大数据应用与技术丛书

Pandas 数据分析实战

[美] 鲍里斯 • 帕斯哈弗(Boris Paskhaver) 著

殷海英 译

清華大學出版社

北 京

北京市版权局著作权合同登记号 图字：01-2022-1217

Boris Paskhaver
Pandas in Action
EISBN: 978-1-61729-743-4

图书在版编目(CIP)数据

Pandas数据分析实战 / (美) 鲍里斯 • 帕斯哈弗(Boris Paskhaver) 著；殷海英译. —北京：清华大学出版社，2022.8
(大数据应用与技术丛书)
书名原文：Pandas in Action
ISBN 978-7-302-61271-1

Ⅰ. ①P… Ⅱ. ①鲍… ②殷… Ⅲ. ①数据处理 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字(2022)第 120016 号

责任编辑：王 军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：丛怀宇

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-83470000 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：大厂回族自治县彩虹印刷有限公司
经 销：全国新华书店
开 本：170mm×240mm **印 张**：24 **字 数**：613 千字
版 次：2022 年 8 月第 1 版 **印 次**：2022 年 8 月第 1 次印刷
定 价：128.00 元

产品编号：095364-01

译 者 序

目前，“数据科学”是一个非常热门的词，而“数据科学家”逐渐成为学生的理想职业。在数据爆炸的今天，人们的生活、工作中充斥着大量数据，但这些数据中，只有少部分具有价值。想在海量数据中找出有价值的数据，不但需要计算机编程的能力，还需要很多其他学科的专业知识，这就是近些年选择“数据科学 101”这门课的学生中出现了好多经济类、管理类，甚至医学类专业学生的原因。在一些看似与计算机关系不大的领域，数据科学也发挥着重要的作用。

从事数据科学工作的时候，大部分时间都在处理各种各样的数据，因为现实世界中的数据不像我们想象的那么完美，往往存在各种缺失、重复，甚至错误的情况。比如，通过数据分析可能会发现一个客户的同一个订单被配送多次，从海洋浮标上采集的海水温度是 2500℃，等等。如何有效地处理采集到的大量数据？Pandas 是再好不过的选择了。

在数据科学的教学过程中，关于数据处理的讲解一般只安排了 20%~30%的时间，每次讲完都会觉得还有好多信息要和学生分享，但无奈时间有限。很高兴清华大学出版社的王军老师向我推荐了一本书——*Pandas in Action*，在 Manning 的网站上看过该书之后，我就将这本书推荐给选修“数据科学 101”这门课的同学，告诉他们有一个“熊猫大侠”可以帮助大家解决各种让人头痛的数据处理问题。

很高兴清华大学出版社可以引进这本书到中国，让我有幸翻译这本书。在这里要感谢清华大学出版社的王军老师，感谢他帮助我出版了十多本关于数据科学和高性能计算的书籍；也要感谢我的学生对我的支持，尤其是经济类、管理类和医学类专业的学生，是他们让我在不惑之年仍然对之前不了解的领域保持好奇心，并和他们一起，利用高性能计算及数据科学技术探索计算机视觉在医疗领域的应用；最后要感谢我的弟弟殷实先生、弟媳及侄子、侄女在生活中为我带来的欢乐，让我在轻松的状态下完成本书。

殷海英
埃尔赛贡多，加利福尼亚州
2022 年 3 月

作 者 简 介

Boris Paskhaver 目前居住在纽约市，是一位资深的全栈软件工程师、顾问和在线教育者。他在电子学习平台 Udemy 上开设了 6 门课程，拥有超过 140 小时的视频、30 万名学生、2 万条评论，视频每月有 100 万分钟的浏览量。在成为软件工程师之前，Boris 是一名数据分析师和系统管理员。他于 2013 年毕业于纽约大学，获得商业经济学和市场营销双学位。

致　谢

为了完成《Pandas 数据分析实战》，我付出了很多努力，我想对那些在两年的写作过程中支持我的人表达最诚挚的谢意。

首先，衷心感谢我出色的女友 Meredith。从一开始，她就坚定地支持我。她是一个活泼、有趣、善良的人，每当我遇到困难时，她总是给我打气。本书的成功离不开她的支持。谢谢你，Merbear。

谢谢我的父母 Irina 和 Dmitriy，给了我一个温暖的家，让我可以在这里找到喘息的机会。

谢谢我的姐妹 Mary 和 Alexandra。她们聪明、充满好奇心、勤奋，我为她们感到自豪。祝你们在大学好运！

感谢 Watson，我们的金毛猎犬。它算不上 Python 专家，但它以有趣和友好的方式弥补了这一缺陷。

非常感谢我的编辑 Sarah Miller，与她一起工作非常愉快。我很感激她在本书出版过程中的耐心和洞察。她是这艘船真正的船长，她让一切得以顺利进行。

如果没有在 Indeed 获得的工作机会，我就不会成为一名软件工程师。我想对我的前经理 Srdjan Bodruzic 表示衷心的感谢，感谢他的慷慨和指导(以及雇用我)。感谢我的 CX 队友——Tommy Winschel、Danny Moncada、JP Schultz 和 Travis Wright——感谢他们的智慧和幽默。感谢在我任职期间提供帮助的其他 Indeed 同事：Matthew Morin、Chris Hatton、Chip Borsi、Nicole Saglimbene、Danielle Scoli、Blairr Swayne 和 George Improglou。感谢与我在 Sophie's Cuban Cuisine 共进晚餐的所有人！

我在 Stride Consulting 担任软件工程师时开始写这本书，我要感谢许多同事的支持与帮助：David "The Dominator" DiPanfilo、Min Kwak、Ben Blair、Kirsten Nordine、Michael "Bobby" Nunez、Jay Lee、James Yoo、Ray Veliz、Nathan Riemer、Julia Berchem、 Dan Plain、Nick Char、Grant Ziolkowski、Melissa Wahnish、Dave Anderson、Chris Aporta、Michael Carlson、John Galioto、Sean Marzug-McCarthy、Travis Vander Hoop、Steve Solomon 和 Jan Mlčoch。

感谢那些当我作为软件工程师和顾问时，与我共事的好朋友们：Francis Hwang、Inhak Kim、Liana Lim、Matt Bambach、Brenton Morris、Ian McNally、Josh Philips、Artem Kochnev、Andrew Kang、Andrew Fader、Karl Smith、Bradley Whitwell、Brad Popiolek、Eddie Wharton、Jen Kwok，以及我最喜欢的咖啡师 Adam McAmis 和 Andy Fritz。

感谢以下朋友为我的生活增添了光彩：Nick Bianco、Cam Stier、Keith David、Michael Cheung、Thomas Philippeau、Nicole DiAndrea 和 James Rokeach。

感谢我最喜欢的乐队 New Found Glory，为许多写作活动提供了配乐。流行朋克还没死！

感谢指导项目并协助营销工作的 Manning 工作人员：Jennifer Houle、Aleksandar Dragosavljević、

Radmila Ercegovac、Candace Gillhoolley、Stjepan Jureković 和 Lucas Weber。还要感谢曼宁负责监督内容的工作人员：Sarah Miller，我的开发编辑；Deirdre Hiam，我的产品经理；Keir Simpso，我的文字编辑；Jason Everett，我的校对。

感谢帮助我解决问题的技术审阅者：Al Pezewski、Alberto Ciarlanti、Ben McNamara、Björn Neuhaus、Christopher Kottmyer、Dan Sheikh、Dragos Manailoiu、Erico Lendzian、Jeff Smith、Jérôme Bâton、Joaquin Beltran、Jonathan Sharley、Jose Apablaza、Ken W. Alger、Martin Czygan、Mathijs Affourtit、Matthias Busch、Mike Cuddy、Monica E. Guimaraes、Ninoslav Cerkez、Rick Prins、Syed Hasany、Viton Vitanis 和 Vybhavreddy Kammireddy Changalreddy。感谢大家的努力，帮助我成为更优秀的作家和教育者。

最后，感谢 Hoboken，这是我过去 6 年生活过的小镇。我在它的公共图书馆、咖啡馆和茶馆里写下了本书的许多章节。在这个小镇，我取得了许多进步，它将永远铭刻在我的记忆中。谢谢你，Hoboken！

关于本书封面

本书封面上的人物插图标题为 *Dame de Calais*，即《来自加来的女士》。这幅插图摘自 Jacques Grasset de Saint-Sauveur(1757—1810)于 1788 年在法国出版的《法国服饰》(*Costume Civil Actuels de Tous Les Peuples Connus*)。《法国服饰》中的每幅插图都是经手工精细绘制和着色的。Grasset de Saint Sauveur 的藏品种类繁多，生动地提醒我们，在 200 年前，世界各地的城镇和地区在文化上的差异如此巨大。人们彼此隔绝，说着不同的方言和语言。无论是在街上还是在乡村，只要看一看他们的衣着，就很容易知道他们住在哪里，从事什么行业，在社会中处于什么地位。

从那时起，人们的穿着方式开始发生改变。不同地域的服饰多样性，在当时是如此丰富，但现在已经逐渐消失了。现在已很难区分不同地域的居民，更不用说区分不同的城镇、地区或国家了。也许我们用文化的多样性换取了更多样化的个人生活——当然也换来了更多样化和快节奏的科技生活。

在这个计算机书籍同质化严重的时代，曼宁出版社用表现两个世纪前人们服饰丰富多样性的图片作为书籍封面，来反映计算机行业的创造性和主动性，利用 Jacques Grasset de Saint-Sauveur 的图片提醒我们生活中存在的多样性。

序

说实话，我发现 Pandas 全靠运气。

2015 年，我在全球最大的招聘网站 Indeed.com 面试了数据运营分析师的职位。面试时，我的最后一项技术挑战是使用 Microsoft Excel 电子表格软件从内部数据集中获得信息。为了给人留下深刻印象，我使用了尽可能多的技巧：列排序、文本操作、数据透视表，当然还有标志性的 VLOOKUP 函数。(好吧，也许“标志性”这个词有点夸张。)

听起来很奇怪，当时我并没有意识到除了 Excel 外还有其他数据分析工具。Excel 无处不在：我的父母使用它，我的老师使用它，我的同事使用它。感觉它就像一个既定的标准。因此，当我收到工作邀请时，我立即购买了约 100 美元的 Excel 书籍并开始学习。是时候成为电子表格专家了！

第一天上班时，我将 50 个最常用的 Excel 函数打印出来放在手边。我刚打开电脑，经理就把我拉进了一间会议室，并告诉我事情的优先级已经发生了变化：团队的数据集已经膨胀到 Excel 无法再支持，我的同事也在寻找方法来自动执行他们每日和每周报告中的冗余步骤。幸运的是，我的经理已经找到了解决这两个问题的方法。他问我是否听说过 Pandas。

“是那个毛茸茸的家伙吗？”我疑惑地问道。

“不是，”他说，“Pandas 是 Python 数据分析库。”

在我做了所有准备之后，又要从头开始学习一项新技术了，我有点紧张。我以前从未写过任何代码。我是一个 Excel 工作者，我有能力做到这一点吗？我开始深入研究 Pandas 的官方文档、YouTube 视频、书籍、研讨会、Stack Overflow 上的问题，以及我可以找到的任何数据集。我发现开始使用 Pandas 是多么容易和快乐，代码非常直观而且直接。该软件库运行很快，并且功能非常丰富。使用 Pandas，可以用很少的代码完成大量的数据操作。

像我这样的故事在 Python 社区中很常见。在过去 10 年中，Python 的使用者数量呈天文数字式增长，其原因在于开发人员可以轻松掌握它。我相信，如果你和我的处境相似，你也可以学习 Pandas。如果你希望将数据分析技能扩展到 Excel 电子表格之外，本书将是你的绝佳选择。

当我对 Pandas 感到满意后，我继续探索 Python，然后是其他编程语言。从许多方面来讲，Pandas 引领我转向全职软件工程师。我对这个强大的软件库感激不尽，很高兴能把知识的火炬传递给大家。希望你能发现代码的魔力。

前　言

本书读者对象

《Pandas 数据分析实战》全面介绍了用于数据分析的 Pandas 库。Pandas 可以帮助你轻松地执行多种数据操作：排序、连接、旋转、清理、删除重复数据、聚合等。本书循序渐进地介绍了 Pandas 的各种功能，每种功能从较小的构建块开始，再到较大的数据结构。

《Pandas 数据分析实战》适合具有电子表格软件(如 Microsoft Excel、Google Sheets 和 Apple Numbers)以及类似的数据分析工具(如 R 和 SAS)使用经验的中级数据分析师。对于想了解更多数据分析知识的 Python 开发人员来说，也是一本非常合适的参考书。

本书的内容结构

《Pandas 数据分析实战》由 14 章组成，分为两部分。

第Ⅰ部分，Pandas 核心基础，循序渐进地介绍了 Pandas 库的基本原理。

第 1 章使用 Pandas 分析了一个示例数据集，以全面概述 Pandas 的功能。

第 2 章介绍了 Series 对象，这是一种 Pandas 的核心数据结构，用于存储有序数据的集合。

第 3 章深入地探讨 Series 对象，探索了各种 Series 操作，包括值排序、删除重复项、提取最小值和最大值等。

第 4 章介绍了二维数据表 DataFrame。本章将前几章的概念应用到新的数据结构中，并引入了额外的操作。

第 5 章展示了如何使用各种逻辑条件从 DataFrame 中过滤行的子集：相等、不等、比较、包含、排除等。

第Ⅱ部分，应用 Pandas，重点介绍更高级的 Pandas 功能，以及如何利用这些功能解决现实世界数据集的问题。

第 6 章介绍了如何在 Pandas 中处理不完美的文本数据，讨论如何解决删除空格、查找和替换字符、字母大小写，以及从单个列中提取多个值等问题。

第 7 章讨论 MultiIndex，它允许将多个列值组合成一行数据的单个标识符。

第 8 章描述了如何在数据透视表中聚合数据，将标题从行轴移到列轴，并将数据由宽格式转换为窄格式。

第 9 章探讨如何将行分组到桶中，并通过 GroupBy 对象对结果集合进行聚合。

第 10 章介绍使用各种连接将多个数据集合并为一个。

第 11 章演示了如何在 Pandas 中处理日期和时间。本章涵盖了排序日期、计算持续时间，以及确定日期是在一个月还是一个季度的开始等主题。

第 12 章展示了如何将其他文件类型导入 Pandas，包括 Excel 和 JSON，还讲解了如何从 Pandas 导出数据。

第 13 章侧重于配置库的设置。本章深入研究了如何修改显示的行数、更改浮点数的精度、将值舍入低于阈值等。

第 14 章探讨了如何使用 Matplotlib 库进行数据可视化，以及如何使用 Pandas 数据创建折线图、条形图、饼图等。

每章都建立在前一章的基础上。对于 Pandas 新手，我建议按照线性顺序阅读每个章节。同时，为了确保本书能够成为一本参考指南，我将每章都写成一个独立的教程，并带有自己的数据集。在每章的开头，都会从头开始编写代码，因此你也可以从自己喜欢的任何章节开始阅读本书。

大多数章节都以代码挑战结束，让你可以将概念应用于实践。我强烈建议你尝试一下这些代码挑战。

Pandas 建立在 Python 编程语言的基础上，建议你在学习本书之前了解 Python 语言的基本知识。对于在 Python 方面经验有限的人，附录 B 提供了对该语言的详尽介绍。

关于代码

本书包含了很多源代码的例子。它们都是用等宽字体来格式化的，以区别于普通的文本。

本书示例的源代码可在 GitHub 存储库 https://github.com/paskhaver/pandas-in-action 中找到。不熟悉 Git 和 GitHub 的人，请在存储库页面上查找 Download Zip 按钮。有 Git 和 GitHub 经验的人可以从命令行来复制。另外，扫描本书封底的二维码也可下载本书示例的源代码。

存储库还包括文本形式的完整数据集。我学习 Pandas 时，最大的挫折之一就是使用的教程喜欢依赖随机生成的数据，没有一致性，没有背景，没有故事，没有乐趣。在本书中，我们将使用许多现实中的真实数据集，涵盖从篮球运动员的薪水到神奇宝贝的类型，再到餐厅健康检查的内容。数据无处不在，Pandas 是当今分析数据的最佳工具之一。我希望你喜欢数据集并时刻保持关注。

liveBook 论坛

购买《Pandas 数据分析实战》可以免费访问由 Manning Publications 运营的私人网络论坛，可以在该论坛上对本书发表评论、提出技术问题，以及从作者和其他用户那里获得帮助。论坛地址为 https://livebook.manning.com/#!/book/pandas-in-action/discussion。你还可以在 https://livebook.manning.com/#!/discussion 上了解有关 Manning 论坛和行为规则的更多信息。

Manning 对读者的承诺是提供一个场所，让读者之间，以及读者与作者之间能够进行有意义的对话。这不是作者对任何具体参与形式的次数的承诺，作者对论坛的贡献仍然是自愿的(和无偿的)。建议读者提出一些有挑战性的问题，以激发作者的兴趣！只要这本书还在印刷，就可以从出版社的网站上访问论坛和以前讨论的归档信息。

其他在线资源

- 官方 Pandas 文档可在 https://pandas.pydata.org/docs 获得。
- 在业余时间，我在 Udemy 上发布了技术视频课程。读者可以在 https://www.udemy.com/user/borispaskhaver 上找到这些课程，其中包括 20 小时的 Pandas 课程和 60 小时的 Python 课程。
- 可随时通过 Twitter(https://twitter.com/borispaskhaver)或 LinkedIn(https://www.linkedin.com/in/boris-paskhaver)与我联系。

目　录

第 I 部分

Pandas 核心基础

欢迎阅读本书！在这部分，我们将熟悉 Pandas 的核心机制及两个主要数据结构：一维 Series 和二维 DataFrame。第 1 章首先使用 Pandas 分析了数据集，读者可以立即了解 Pandas 的功能。第 2 章和第 3 章将深入介绍 Series，包括如何从头开始创建 Series，如何从外部数据集导入它，并对其应用大量数学、统计和逻辑运算。第 4 章介绍表格 DataFrame，以及从其数据中提取行、列和值的各种方法。第 5 章侧重于通过应用逻辑标准提取 DataFrame 行的子集。在此过程中，我们将研究 8 个数据集，涵盖从票房收入到 NBA 球员，再到神奇宝贝的所有内容。

本部分涵盖了 Pandas 的基本内容，以及有效使用 Pandas 需要了解的基础知识。我已尽一切努力从头开始，从尽可能小的构建块开始，然后扩展到更大和更复杂的元素。本部分的 5 章内容将为你掌握 Pandas 奠定基础。祝你好运！

第1章

Pandas 概述

本章主要内容

- 21 世纪数据科学的发展
- 用于数据分析的 Pandas 库的发展历史
- Pandas 及其竞争对手的利弊
- Excel 中的数据分析与使用编程语言的数据分析
- 通过一个例子了解 Pandas 的功能

欢迎来到《Pandas 数据分析实战》！Pandas 是一个建立在 Python 编程语言基础上的数据分析库。库(也称为包)是解决特定领域问题的代码集合。Pandas 是一个用于数据操作的工具箱，它支持对数据进行排序、过滤、清理、重复数据删除、聚合、旋转等操作。作为 Python 庞大的数据科学生态系统的中心，Pandas 与其他用于统计、自然语言处理、机器学习、数据可视化等的软件库很好地结合在了一起。

本章将探讨现代数据分析工具的历史和发展，介绍 Pandas 如何从一个金融分析师的小型项目发展到 Stripe、谷歌和摩根大通等公司使用的行业标准，并将 Pandas 与它的竞争对手(包括 Excel 和 R)进行比较，讨论使用编程语言和使用图形电子表格应用程序之间的区别。最后，本章将使用 Pandas 来分析一个真实世界的数据集。本章可以作为全书所使用技术的概览。现在就让我们一探究竟吧！

1.1 21 世纪的数据

在阿瑟·柯南·道尔的经典短篇小说《波西米亚丑闻》中，福尔摩斯给他的助手约翰·华生的建议是：“在没有数据之前就得出结论是一个重大错误。”“不知不觉中，人们开始扭曲事实以适应理论，而不是让理论来适应事实。”

道尔的作品出版一个多世纪后，这位睿智侦探的话听起来仍然很有道理。如今，数据的使用在人们生活的方方面面都变得越来越普遍。2017 年《经济学人》的一篇评论文章中宣称，“世界上最有价值的资源不再是石油，而是数据”。数据就是证据，而证据对于企业、政府、机构和个人来说，

在这个相互关联的世界中变得越来越重要。从 Facebook 到亚马逊(Amazon)，再到 Netflix，世界上最成功的公司都将数据列为其投资组合中最宝贵的资产。联合国秘书长安东尼奥 • 古特雷斯称，准确的数据是“良好政策和决策的命脉”。从电影推荐到医疗，从供应链物流到减贫计划，数据为这一切提供了支持。在 21 世纪，社区、公司甚至国家的成功将取决于自身获取、汇总和分析数据的能力。

1.2　Pandas 介绍

数据处理工具的技术生态系统在过去十年中发展迅猛。今天，开源的 Pandas 库是数据分析和数据操作最流行的解决方案之一。开源，意味着库的源代码可以被公开下载、使用、修改和分发。Pandas 的许可证授予用户比专有软件(如 Excel)更多的权限。Pandas 库是免费使用的，一个由软件开发志愿者组成的全球团队维护着这个库，可以在 GitHub(https://github.com/pandas-dev/pandas) 上找到它的完整源代码。

Pandas 可以与微软的 Excel 电子表格软件和谷歌的基于浏览器的 Google Sheets 相媲美。在这三种技术中，用户都与由数据的行和列组成的表进行交互。一行表示一条记录，或者一列表示一个集合。应用转换将数据调整到所需的状态。

图 1-1 显示了一个表格数据集的转换示例。分析员对左边的四行数据集应用一个操作，从而得到右边的两行数据集。例如，可以选择符合条件的行，或者从原始数据集中删除重复的行。

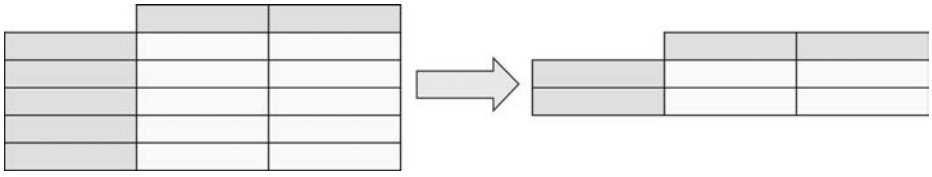

图 1-1　表格数据集的转换示例

Pandas 的独特之处在于它在处理能力和用户生产力之间的平衡。Pandas 通过低级语言(如 C)进行大量计算，进而在毫秒内高效地转换百万行数据集。同时，它维护一组简单而直观的命令。在 Pandas 中，用几行代码就可以很容易地完成很多事情。

图 1-2 显示了一个在 Pandas 中导入和排序数据集的代码示例。现在还不用担心代码，需要说明的是，整个操作只需要两行代码。

```
In [2]: populations = pd.read_csv("populations.csv")
        populations.sort_values(by = "Population", ascending = False)
```

Out[2]:

	Country	Population
144	China	1433783686
21	India	1366417754
156	United States	329064917
76	Indonesia	270625568
147	Pakistan	216565318
79	Brazil	211049527
6	Nigeria	200963599
123	Bangladesh	163046161

图 1-2　一个在 Pandas 中导入和排序数据集的代码示例

Pandas 可以无缝地处理数字、文本、日期、时间、缺失数据等。本书将通过 30 多个数据集来探索它令人难以置信的众多功能。

2008 年，软件开发者韦斯·麦金尼开发出了第一个 Pandas 版本，他当时在纽约的 AQR 资本管理投资公司(AQR Capital Management)工作。麦金尼对 Excel 和统计编程语言 R 都不满意，于是他希望有一种工具，可以很容易地解决金融行业常见的数据问题，特别是清理和汇总数据。由于找不到理想的产品，他决定自己开发一个。当时，Python 还远没有今天这样强大，但是这种语言的魅力激发了麦金尼在它的基础上建立库。他在 Quartz(http://mng.bz/w0Na)上表示：“我喜欢 Python，因为它的表达式更简洁。”“你可以用很少的代码通过 Python 表达复杂的思想，而且非常容易阅读。”

自 2009 年 12 月向公众发布以来，Pandas 的用户数量有了持续、广泛的增长，估计用户数量为 500 万到 1000 万[1]。截至 2021 年 6 月，PyPi(集中式 Python 包在线存储库，https://pepy.tech/project/pandas)网站上，Pandas 已被下载超过 7.5 亿次。它的 GitHub 代码库获得了超过 3 万颗星(一颗星相当于社交平台上的一个“赞”)。与 Pandas 相关的问题在 Stack Overflow 上所占的比例越来越大，这表明用户兴趣不断增加。

我认为，甚至可以将 Python 用户数量的天文数字式增长归功于 Pandas。Python 语言因其在数据科学领域的流行而大受欢迎，Pandas 对这个领域的发展做出了巨大贡献。大学讲授的第一门语言中，Python 是最常见的。TIOBE 指数(搜索引擎得出的编程语言流行度排名)宣布 Python 是 2018 年用户数量增长最快的语言[2]，“如果 Python 的用户数量能保持这个增长速度，它可能在 3～4 年的时间内取代 C 和 Java，成为世界上最流行的编程语言”，TIOBE 在一份新闻稿中写道。在学习 Pandas 的同时，你也将学习 Python，这是 Pandas 的另一个好处。

1.2.1 Pandas 与图形电子表格应用程序

Pandas 与 Excel 等图形电子表格应用程序的思维方式不同。编程本质上更多的是通过语言表达而不是通过视觉形象表达，通过命令而不是点击与计算机进行交互。因为它对你想要完成的事情做的假设更少，所以编程语言往往更苛刻，需要明确地告诉它该做什么。我们需要以正确的顺序使用正确的输入发出正确的指令，否则程序将无法运行。

由于要求更加严格，Pandas 的学习曲线比 Excel 或 Sheets 更陡峭。但是，如果你在 Python 或一般编程方面的经验有限，也无须担心。当你在 Excel 中调用 SUMIF 和 VLOOKUP 等函数时，你已经在像程序员一样思考了。这些过程是相同的：确定要使用的函数，然后以正确的顺序提供正确的输入。Pandas 也需要使用相同的技能，不同之处在于以更复杂的语言与计算机进行通信。

当你熟悉它的复杂性后，Pandas 会赋予你更大的数据操作能力和灵活性。除了扩展可用程序的范围之外，Pandas 还允许你将它们自动化，可以编写一段代码并在多个文件中重复使用它——这非常适合处理那些烦人的每日和每周报告。请务必注意，Excel 与 VBA(Visual Basic for Applications)捆绑在一起，VBA 是一种编程语言，可帮助你自动执行电子表格程序。然而，我认为 Python 比 VBA 更容易上手，并且具有数据分析以外的用途，可以成为你更好的选择。

---

1 参见“What’s the future of the pandas library?” Data School，https://www.dataschool.io/future-of-pandas。

2 参见 Oliver Peckham，“TIOBE Index: Python Reaches Another All-Time High”，HPC Wire，http://mng.bz/w0XP。

从 Excel 转向使用 Python 还有其他好处。Jupyter Notebook 是一种经常与 Pandas 搭配使用的编码环境，可提供更具动态性、交互性和综合性的报告。Jupyter Notebook 由多个单元格组成，每个单元格包含一段可执行代码。分析师可以将这些单元格与标题、图表、描述、注释、图像、视频、图表等集成在一起。其他人可以按照分析师的分步逻辑来了解他们是如何得出结论的，而不仅仅是最终结果。

Pandas 的另一个优势是 Python 的大数据科学生态系统，可轻松与统计、自然语言处理、机器学习、网页抓取、数据可视化等软件库集成。Python 每年都会增加大量新的软件库，这些强大的工具有时是 Pandas 的企业级竞争对手所不具备的，因为它们缺乏大型全球贡献者社区的支持。

随着数据集的增长，图形电子表格应用程序也开始陷入困境。在这方面，Pandas 明显比 Excel 强大。软件库能够处理的数据量仅受计算机内存和处理能力的限制。在大多数当代计算机上，Pandas 可以很好地处理具有数百万行的超大数据集，尤其是当开发人员知道如何对性能进行优化时。Pandas 的创建者麦金尼在描述 Pandas 局限性的博客文章中写道：“现在，根据我使用 Pandas 的经验而总结的规则是，你的内存应该是数据集大小的 5～10 倍。”(http://mng.bz/qeK6)

当你为工作选择最佳工具时，你的组织和项目如何定义数据分析和大数据等术语是必须考虑的因素。Excel 被全球约 7.5 亿在职专业人士使用，但是电子表格的最大行数限制为 1 048 576 行[1]。对于某些分析师而言，100 万行数据可能是一个天文数字，但对另外一些分析师而言，100 万行数据可能只是冰山一角。

建议你不要将 Pandas 视为最好的数据分析解决方案，而是将其视为与其他现代技术一起使用的强大技术。Excel 仍然是快速、简单的数据操作的绝佳选择。电子表格应用程序通常会对你的意图做出假设，这就是为什么只需要单击几下即可导入 CSV 文件或对包含 100 个值的列进行排序的原因，将 Pandas 用于此类简单任务并没有真正的优势(尽管它能够胜任这些任务)。但是，当你需要清理两个各有 1000 万行的数据集内的文本值、删除重复记录、连接它们，并对 100 批文件重复该逻辑时，你会使用哪种解决方案？对于这些场景，使用 Python 和 Pandas 更容易、更省时。

1.2.2　Pandas 与它的竞争对手

数据科学爱好者经常将 Pandas 与开源编程语言 R，以及专有软件套件 SAS 进行比较，每个解决方案都有自己的拥护者社区。

R 是一种具有统计学基础的专业语言，而 Python 是一种用于多个技术领域的通用语言。可以预见，这两种语言往往会吸引具有特定领域专业知识的用户。哈德利 • 威克姆是 R 社区的一位杰出开发人员，他构建了一个名为 tidyverse 的数据科学包集合，并建议用户将这两种语言视为合作者而不是竞争对手。“这些编程语言独立存在，而且以不同的方式为大家提供各种叹为观止的功能，”他在 Quartz(http://mng.bz/Jv9V)中说道，“我看到的一个模式是，一家公司的数据科学团队使用 R，而数据工程团队使用 Python。Python 的用户往往具有软件工程背景，并且对自己的编程技术非常有信心……R 用户非常喜欢 R，但无法与数据工程团队争论，因为他们在编程方面没有共同语言。”

1 参见 Andy Patrizio，“Excel: Your entry into the world of data analytics”，Computer World，http://mng.bz/qe6r。

一种语言可能具有另一种语言没有的高级功能，但在数据分析的常见任务中，这两种语言几乎达到了同等水平。开发人员和数据科学家往往只会被他们最了解的东西所吸引。

SAS 是一套支持统计、数据挖掘、计量经济学等功能的软件工具，由位于北卡罗来纳州的 SAS 研究所开发，一般按年收取软件使用许可费用。带有技术支持的商业化产品的优势包括跨工具的技术、视觉一致性、强大的文档，以及面向企业客户需求的产品路线图。但像 Pandas 这样的开源技术，享有更自由的开发方法，开发人员为自己的需求和其他开发人员的需求而工作，这有时会与商业软件市场趋势不符。

某些技术与 Pandas 具有相同的功能，但被用于不同的应用场景，SQL 就是一个例子。SQL(结构化查询语言)是一种用于与关系数据库通信的语言。关系数据库由通过公共键链接的数据表组成。我们可以使用 SQL 进行基本的数据操作，例如从表中提取列和按条件过滤行，但它的功能范围更广，根本上是围绕数据管理。建立数据库是为了存储数据，数据分析是次要的应用场景。SQL 可以创建新表、使用新值更新现有记录、删除现有记录等，相比之下，Pandas 完全是为数据分析而构建的，其功能包括统计计算、数据整理、数据合并等。在典型的工作环境中，这两种工具通常是互补的。分析师可能会使用 SQL 来提取初始数据集，然后使用 Pandas 对其进行操作。

总而言之，Pandas 不是唯一的工具，但它是解决大多数数据分析问题的强大、流行且有价值的解决方案，并且 Python 在其对简洁性和生产力的关注方面确实大放异彩。正如其创建者吉多·范罗苏姆所说：“Python 编码的乐趣应该在于简洁、可读性好的数据结构，这些数据结构用少量清晰的代码表达了很多操作”(http://mng.bz/7jo7)。Pandas 符合这一标准，对于渴望通过强大的现代数据分析工具包提高编程技能的电子表格分析师来说，这是一个极好的转型方向。

1.3 Pandas 之旅

掌握 Pandas 的最好方法是亲自了解它如何运作。下面以分析有史以来票房最高的 700 部电影的数据集为例介绍 Pandas 如何工作。希望读者对 Pandas 的语法的直观程度感到惊喜，即使你是编程新手。

阅读本节内容时，尽量不要过度分析代码示例，甚至不需要复制它们。本节主要对 Pandas 的特征和功能进行总体介绍，也会详细地讨论 Pandas 的具体功能与特性。

全书使用 Jupyter Notebook 开发环境来编写代码。如果你想在计算机上设置 Pandas 和 Jupyter Notebook，请参阅附录 A，可以登录 https://www.github.com/paskhaver/pandas-in-action 下载所有数据集和完整的 Jupyter Notebook。

1.3.1 导入数据集

首先在 movies.csv 文件所在的目录中创建一个新的 Jupyter Notebook，然后导入 Pandas 库以访问其功能：

```
In [1] import pandas as pd
```

代码左侧的框(在上面的示例中显示数字 1)标记单元格相对于 Jupyter Notebook 启动或重新启动的执行顺序，可以按任意顺序执行单元格，并且可以多次执行同一个单元格。

通读本书时，可以通过在 Jupyter 单元格中执行不同的代码片段来进行实验。因此，如果执行编号与文本中的编号不匹配也没关系。

数据存储在一个单一的 movies.csv 文件中。CSV(逗号分隔值)文件是一个纯文本文件，它用换行符分隔每一行数据，用逗号分隔每一列值。文件的第一行包含数据的列标题。下面是 movies.csv 前三行的预览：

```
Rank,Title,Studio,Gross,Year
1,Avengers: Endgame,Buena Vista,"$2,796.30",2019
2,Avatar,Fox,"$2,789.70",2009
```

第一行列出了数据集中的 5 个列标题：Rank、Title、Studio、Gross 和 Year。第二行为记录集中的第一条记录，或者理解为第一部电影的数据。排名第 1 的电影名字为“Avengers：Endgame”，电影公司为“Buena Vista”，总收入为“$2796.30”，并在 2019 年发行。下一行包含下一部电影的数据，数据集里面这样的数据有 750 多行。

Pandas 可以导入各种类型的文件，每种文件类型在库的顶层都有一个关联的导入函数。Pandas 中的函数相当于 Excel 中的函数，是向软件库或其中的实体发出的命令。在这种情况下，我们将使用 read_csv 函数导入 movies.csv 文件：

```
In  [2] pd.read_csv("movies.csv")
```

Out [2]

	Rank	Title	Studio	Gross	Year
0	1	Avengers: Endgame	Buena Vista	$2,796.30	2019
1	2	Avatar	Fox	$2,789.70	2009
2	3	Titanic	Paramount	$2,187.50	1997
3	4	Star Wars: The Force Awakens	Buena Vista	$2,068.20	2015
4	5	Avengers: Infinity War	Buena Vista	$2,048.40	2018
…	…	…	…	…	…
777	778	Yogi Bear	Warner Brothers	$201.60	2010
778	779	Garfield: The Movie	Fox	$200.80	2004
779	780	Cats & Dogs	Warner Brothers	$200.70	2001
780	781	The Hunt for Red October	Paramount	$200.50	1990
781	782	Valkyrie	MGM	$200.30	2008

782 rows × 5 columns

Pandas 将 CSV 文件的内容导入一个名为 DataFrame 的对象中，你可以将这个对象视为存储数据的容器。不同的对象针对不同类型的数据进行了优化，我们以不同的方式与它们进行交互。Pandas 使用 DataFrame 对象存储多列数据集，使用 Series 对象存储单列数据集。DataFrame 相当于 Excel 中的多列表格。

为避免显示结果过多导致页面过长，Pandas 仅显示 DataFrame 的前五行和后五行。省略了的数据使用省略号(…)代替。

DataFrame 由 5 列(Rank、Title、Studio、Gross、Year)和一个索引组成。索引是 DataFrame 左侧的升序数字，用作数据行的标识符，我们可以将任何列设置为 DataFrame 的索引。如果没有明确说明 Pandas 使用哪一列数据，Pandas 会生成一个从 0 开始的数字索引。

哪一列是索引的最佳选择？该列的值可以作为每一行的主要标识符或参考点。在文件的 5 列

中，Rank 和 Title 是两个最佳选择。下面将自动生成的数字索引与 Title 列中的值交换，可以在 CSV 导入期间直接执行此操作。

```
In  [3] pd.read_csv("movies.csv", index_col = "Title")

Out [3]
```

Title	Rank	Studio	Gross	Year
Avengers: Endgame	1	Buena Vista	$2,796.30	2019
Avatar	2	Fox	$2,789.70	2009
Titanic	3	Paramount	$2,187.50	1997
Star Wars: The Force Awakens	4	Buena Vista	$2,068.20	2015
Avengers: Infinity War	5	Buena Vista	$2,048.40	2018
…	…	…	…	…
Yogi Bear	778	Warner Brothers	$201.60	2010
Garfield: The Movie	779	Fox	$200.80	2004
Cats & Dogs	780	Warner Brothers	$200.70	2001
The Hunt for Red October	781	Paramount	$200.50	1990
Valkyrie	782	MGM	$200.30	2008

782 rows × 4 columns

接下来，将 DataFrame 分配给一个 movies 变量，以便在程序的其他地方引用它。变量是用户为程序中的对象分配的名称：

```
In [4] movies = pd.read_csv("movies.csv", index_col = "Title")
```

有关变量的更多信息，请查看附录 B。

1.3.2 操作 DataFrame

我们可以从多个角度来查看 DataFrame，比如可以从开头提取几行：

```
In  [5] movies.head(4)

Out [5]
```

Title	Rank	Studio	Gross	Year
Avengers: Endgame	1	Buena Vista	$2,796.30	2019
Avatar	2	Fox	$2,789.70	2009
Titanic	3	Paramount	$2,187.50	1997
Star Wars: The Force Awakens	4	Buena Vista	$2,068.20	2015

可以查看数据集的尾部记录：

```
In  [6] movies.tail(6)

Out [6]
```

Title	Rank	Studio	Gross	Year
21 Jump Street	777	Sony	$201.60	2012
Yogi Bear	778	Warner Brothers	$201.60	2010

```
Garfield: The Movie              779                Fox        $200.80          2004
Cats & Dogs                      780    Warner Brothers        $200.70          2001
The Hunt for Red October         781          Paramount        $200.50          1990
Valkyrie                         782                MGM        $200.30          2008
```

可以通过如下命令找出 DataFrame 有多少行：

```
In  [7] len(movies)

Out [7] 782
```

可以向 Pandas 查询 DataFrame 中的行数和列数。这个数据集有 782 行和 4 列：

```
In  [8] movies.shape

Out [8] (782, 4)
```

可以查询单元格总数：

```
In  [9] movies.size

Out [9] 3128
```

可以查询 4 列的数据类型。在以下输出中，int64 表示整数列，object 表示文本列：

```
In  [10] movies.dtypes

Out [10]

Rank             int64
Studio          object
Gross           object
Year             int64
dtype: object
```

可以按行中的数字顺序从数据集中提取一行，称为其索引位置。在大多数编程语言中，索引从 0 开始计数。因此，如果想提取数据集中的第 500 部电影的信息，应将索引位置定位为 499：

```
In  [11] movies.iloc[499]

Out [11] Rank          500
         Studio        Fox
         Gross      $288.30
         Year          2018
         Name: Maze Runner: The Death Cure, dtype: object
```

Pandas 在这里返回一个叫作 Series 的新对象，这是一个一维标记的值数组，可以将其视为单列数据表，每行都有一个标识符。请注意，Series 的索引标签(Rank、Studio、Gross 和 Year)来自 movies DataFrame 的 4 列。Pandas 改变了原始行值的显示方式。

还可以使用索引标签来访问 DataFrame 行。提醒一下，DataFrame 索引保存了电影的标题。下面以《阿甘正传》为例介绍如何提取数据值。该例通过其索引标签而不是其数字位置提取一行数据：

```
In  [12] movies.loc["Forrest Gump"]

Out [12]  Rank             119
```

```
Studio      Paramount
Gross         $677.90
Year             1994
Name: Forrest Gump, dtype: object
```

索引标签可以包含重复项。例如，DataFrame 中的两部电影的标题是“101 Dalmatians”(1961 年的原版和 1996 年的翻拍版本)：

```
In  [13] movies.loc["101 Dalmatians"]
```

Out [13]

Title	Rank	Studio	Gross	Year
101 Dalmatians	425	Buena Vista	$320.70	1996
101 Dalmatians	708	Buena Vista	$215.90	1961

尽管 Pandas 允许重复，但建议尽可能保持索引标签的唯一性。唯一的标签集合可以加快 Pandas 定位和提取特定行的速度。

CSV 中的电影按 Rank 列中的值进行排序。如果想看最新上映的 5 部电影，可以根据另一列中的值对 DataFrame 进行排序，例如 Year：

```
In  [14] movies.sort_values(by = "Year", ascending = False).head()
```

Out [14]

Title	Rank	Studio	Gross	Year
Avengers: Endgame	1	Buena Vista	2796.3	2019
John Wick: Chapter 3 - Parab...	458	Lionsgate	304.7	2019
The Wandering Earth	114	China Film Corporation	699.8	2019
Toy Story 4	198	Buena Vista	519.8	2019
How to Train Your Dragon: Th...	199	Universal	519.8	2019

还可以根据多列的值对 DataFrame 进行排序。先按 Studio 列的值排序，如果出现重复值，再按 Year 列的值对电影进行排序，就可以看到按电影公司和上映日期顺序排列的电影信息：

```
In  [15] movies.sort_values(by = ["Studio", "Year"]).head()
```

Out [15]

Title	Rank	Studio	Gross	Year
The Blair Witch Project	588	Artisan	$248.60	1999
101 Dalmatians	708	Buena Vista	$215.90	1961
The Jungle Book	755	Buena Vista	$205.80	1967
Who Framed Roger Rabbit	410	Buena Vista	$329.80	1988
Dead Poets Society	636	Buena Vista	$235.90	1989

如果想按字母顺序查看电影信息，还可以对索引进行排序：

```
In  [16] movies.sort_index().head()
```

Out [16]

Title	Rank	Studio	Gross	Year
10,000 B.C.	536	Warner Brothers	$269.80	2008
101 Dalmatians	708	Buena Vista	$215.90	1961
101 Dalmatians	425	Buena Vista	$320.70	1996
2 Fast 2 Furious	632	Universal	$236.40	2003
2012	93	Sony	$769.70	2009

到目前为止，所执行的操作返回新的 DataFrame 对象。Pandas 没有对原始的 movies DataFrame 对象进行修改。这类操作的非破坏性是有益的，因为可以利用这个特性反复测试，直到取得正确的结果。

1.3.3　计算 Series 中的值

下面尝试一个更复杂的分析。如何找出哪家电影公司拥有最卖座的电影呢？为了解决这个问题，需要计算每个电影公司出现在 Studio 列中的次数。

从 DataFrame 中提取一列数据作为 Series。请注意，Pandas 在 Series 中保存了 DataFrame 的索引，即电影标题：

```
In  [17]  movies["Studio"]

Out [17] Title
         Avengers: Endgame                          Buena Vista
         Avatar                                         Fox
         Titanic                                    Paramount
         Star Wars: The Force Awakens               Buena Vista
         Avengers: Infinity War                     Buena Vista
                                                        ...
         Yogi Bear                                Warner Brothers
         Garfield: The Movie                            Fox
         Cats & Dogs                              Warner Brothers
         The Hunt for Red October                   Paramount
         Valkyrie                                       MGM
         Name: Studio, Length: 782, dtype: object
```

如果 Series 有很多行数据，Pandas 会截断数据集，只显示前五行和后五行。

分离了 Studio 列后就可以计算每个唯一值出现的次数，将结果限制在排名前 10 的电影公司：

```
In  [18] movies["Studio"].value_counts().head(10)

Out [18] Warner Brothers        132
         Buena Vista            125
         Fox                    117
         Universal              109
         Sony                    86
         Paramount               76
         Dreamworks              27
         Lionsgate               21
         New Line                16
         MGM                     11
         Name: Studio, dtype: int64
```

返回值是另一个 Series 对象。此时，Pandas 使用 Studio 列中的公司名称作为索引标签，它们的计数作为 Series 值。

1.3.4 根据一个或多个条件筛选列

人们有时需要根据一个或多个条件提取行的子集，Excel 为此提供了 Filter 工具。

如果只想找到 Universal Studios 发行的电影，就可以用 Pandas 中的一行代码来完成这个任务：

```
In  [19] movies[movies["Studio"] == "Universal"]
```

Out [19]

Title	Rank	Studio	Gross	Year
Jurassic World	6	Universal	$1,671.70	2015
Furious 7	8	Universal	$1,516.00	2015
Jurassic World: Fallen Kingdom	13	Universal	$1,309.50	2018
The Fate of the Furious	17	Universal	$1,236.00	2017
Minions	19	Universal	$1,159.40	2015
…	…	…	…	…
The Break-Up	763	Universal	$205.00	2006
Everest	766	Universal	$203.40	2015
Patch Adams	772	Universal	$202.30	1998
Kindergarten Cop	775	Universal	$202.00	1990
Straight Outta Compton	776	Universal	$201.60	2015

109 rows × 4 columns

可以将过滤条件赋给一个变量以便在后续代码中使用：

```
In  [20] released_by_universal = (movies["Studio"] == "Universal")
         movies[released_by_universal].head()
```

Out [20]

Title	Rank	Studio	Gross	Year
Jurassic World	6	Universal	$1,671.70	2015
Furious 7	8	Universal	$1,516.00	2015
Jurassic World: Fallen Kingdom	13	Universal	$1,309.50	2018
The Fate of the Furious	17	Universal	$1,236.00	2017
Minions	19	Universal	$1,159.40	2015

还可以根据多个条件过滤 DataFrame 行。以下代码可以获取 2015 年由 Universal Studios 发行的所有电影：

```
In  [21] released_by_universal = movies["Studio"] == "Universal"
         released_in_2015 = movies["Year"] == 2015
         movies[released_by_universal & released_in_2015]
```

Out [21]

Title	Rank	Studio	Gross	Year
Jurassic World	6	Universal	$1,671.70	2015
Furious 7	8	Universal	$1,516.00	2015
Minions	19	Universal	$1,159.40	2015
Fifty Shades of Grey	165	Universal	$571.00	2015
Pitch Perfect 2	504	Universal	$287.50	2015
Ted 2	702	Universal	$216.70	2015
Everest	766	Universal	$203.40	2015
Straight Outta Compton	776	Universal	$201.60	2015

前面的示例包含同时满足这两个条件的行。若要筛选符合以下两个条件中任意一个的电影：Universal Studios 发行的电影或 2015 年发行的电影，则结果的 DataFrame 更大。

```
In  [22] released_by_universal = movies["Studio"] == "Universal"
         released_in_2015 = movies["Year"] == 2015
         movies[released_by_universal | released_in_2015]

Out [22]
```

Title	Rank	Studio	Gross	Year
Star Wars: The Force Awakens	4	Buena Vista	$2,068.20	2015
Jurassic World	6	Universal	$1,671.70	2015
Furious 7	8	Universal	$1,516.00	2015
Avengers: Age of Ultron	9	Buena Vista	$1,405.40	2015
Jurassic World: Fallen Kingdom	13	Universal	$1,309.50	2018
…	…	…	…	…
The Break-Up	763	Universal	$205.00	2006
Everest	766	Universal	$203.40	2015
Patch Adams	772	Universal	$202.30	1998
Kindergarten Cop	775	Universal	$202.00	1990
Straight Outta Compton	776	Universal	$201.60	2015

140 rows × 4 columns

Pandas 提供了额外的方法来过滤 DataFrame。例如，可以将小于或大于特定值的列值作为筛选条件。下面以 1975 年之前发行的电影为筛选条件：

```
In  [23] before_1975 = movies["Year"] < 1975
         movies[before_1975]

Out [23]
```

Title	Rank	Studio	Gross	Year
The Exorcist	252	Warner Brothers	$441.30	1973
Gone with the Wind	288	MGM	$402.40	1939
Bambi	540	RKO	$267.40	1942
The Godfather	604	Paramount	$245.10	1972
101 Dalmatians	708	Buena Vista	$215.90	1961
The Jungle Book	755	Buena Vista	$205.80	1967

指定一个范围，所有值必须在该范围之内。例如，筛选 1983—1986 年发行的电影：

```
In  [24] mid_80s = movies["Year"].between(1983, 1986)
         movies[mid_80s]

Out [24]
```

Title	Rank	Studio	Gross	Year
Return of the Jedi	222	Fox	$475.10	1983
Back to the Future	311	Universal	$381.10	1985
Top Gun	357	Paramount	$356.80	1986
Indiana Jones and the Temple of Doom	403	Paramount	$333.10	1984
Crocodile Dundee	413	Paramount	$328.20	1986
Beverly Hills Cop	432	Paramount	$316.40	1984
Rocky IV	467	MGM	$300.50	1985
Rambo: First Blood Part II	469	TriStar	$300.40	1985
Ghostbusters	485	Columbia	$295.20	1984
Out of Africa	662	Universal	$227.50	1985

也使用 DataFrame 索引来过滤行。例如，先将索引的电影标题中的大写字母转换为小写字母，并查找标题中带有“dark”一词的所有电影：

```
In  [25] has_dark_in_title = movies.index.str.lower().str.contains("dark")
         movies[has_dark_in_title]

Out [25]
```

Title	Rank	Studio	Gross	Year
Transformers: Dark of the Moon	23	Paramount	$1,123.80	2011
The Dark Knight Rises	27	Warner Brothers	$1,084.90	2012
The Dark Knight	39	Warner Brothers	$1,004.90	2008
Thor: The Dark World	132	Buena Vista	$644.60	2013
Star Trek Into Darkness	232	Paramount	$467.40	2013
Fifty Shades Darker	309	Universal	$381.50	2017
Dark Shadows	600	Warner Brothers	$245.50	2012
Dark Phoenix	603	Fox	$245.10	2019

注意，无论“dark”出现在标题的哪个位置，Pandas 都会查找到。

1.3.5 对数据分组

有时人们需要筛选出总票房最高的电影公司，首先按电影公司名称汇总 Gross 列的值。

注意，Gross 列的值存储为文本而不是数字。Pandas 将列的值作为文本导入，以保留原始 CSV 中的美元符号和逗号符号。可以将列的值转换为十进制数，但前提是删除美元符号和逗号符号。例如，将出现的所有“$”和“,”替换为空文本，此操作类似于 Excel 中的查找和替换：

```
In  [26] movies["Gross"].str.replace(
             "$", "", regex = False
         ).str.replace(",", "", regex = False)

Out [26] Title
         Avengers: Endgame                  2796.30
```

```
Avatar                                2789.70
Titanic                               2187.50
Star Wars: The Force Awakens          2068.20
Avengers: Infinity War                2048.40
                                         ...
Yogi Bear                              201.60
Garfield: The Movie                    200.80
Cats & Dogs                            200.70
The Hunt for Red October               200.50
Valkyrie                               200.30
Name: Gross, Length: 782, dtype: object
```

将美元符号和逗号符号替换为空文本后，可以将 Gross 列的值由文本转换为浮点数：

```
In  [27] (
            movies["Gross"]
            .str.replace("$", "", regex = False)
            .str.replace(",", "", regex = False)
            .astype(float)
          )

Out  [27] Title
          Avengers: Endgame                   2796.3
          Avatar                              2789.7
          Titanic                             2187.5
          Star Wars: The Force Awakens        2068.2
          Avengers: Infinity War              2048.4
                                                 ...
          Yogi Bear                            201.6
          Garfield: The Movie                  200.8
          Cats & Dogs                          200.7
          The Hunt for Red October             200.5
          Valkyrie                             200.3
          Name: Gross, Length: 782, dtype: float64
```

同样，这些操作是临时的，不会修改原始 Gross 这个 Series 对象。在前面的所有示例中，Pandas 创建了原始数据结构的副本，对副本执行操作，并返回一个新对象。例如，使用新的带有小数点的数字列明确覆盖 movies 中的 Gross 列，这种转换是永久性的：

```
In  [28] movies["Gross"] = (
         movies["Gross"]
         .str.replace("$", "", regex = False)
         .str.replace(",", "", regex = False)
         .astype(float)
         )
```

数据类型转换为更多的计算和操作提供了方便，例如计算电影的平均票房收入：

```
In  [29] movies["Gross"].mean()

Out [29] 439.0308184143222
```

回到最初的问题：计算每个电影公司的总票房收入。首先确定电影公司并将每个公司的电影(或记录行)分类，这个过程称为分组。例如，根据 Studio 列的值对 DataFrame 的行进行分组：

```
In [30] studios = movies.groupby("Studio")
```

计算每个电影公司的电影数量：

```
In  [31] studios["Gross"].count().head()

Out [31] Studio
         Artisan                    1
         Buena Vista              125
         CL                         1
         China Film Corporation     1
         Columbia                   5
         Name: Gross, dtype: int64
```

将之前的结果按电影公司名称字母的顺序排序，也可以按电影出品数量由多到少的顺序对 Series 进行排序：

```
In  [32] studios["Gross"].count().sort_values(ascending = False).head()

Out [32] Studio
         Warner Brothers          132
         Buena Vista              125
         Fox                      117
         Universal                109
         Sony                      86
         Name: Gross, dtype: int64
```

接下来，添加每个电影公司的 Gross 列的值。Pandas 将识别每个电影公司的电影子集，并计算它们的总票房：

```
In  [33] studios["Gross"].sum().head()

Out [33] Studio
         Artisan                 248.6
         Buena Vista           73585.0
         CL                      228.1
         China Film Corporation  699.8
         Columbia               1276.6
         Name: Gross, dtype: float64
```

同样，也可以按电影公司名称对结果进行排序。若要找出总票房最高的电影公司，应按降序对 Series 值进行排序。以下是票房最高的 5 个电影公司：

```
In  [34] studios["Gross"].sum().sort_values(ascending = False).head()

Out [34] Studio
         Buena Vista            73585.0
         Warner Brothers        58643.8
         Fox                    50420.8
         Universal              44302.3
         Sony                   32822.5
         Name: Gross, dtype: float64
```

只需要几行代码，我们就可以从这个复杂的数据集中得出一些有趣的见解。例如，Warner Brothers 电影公司在列表中的电影比 Buena Vista 多，但 Buena Vista 的所有电影的总票房更高。这一事实表明，Buena Vista 出品的电影平均票房高于 Warner Brothers。

本章只对 Pandas 做了初步介绍，实际上，Pandas 可以对数据做各种处理。本书的后续章节将更详细地讨论本章使用的所有代码，下一章将深入研究 Pandas 的核心构建块：Series 对象。

1.4　本章小结

- Pandas 是一个基于 Python 编程语言的数据分析库。
- Pandas 擅长用简洁的语法对大型数据集执行复杂的操作。
- Pandas 的竞争对手包括电子表格应用程序 Excel、统计编程语言 R 和 SAS 软件套件。
- 编程需要的技能与使用 Excel 或表格需要的技能不同。
- Pandas 可以导入多种文件格式，包括 CSV。CSV 是一种流行的文件格式，它用换行符分隔行，用逗号分隔列值。
- DataFrame 是 Pandas 的主要数据结构，它实际上是一个包含多列的数据表。
- Series 是一个一维标记数组，可以把它想象成只有一列的数据表。
- 可以通过行号或索引标签访问 Series 或 DataFrame 中的数据行。
- 可以通过一列或多列的值对 DataFrame 进行排序。
- 可以使用逻辑条件从 DataFrame 中提取数据子集。
- 可以根据列的值对 DataFrame 行进行分组，还可以对分组结果执行聚合操作，例如求和。