HZ BOOKS
华章科技

Python for Data Analysis

利用Python进行数据分析

O'REILLY®

机械工业出版社
China Machine Press

Wes McKinney 著
唐学韬 等译

利用Python进行数据分析

Wes McKinney 著

唐学韬 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权机械工业出版社出版

机械工业出版社

图书在版编目（CIP）数据

利用Python进行数据分析 /（美）麦金尼（McKinney, W.）著；唐学韬等译.
—北京：机械工业出版社，2013.9
（O'Reilly精品图书系列）

书名原文：Python for Data Analysis
ISBN 978-7-111-43673-7

I. 利… II. ①麦… ②唐… III. ①统计分析－应用软件 IV. C819
中国版本图书馆CIP数据核字（2013）第187885号

北京市版权局著作权合同登记
图字：01-2013-4246号

书　　名/　利用Python进行数据分析
书　　号/　ISBN 978-7-111-43673-7
责任编辑/　秦健
封面设计/　Karen Montgomery，张健
出版发行/　机械工业出版社
地　　址/　北京市西城区百万庄大街22号（邮政编码 100037）
印　　刷/　藁城市京瑞印刷有限公司
开　　本/　178毫米×233毫米　16开本　29印张
版　　次/　2014年1月第1版　2014年1月第1次印刷
定　　价/　89.00元（册）

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)88378991；88361066
购书热线：(010)68326294；88379649；68995259
投稿热线：(010)88379604
读者信箱：hzjsj@hzbook.com

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。无论是通过书籍出版，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

译者序

说句真心话，我非常感谢有机会翻译这本书，所以这可算是第一篇我自己真正想写的译者序。虽然之前也翻译过好几本书，但都没有这次的感悟这么多、这么深！这本书是我花精力和时间最多，同时也是最不满意的一本，就是因为这些感悟 —— 我始终觉得，如果再多点时间的话，我还可以翻译得更好。

本书的内容非常好，至少有一点非常好 —— 集中火力对付特定的应用领域。市面上介绍编程的书多如牛毛，但几乎没有几本书是针对特定应用场景的。这本书对新手来说绝对是福音，因为每看完一点就可以马上将自己手上的工作直接拿来当例子练手，这种立竿见影的学习效果，绝对会增强新手的学习信心。

本书内容虽好，但由于作者是编辑界牛人，平时的工作肯定不少，写书方面的精力自然就不可能太多。加之美式英语本来就很口语化，导致原书口水话非常多，有些地方的从句跟绕口令似的。我在翻译的过程中尽量排除了一些，两次校稿的过程中又删除或大幅修改了一些废话，虽然这种“口水话”还存在不少，但至少不会对阅读造成太大影响。如果实在觉得语言不通顺，请随时发邮件给我，欢迎大家的善意指导（*tonytang1999@126.com*）。

此外，在翻译的过程中发现了不少小问题，用词方面的错误几乎都是直接改的（小部分写了译者注，因为编辑要求我尽量标出一些来以便核对），而其他错误则几乎全部采用译者注的形式说明，还有一些原文有歧义或不详尽的地方也通过译者注的形式给出了简单说明。

本书共12章，除非你已经什么都会了，否则我建议全部阅读。如果没有学过Python，建议先看看本书后面的附录。本书所用到的Python编程基础知识很少，所以只看那个附录完全

足够了。但是，如果你一点儿编程基础都没有的话，可能需要再看一本有关Python入门的书才行（比如《Python编程实践》[编注1]）。

对了，还有几件事情需要说明一下：

- 每章的代码示例最好在一个IPython会话中完成，否则可能会出现一些不必要的麻烦，比如“xxx未定义”。
- 如果在Windows里面用IPython，复制代码的时候建议使用cpaste，这个不多解释了。
- 有关地图的那段代码可能需要找英文资料看才行，我在译者注中也说明了。这可能需要花不少时间和精力。
- 由于原文各种说法不统一（甚至包括术语），虽然我尽量做了统一处理，但由于精力和时间有限，无法完全修改，所以译文中的“xxx接受yyy”、“将yyy传入xxx”说的都是“xxx函数有yyy这么个参数”；“选项”、“位置参数”、“关键字参数”、“形参”、“实参”说的都是“参数”……还有不少，我也记不清了。
- “金融和经济数据”那一章翻译得非常痛苦，因为我根本不了解那个行业，原文的术语又不标准，于是我基本都是用wikipedia和bing查英文资料，看懂之后再到baidu找中文资料，并最终确定译文。因此，可能会有不准确的情况，如果您发现了，请及时通过邮件告诉我，万分感谢。

此外，我必须感谢华章公司的编辑们。非常感谢他们能够给我这样的机会，也非常感谢他们在整个过程中给予我的各种支持和理解。希望以后还能有更加愉快的合作。

本书大部分内容的翻译工作以及全书的统稿工作由我完成，参与本书翻译校对工作的还有黄惠庄、卢彦良、蒲巧惠、陈丽丽、胡元江、张杨、赵杰、吴斌、郭敏、林丹、王跃等。

由于译者水平有限，书中肯定会存在一些错误或不妥之处，因此，在阅读过程中发现有任何问题，请随时联系我们（*tonytang1999@126.com*）或机械工业出版社，我们将及时更新本书的勘误表。当然，也非常欢迎大家对本书提出宝贵的意见和建议。

唐学韬

2013年6月于广州

编注1：本书已由机械工业出版社出版，ISBN:978-7-111-36478-8。

目录

前言

针对科学计算领域的Python开源库生态系统在过去10年中得到了飞速发展。2011年底，我深深地感觉到，由于缺乏集中的学习资源，刚刚接触数据分析和统计应用的Python程序员举步维艰。针对数据分析的关键项目（尤其是NumPy、matplotlib和pandas）已经很成熟了，也就是说，写一本专门介绍它们的图书貌似不会很快过时。因此，我下定决心要开始这样的一个写作项目。我在2007年刚开始用Python进行数据分析工作时就希望能够得到这样一本书。希望你也能觉得本书有用，同时也希望你能将书中介绍的那些工具高效地运用到实际工作中去。

本书的约定

本书使用了以下排版约定：

斜体（Italic）
: 用于新术语、URL、电子邮件地址、文件名与文件扩展名。

等宽字体（`Constant width`）
: 用于表明程序清单，以及在段落中引用的程序中的元素，如变量、函数名、数据库、数据类型、环境变量、语句、关键字等。

等宽粗体（**`Constant width bold`**）
: 用于表明命令，或者需要读者逐字输入的文本内容。

等宽斜体（*`Constant width italic`*）
: 用于表示需要使用用户提供的值或者由上下文决定的值来替代的文本内容。

注意：代表一个技巧、建议或一般性说明。

警告：代表一个警告或注意事项。

示例代码的使用

本书提供代码的目的是帮你快速完成工作。一般情况下，你可以在你的程序或文档中使用本书中的代码，而不必取得我们的许可，除非你想复制书中很大一部分代码。例如，你在编写程序时，用到了本书中的几个代码片段，这不必取得我们的许可。但若将O'Reilly图书中的代码制作成光盘并进行出售或传播，则需获得我们的许可。引用示例代码或书中内容来解答问题无需许可。将书中很大一部分的示例代码用于你个人的产品文档，这需要我们的许可。

如果你引用了本书的内容并标明版权归属声明，我们对此表示感谢，但这不是必需的。版权归属声明通常包括：标题、作者、出版社和ISBN号，例如："*Python for Data Analysis* by William Wesley McKinney (O'Reilly). Copyright 2013 William Wesley McKinney, 978-1-449-31979-3"。

如果你认为你对示例代码的使用已经超出上述范围，或者你对是否需要获得示例代码的授权还不清楚，请随时联系我们：*permissions@oreilly.com*。

联系我们

有关本书的任何建议和疑问，可以通过下列方式与我们取得联系：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

我们会在本书的网页中列出勘误表、示例和其他信息。可以通过*http://oreil.ly/Python_for_Data_Analysis*访问该页面。

要评论或询问本书的技术问题，请发送电子邮件到：

bookquestions@oreilly.com

想了解关于O'Reilly图书、课程、会议和新闻的更多信息，请访问以下网站：

http://www.oreilly.com.cn

http://www.oreilly.com

还可以通过以下网站关注我们：

我们在Facebook上的主页：*http://facebook.com/oreilly*

我们在Twitter上的主页：*http://twitter.com/oreillymedia*

我们在YouTube上的主页：*http://www.youtube.com/oreillymedia*

第1章

准备工作

本书主要内容

本书讲的是利用Python进行数据控制、处理、整理、分析等方面的具体细节和基本要点。同时，它也是利用Python进行科学计算的实用指南（专门针对数据密集型应用）。本书重点介绍了用于高效解决各种数据分析问题的Python语言和库。本书没有阐述如何利用Python实现具体的分析方法。

当书中出现“数据”时，究竟指的是什么呢？主要指的是结构化数据（structured data），这个故意含糊其辞的术语代指了所有通用格式的数据，例如：

- 多维数组（矩阵）。
- 表格型数据，其中各列可能是不同的类型（字符串、数值、日期等）。比如保存在关系型数据库中或以制表符/逗号为分隔符的文本文件中的那些数据。
- 通过关键列（对于SQL用户而言，就是主键和外键）相互联系的多个表。
- 间隔平均或不平均的时间序列。

这绝不是一个完整的列表。大部分数据集都能被转化为更加适合分析和建模的结构化形式，虽然有时这并不是很明显。如果不行的话，也可以将数据集的特征提取为某种结构化形式。例如，一组新闻文章可以被处理为一张词频表，而这张词频表就可以用于情感分析。

大部分电子表格软件（比如Microsoft Excel，它可能是世界上使用最广泛的数据分析工具了）的用户不会对此类数据感到陌生。

为什么要使用Python进行数据分析

许许多多的人（包括我自己）都很容易爱上Python这门语言。自从1991年诞生以来，Python现在已经成为最受欢迎的动态编程语言之一，其他还有Perl、Ruby等。由于拥有大量的Web框架（比如Rails（Ruby）和Django（Python）），最近几年非常流行使用Python和Ruby进行网站建设工作。这些语言常被称作脚本（scripting）语言，因为它们可以用于编写简短而粗糙的小程序（也就是脚本）。我个人并不喜欢“脚本语言”这个术语，因为它好像在说这些语言无法用于构建严谨的软件。在众多解释型语言中，Python最大的特点是拥有一个巨大而活跃的科学计算（scientific computing）社区。进入21世纪以来，在行业应用和学术研究中采用Python进行科学计算的势头越来越猛。

在数据分析和交互、探索性计算以及数据可视化等方面，Python将不可避免地接近于其他开源和商业的领域特定编程语言/工具，如R、MATLAB、SAS、Stata等。近年来，由于Python有不断改良的库（主要是pandas），使其成为数据处理任务的一大替代方案。结合其在通用编程方面的强大实力，我们完全可以只使用Python这一种语言去构建以数据为中心的应用程序。

把Python当做粘合剂

作为一个科学计算平台，Python的成功部分源于其能够轻松地集成C、C++以及Fortran代码。大部分现代计算环境都利用了一些Fortran和C库来实现线性代数、优选、积分、快速傅里叶变换以及其他诸如此类的算法。许多企业和国家实验室也利用Python来“粘合”那些已经用了30多年的遗留软件系统。

大多数软件都是由两部分代码组成的：少量需要占用大部分执行时间的代码，以及大量不经常执行的“粘合剂代码”。粘合剂代码的执行时间通常是微不足道的。开发人员的精力几乎都是花在优化计算瓶颈上面的，有时更是直接转用更低级的语言（比如C）。

最近这几年，Cython项目（*http://cython.org*）已经成为Python领域中创建编译型扩展以及对接C/C++代码的一大途径。

解决“两种语言”问题

很多组织通常都会用一种类似于领域特定的计算语言（如MATLAB和R）对新的想法进行研究、原型构建和测试，然后再将这些想法移植到某个更大的生产系统中去（可能是用Java、C#或C++编写的）。人们逐渐意识到，Python不仅适用于研究和原型构建，同时也适用于构建生产系统。我相信越来越多的企业也会这样看，因为研究人员和工程技术人员使用同一种编程工具将会给企业带来非常显著的组织效益。

为什么不选Python

虽然Python非常适合构建计算密集型科学应用程序以及几乎各种各样的通用系统，但它对于不少应用场景仍然力有不逮。

由于Python是一种解释型编程语言，因此大部分Python代码都要比用编译型语言（比如Java和C++）编写的代码运行慢得多。由于程序员的时间通常都比CPU时间值钱，因此许多人也愿意在这里做一些权衡。但是，在那些要求延迟非常小的应用程序中（例如高频交易系统），为了尽最大可能地优化性能，耗费时间使用诸如C++这样更低级、更低生产率的语言进行编程也是值得的。

对于高并发、多线程的应用程序而言（尤其是拥有许多计算密集型线程的应用程序），Python并不是一种理想的编程语言。这是因为Python有一个叫做全局解释器锁（Global Interpreter Lock，GIL）的东西，这是一种防止解释器同时执行多条Python字节码指令的机制。有关“为什么会存在GIL”的技术性原因超出了本书的范围，但是就目前来看，GIL并不会在短时间内消失。虽然很多大数据处理应用程序为了能在较短的时间内完成数据集的处理工作都需要运行在计算机集群上，但是仍然有一些情况需要用单进程多线程系统来解决。

这并不是说Python不能执行真正的多线程并行代码，只不过这些代码不能在单个Python进程中执行而已。比如说，Cython项目可以集成OpenMP（一个用于并行计算的C框架）以实现并行处理循环进而大幅度提高数值算法的速度。

重要的Python库

考虑到那些还不太了解Python科学计算生态系统和库的读者，下面我先对各个库做一个简单的介绍。

NumPy

NumPy（Numerical Python的简称）是Python科学计算的基础包。本书大部分内容都基于NumPy以及构建于其上的库。它提供了以下功能（不限于此）：

- 快速高效的多维数组对象ndarray。
- 用于对数组执行元素级计算以及直接对数组执行数学运算的函数。
- 用于读写硬盘上基于数组的数据集的工具。
- 线性代数运算、傅里叶变换，以及随机数生成。
- 用于将C、C++、Fortran代码集成到Python的工具。