ····· 82
3.1　内省 Excel 工作簿 ····· 84
3.2　处理单个工作表 ····· 88
3.2.1　读写 Excel 文件 ····· 88
3.2.2　筛选特定行 ····· 92
3.2.3　选取特定列 ····· 98
3.3　读取工作簿中的所有工作表 ····· 101
3.3.1　在所有工作表中筛选特定行 ····· 102
3.3.2　在所有工作表中选取特定列 ····· 104
3.4　在 Excel 工作簿中读取一组工作表 ····· 106
3.5　处理多个工作簿 ····· 108
3.5.1　工作表计数以及每个工作表中的行列计数 ····· 110
3.5.2　从多个工作簿中连接数据 ····· 111
3.5.3　为每个工作簿和工作表计算总数和均值 ····· 113
3.6　本章练习 ····· 117
第 4 章　数据库 ····· 118
4.1　Python 内置的 sqlite3 模块 ····· 119
4.1.1　向表中插入新记录 ····· 124
4.1.2　更新表中记录 ····· 128
4.2　MySQL 数据库 ····· 131
4.2.1　向表中插入新记录 ····· 135

4.2.2 查询一个表并将输出写入 CSV 文件……140
4.2.3 更新表中记录……142
4.3 本章练习……146

第 5 章 应用程序……147

5.1 在一个大文件集合中查找一组项目……147
5.2 为 CSV 文件中数据的任意数目分类计算统计量……158
5.3 为文本文件中数据的任意数目分类计算统计量……167
5.4 本章练习……174

第 6 章 图与图表……175

6.1 matplotlib……175
6.1.1 条形图……175
6.1.2 直方图……177
6.1.3 折线图……178
6.1.4 散点图……180
6.1.5 箱线图……181
6.2 pandas……183
6.3 ggplot……184
6.4 seaborn……186

第 7 章 描述性统计与建模……192

7.1 数据集……192
7.1.1 葡萄酒质量……192
7.1.2 客户流失……193
7.2 葡萄酒质量……194
7.2.1 描述性统计……194
7.2.2 分组、直方图与 *t* 检验……195
7.2.3 成对变量之间的关系和相关性……196
7.2.4 使用最小二乘估计进行线性回归……198
7.2.5 系数解释……200
7.2.6 自变量标准化……200
7.2.7 预测……202
7.3 客户流失……203
7.3.1 逻辑斯蒂回归……205
7.3.2 系数解释……207
7.3.3 预测……208

第 8 章 按计划自动运行脚本……209

8.1 任务计划程序（Windows 系统）……209
8.2 cron 工具（macOS 系统和 Unix 系统）……215

8.2.1 cron 表文件：一次性设置 ····216
8.2.2 向 cron 表文件中添加 cron 任务 ····216
第 9 章 从这里启航 ····220
9.1 更多的标准库模块和内置函数 ····221
9.1.1 Python 标准库（PSL）：更多的标准模块 ····221
9.1.2 内置函数 ····222
9.2 Python 包索引（PyPI）：更多的扩展模块 ····222
9.2.1 NumPy ····223
9.2.2 SciPy ····227
9.2.3 Scikit-Learn ····230
9.2.4 更多的扩展包 ····232
9.3 更多的数据结构 ····232
9.3.1 栈 ····233
9.3.2 队列 ····233
9.3.3 图 ····233
9.3.4 树 ····234
9.4 从这里启航 ····234
附录 A 下载指南 ····236
附录 B 练习答案 ····245
作者介绍 ····247
封面介绍 ····247

前言

本书面向的读者是那些经常使用电子表格软件进行数据处理，但从未写过一行代码的人。前几章会教你设置 Python 运行环境，告诉你计算机是如何看待数据并对其进行简单处理的。你很快就能掌握在电子表格（包括 CSV 文件）和数据库中处理数据的方法。

刚开始，你可能会觉得这样做是一种退步，如果你能熟练使用 Excel，这种感受会更加强烈。以前你只需复制粘贴就能完成的工作，现在却要煞费苦心地告诉 Python 如何在列的每个单元格之间循环，这效率太低了，想想就令人沮丧（特别是当你几次三番地回头去找某一处输入错误的时候）。但是当你逐渐掌握了 Python 之后，就会不断地发现它的真正价值所在，而其中一个极好的例子就是它可以自动完成你现在不断重复的工作。

本书的写作目的是让你全面地掌握 Python，然后充满信心地写出按照你的期望运行的有效代码。一开始输入一些代码或许是个好主意，这样你就会熟悉像制表符、闭括号和引用之类的技术细节。但是，本书中的所有代码在网上都能找到（https://github.com/cbrownley/foundations-for-analytics-with-python）。你在做自己的工作时，完全可以通过复制粘贴来重用这些代码。没关系！适时地进行复制和粘贴也是高效编程的一部分。在阅读本书的同时完成示例程序，会使你更好地理解示例代码的原理。

祝你在成为程序员的道路上好运连连！

为什么要读这本书，为什么要学习这些技能

如果你经常做数据处理工作，就一定会为学习编程而兴奋。学习编程的一个好处是，你可以完成那些靠手工难以完成或者根本不可能完成的数据处理与分析工作。可能你已经遇到了这样的问题：需要处理的文件包含太多数据，以至于打开文件都非常困难或者根本不可行。即使打开了这些文件，手动处理也会花费大量时间，并且极易出错，因为你对数据进行的任何修改都需要很长时间才能更新，而且面对如此多的数据，进行修改时很容易漏掉某一行或某一列。你可能还遇到了其他情况，如需要处理大量的文件，以至于手动处理根本不可能完成。有些时候，你需要的数据来自于几十、几百甚至上千个文件。当所需的文件数量不断增加时，手动处理会变得越来越困难。在以上所有这些情况之下，写一个

Python 脚本来处理文件就可以解决你的问题，因为 Python 脚本可以快速有效地处理大型文件和大批量的文件。

学习编程的另一个好处是，你可以自动地重复数据处理和数据分析过程。在很多情况下，我们针对数据做的都是耗时的重复性工作。例如，一般的数据管理过程是，先从客户或供应商处获取数据，然后提取并保留所需的数据，之后还可能会进行一些数据转换或重新格式化，最后将数据保存到数据库或数据仓库中［这就是数据科学家熟知的数据 ETL（extract、transform、load，即抽取、转换和加载）过程］。类似地，典型的数据分析过程包括数据获取、数据准备、数据分析和结果展示。在数据管理和数据分析过程中，一旦建立了流程，就可以编写 Python 代码来进行各种操作。通过创建 Python 脚本来执行操作，你可以将耗时的重复性工作简化为执行一个脚本，并用节省下来的时间去做其他更有意义的工作。

最重要的是，在进行数据处理和数据分析时，使用 Python 脚本代替手动操作可以减小出错的可能性。手动进行数据处理时，非常可能出现复制粘贴错误或输入错误。导致出错的原因有很多：你可能因过于匆忙而忽略了错误，或者有些事导致你分心了，或者仅是因为你太累了。而且，当你处理大型文件或大批量的文件，或者进行重复性操作时，出错的可能性会更大。相反，Python 脚本从来不会分心或疲劳。一旦你调试好脚本，确认它可以按照你的期望处理数据，它就会一如既往、不知疲倦地工作下去。

最后，学习编程非常有趣，而且能提高自身能力。只要熟悉了基本的语法，你就会非常乐于找到所需的语言功能，然后将它们组合在一起，以完成整体的数据分析目标。至于代码和语法，网上有许多示例可以教会你如何使用专门的功能来完成特定的任务。不过，这些示例虽能提供帮助，但是你需要通过自己的创造力和解决问题的能力来弄清楚如何修改这些代码，以使它们满足你的实际需要。找到合适的代码，并想办法让它们为你工作，这是个非常有意思的过程。此外，学习编程能极大地提高自身的能力。举个例子，考虑一下我前面提到过的情况，即要处理大型文件和大批量文件。如果不会编程，那么你要么需要花费大量时间，要么束手无策。一旦学会了编程，你就可以通过 Python 脚本轻松愉快地解决所有问题。有些数据处理和数据分析任务以前是非常困难或根本不可能完成的，但是现在你都可以轻松搞定，这会使你充满信心，能量爆棚，从而积极主动地寻找更多的机会，使用 Python 来迎接数据处理方面的挑战。

目标读者

本书的目标读者是那些经常从事数据处理工作，又具有极少或根本没有编程经验的人。书中的示例覆盖了常用的数据源和数据格式，包括文本文件、逗号分隔值（CSV）文件、Excel 文件和数据库。在某些情况下，由于文件中数据过多，或由于文件数量太多，造成文件难以打开或不能通过手动处理。在其他一些情况下，从文件中抽取和使用数据的过程非常耗时并且容易出错。在这些情况下，如果你不会编程，就会将大量时间浪费在数据搜索、打开与关闭文件，以及复制和粘贴数据上面。

鉴于你可能从未运行过脚本，本书从最基本的操作开始，介绍如何在文本文件中编写代码以创建 Python 脚本。然后，我们会学习如何通过命令行窗口（Windows 用户）或终端窗口（macOS 用户）来运行 Python 脚本。（如果你做过一点编程，可以跳过第 1 章，直接学习第 2 章中的数据分析内容。）

本书的编写方式特别适合编程新手。书中提供的示例包含了完成某项任务所需的全部Python 代码，而不是仅提供一些代码片段，让你自己将它们组合起来以完成任务。你以后可能会经常使用本书作为参考，而且会发现书中的代码确实有帮助。最后，正所谓“一图胜千言”，书中使用了大量屏幕截图来展示输入文件、Python 脚本、命令行窗口、终端窗口和输出文件，这样你就可以真实地看到如何创建输入、代码、命令和输出了。

我会详细讲解代码的原理，也会推荐一些工具供你使用。这种方法可以帮助你打下坚实的基础，以理解在程序背后到底发生了什么。有时候，你需要在 Google 上搜索问题的解决方案并找到一些有用的代码。做完了书中的练习之后，你可以更好地理解这些代码的工作原理，也就是说，你不但知道如何根据具体情况使用它们，而且知道在出现问题时如何进行修复。因为你在每一章中都会编写一些代码，所以你会发现可以将本书作为参考书，或指导手册，然后在里面找到完成具体任务的方法。但是请记住，这仅是一本“学习如何编程”的书，你还需要不断提高和扩展编程技能，以便综合运用它们来完成各种任务。

为什么使用Windows

本书中的大部分示例都是演示在 Microsoft Windows 系统下如何创建和运行 Python 脚本。将重点放在 Windows 系统上的原因很简单：我想让本书帮助尽可能多的人。根据估计，大多数台式机和笔记本电脑（特别是用于商业分析的计算机）运行的是 Windows 操作系统。例如，根据 Net Applications 的调查，截至 2014 年 12 月，Microsoft Windows 占领了大约 90% 的台式机和笔记本电脑操作系统市场。因为我想让本书满足台式机用户和笔记本电脑用户的需求，而且这些电脑中多数都安装了 Windows 操作系统，所以本书将集中讲述如何在 Windows 系统下创建和运行 Python 脚本。

尽管本书将重点放在了 Windows 上，但在适当情况下，我也提供如何在 macOS 系统上创建和运行 Python 脚本的示例。不论在哪种机器上运行，Python 中几乎所有功能的表现都是一样的。当因为操作系统不同而出现差别时，我会分别给出具体的说明。例如，第 1 章的第一个例子演示了如何在 Microsoft Windows 和 macOS 系统下创建和运行 Python 脚本。类似地，第 2 章和第 3 章的第一个例子也演示了如何在 Windows 和 macOS 系统下创建和运行 Python 脚本。此外，第 8 章涵盖了两种操作系统，介绍了如何在 Windows 中建立计划任务以及如何在 macOS 中建立定时作业。如果你是 Mac 用户，可以使用每章的第一个例子作为模板，来学习如何创建 Python 脚本，如何使其可以执行，以及如何运行脚本，然后重复这些步骤来创建和运行每章中其余的示例程序。

为什么使用Python

如果你的目的是学习一门编程语言来使数据处理和数据分析任务规模化和自动化，那么Python 绝对是一个好的选择。Python 的一个显著特点就是使用空白字符和缩进来表示行的结尾和代码分块，这与很多其他语言不同，其他语言使用特殊字符（比如分号和花括号）来达到同样的目的。Python 的这个特点使你一眼就能看出程序的组织方式。

在其他语言中，特殊字符的使用对于编程新手来说是个困扰，原因至少有两个。第一，这使得学习曲线更长并且更加陡峭。当你学习编程时，实质上是在学习一门新的语言，你必

须花时间学习这些特殊字符的用法，然后才能有效地使用这门语言。第二，特殊字符使代码难以阅读。这是因为在使用分号和花括号表示代码块的语言中，并不总是使用缩进来标明代码块。如果没有缩进，多个代码块看上去就是乱七八糟的。

Python 使用空白字符和缩进来表示代码分块，而不使用分号和花括号，这样就避免了上述问题。当你阅读 Python 代码时，你的视线会集中在实际的代码行上，而不是代码块的分隔符上，因为代码周围只有空白字符。Python 要求代码块必须缩进，这样你会很容易看出代码块在哪里结束，新的代码块又在哪里开始。而且，Python 社区特别强调代码的可读性，因此已经形成了一种文化，就是一定要书写易于阅读和理解的代码。Python 的这些特点使学习曲线更短并且更加平坦，与其他语言相比，使用 Python 进行数据处理可以更快也更容易上手。

Python 适用于数据处理与分析的另一个显著特点，是其具有大量的标准模块、附加模块以及函数，可以非常方便地完成一般的数据处理与分析操作。内建库和标准库中的模块和函数是 Python 的标准配置，所以只要你下载并安装了 Python，就可以立即使用这些内建的模块和函数。在 Python 标准库页面（https://docs.python.org/3/library）中，你可以找到所有内建模块和标准模块的介绍。Python 附加模块需要单独下载并安装，然后才能使用它们提供的附加功能。你可以在 Python 程序包索引页面（https://pypi.python.org/pypi）详细查看很多附加模块的介绍。

标准库中的模块提供的功能包括读取各种类型的文件（如文本文件、CSV、JSON、HTML、XML 等），处理数值、字符串和日期型数据，使用正则表达式进行模式匹配，解析 CSV 文件，计算基本的统计量，以及向各种类型的输出文件和磁盘写入数据。有用的附加模块太多，无法一一介绍。本书要讨论和使用的附加模块如下所示。

- `xlrd` 和 `xlwt`

  功能：解析与读写 Microsoft Excel 工作簿。

- `mysqlclient/MySQL-python/MySQLdb`

  功能：连接 MySQL 数据库，在数据库表上运行查询。

- `pandas`

  功能：读取各种类型的文件；管理、筛选和转换数据；聚合数据并计算基本统计量；创建各种类型的统计图表。

- `statsmodels`

  功能：估计各种统计模型，包括线性回归模型、广义线性模型和分类模型。

- `scikit-learn`

  功能：估计机器学习统计模型，包括回归、分类和聚类，以及执行数据处理、维度归约和交叉验证。

如果你是编程新手，并且正在寻找一门可以使数据处理与分析任务自动化和规模化的编程语言，那么 Python 就是理想的选择。Python 对于空白字符和缩进的强调使代码更易于阅读和理解，因而和其他语言相比，它的学习曲线没有那么陡峭。Python 的内建库和附加库可以方便地完成许多一般的数据处理和分析操作，让你可以轻松地一站式完成数据处理与分析任务。