清華
科技大讲堂
前沿·科技·分享

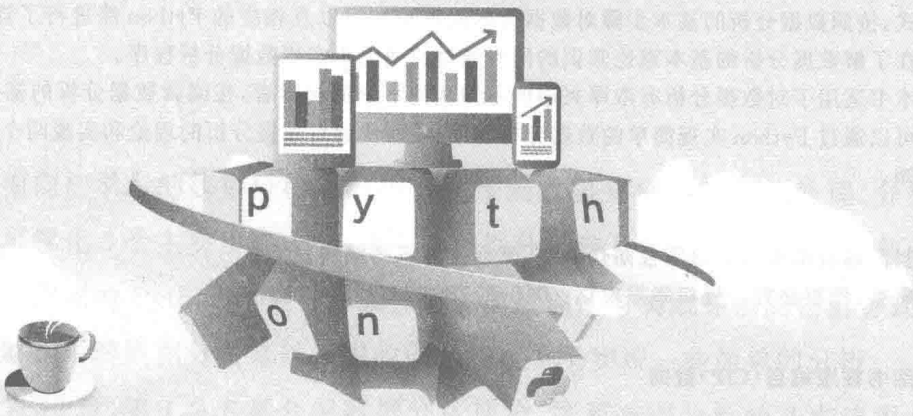
Python 数据分析实战

吕云翔 李伊琳 王肇一 张雅素 ◎ 编著

深入浅出，语言精练
图文并茂，范例典型
提供实例源代码和电子课件

清華大學出版社

清華
科技大讲堂
前沿·科技·分享



Python 数据分析实战

吕云翔 李伊琳 王肇一 张雅素 ◎ 编著

清華大學出版社
北京

内 容 简 介

使用 Python 进行数据分析是十分便利且高效的，因此它被认为是最优秀的数据分析工具之一。本书从理论和实战两个角度对 Python 数据分析工具进行了介绍，并采用理论分析和 Python 实践相结合的形式，按照数据分析的基本步骤对数据分析的理论知识以及相应的 Python 库进行了详细的介绍，让读者在了解数据分析的基本理论知识的同时能够快速上手实现数据分析程序。

本书适用于对数据分析有浓厚兴趣但不知从何下手的初学者，在阅读数据分析的基础理论知识的同时可以通过 Python 实现简单的数据分析程序，从而快速对数据分析的理论和实现两个层次形成一定的认知。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

Python 数据分析实战/吕云翔等编著. —北京：清华大学出版社，2019
(清华科技大讲堂)
ISBN 978-7-302-51838-9

Ⅰ. ①P… Ⅱ. ①吕… 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(2018)第 272081 号

策划编辑：魏江江
责任编辑：王冰飞
封面设计：刘　键
责任校对：徐俊伟
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：010-62770175　　**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
课件下载：http://www.tup.com.cn,010-62795954
印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：12.25　　**字　　数**：211 千字
版　　次：2019 年 1 月第 1 版　　**印　　次**：2019 年 1 月第 1 次印刷
印　　数：1～2000
定　　价：39.00 元

产品编号：077077-01

前言

本书是面向初学者的数据分析入门指南。按照数据分析的数据预处理、分析与知识发现和可视化3个主要步骤，本书逐步对数据分析涉及的理论进行讲解，并对实现这些步骤所用到的Python库进行详细介绍。通过理论与实践穿插的讲解方式，本书使读者能够在了解数据分析基础知识的同时快速上手实现一些简单的分析。

全书分为10章，第1、3、6章介绍数据分析理论，按照数据分析的基本流程介绍了理论知识和一些常用方法，穿插在理论章节之间的Python实战章节可以让读者在了解理论之后用相应的Python库来进行实战操作。通过阅读第1～8章的内容，读者已经对数据分析的各主要流程形成了一定的认识，但这些知识可能还未形成一个完整的体系，因此本书在第9和第10章引入了两个完整的数据分析实例，帮助读者建立知识点之间的联系，形成对数据分析整个知识面的清晰认知。建议读者在阅读实战章节时跟随介绍自己动手尝试一下，这样一定会发现数据的魅力所在。

作为一本数据分析入门书籍，本书着重介绍基础知识，对前沿的内容涉及较少，这些内容留待读者在更进一步的学习中深入探索。对于Python语言的知识，本书仅对与数据分析相关的库进行了介绍，如果读者对Python语言本身有兴趣，可以参考Python语言工具书及官方文档等详细了解Python的语法和底层原理等。另外，本书所有数据分析程序的实现均在单机情况下进行，并没有对如何使用Python进行分布式数据分析作介绍，有兴趣的读者可以了解一下Python分布式数据分析的相关库，如pyspark等。

本书主要由吕云翔、李伊琳、王肇一、张雅素编写，曾洪立、吕彼佳、姜彦华也参与了部分内容的编写并进行了素材整理及配套资源制作等。

源码下载

由于作者的水平和能力有限，本书难免有疏漏之处，恳请各位同仁和广大读者给予批评指正，也希望各位能将实践过程中的经验和心得与我们交流(yunxianglu@hotmail.com)。

作　者

2018年9月

目 录

试读结束：需要全本请在线购买： www.ertongbook.com

第1章

数据分析是什么

1.1 海量数据背后蕴藏的知识

自古以来，人们观察世界中的对象，对观察得到的数据进行分析，从而发现各种规律和法则，例如开普勒通过天体观测数据发现了开普勒定律。通过记录过去发生的事情，可推断得到一些可能的规律，这些规律可以解释当前发生的事情，并可用于对未来进行预测。在这个过程中，数据是十分宝贵的材料，其背后蕴藏着能够指导未来的知识。

随着计算机数据库技术的发展成熟和计算机的普及深化，各行各业每天都在产生和收集大量数据。例如，社交网络媒体每天产生的数据十分惊人，2012 年的微博日发量高达 4 亿条，Twitter 的信息量几乎每年都在翻番增长，另外各种商业领域、政府部门累计的数据量也令人瞠目。管理者们希望从数据中获得隐藏在数据中的有价值的信息来帮助决策，例如在制造业中，决策者需要了解客户偏好，设计受欢迎的产品；需要制定合适的价格，在确保利润的同时保证市场；需要了解市场需求，调整生产计划等。但是面对海量、无序的数据，如果管理者们得不到想要的信息，就会造成

信息爆炸的问题。数据分析的任务则是尝试将这些数据赋予意义，并为决策提供参考。

1.2 数据分析与数据挖掘的关系

传统的统计分析是在已定假设、先验约束上，对数据进行整理、筛选和加工，由此得到一些信息，而这些信息需要得到进一步的认知，用于有效的预测和决策，这样的过程则是数据挖掘的过程。统计分析是把数据变成信息的工具，数据挖掘是把信息变成认知的工具。广义上的数据分析是指整个过程，即从数据到认知。本书是指广义上的数据分析，将统计分析部分放入数据预处理阶段，即数据经整理、筛选、加工转换为信息的过程；将挖掘部分放入数据分析与知识发现阶段，即将信息进一步处理，获得认知，并进行预测和决策的过程。

1.3 机器学习与数据分析的关系

机器学习是人工智能的核心研究领域之一，最初的目的是让机器具有学习能力，从而拥有智能，目前公认的定义是利用经验来改善计算机系统自身的性能。由于“经验”在计算机系统中主要以数据形式存在，因此机器学习需要对数据进行分析。

数据分析的定义则是识别出海量数据中有效的、新颖的、潜在有用的、最终可理解的模式的非平凡过程，即从海量数据中找到有用的知识，主要利用机器学习领域提供的技术来分析海量数据。

1.4 数据分析的基本步骤

数据分析的步骤为数据收集—数据预处理—数据分析与知识发现—数据后处理。

1. 数据收集

之前的数据收集包含抽样、测量、编码、输入、核对等操作，这是一种主动的收集

试读结束：需要全本请在线购买： www.ertongbook.com

数据的方法。

如今由于传感器、照相机等电子设备普及，大量的数据会涌入，无法像传统的数据收集那样得到少而精的数据，而是产生了大量的、冗余的但是信息量少的数据，从这样的数据中得到所需要信息的过程是目前数据分析的重点和难点，也是本书的主要关注点。

2. 数据预处理

数据预处理完成从数据到信息的转化过程：首先对数据进行初步的统计方面的分析，得到数据的基本档案；其次分析数据质量，从数据的一致性、完整性、准确性和及时性4个方面进行分析；再次根据发现的数据质量问题对数据进行清洗，包括缺失值处理、噪声处理等；最后对其进行特征抽取，为后续的数据分析工作做准备。

3. 数据分析与知识发现

数据分析与知识发现则是将预处理后的数据进行进一步分析，完成从信息到认知的转化过程。从整理后的数据中学习和发现知识，主要分为有监督的和无监督的。有监督的分析包括分类分析、关联分析和回归分析；无监督的分析包括聚类分析、异常检测。

4. 数据后处理

数据后处理主要包括提供数据给决策支撑系统、数据可视化等。本书主要关注数据可视化的一些内容。

1.5　Python 和数据分析

数据分析需要与数据进行大量的交互、探索性计算以及过程数据和结果的可视化等，过去有很多专用于实验性数据分析或者领域的特定语言，如R语言、MATLAB、SAS、SPSS等。与这些语言相比，Python具有以下优点：

1. Python 是面向生产的

大部分数据分析过程都是首先进行实验性的研究、原型构建，再移植到生产系统

中。上述语言都无法直接用于生产，需要使用 C/C++语言等对算法进行再次实现；而 Python 是多功能的，不仅适用于原型构建，还可以直接运用到生产系统中。

2. 强大的第三方库的支持

Python 是多功能的语言，数据统计更多的是通过第三方的库来实现的，常用的有 NumPy、SciPy、Pandas、scikit-learn、Matplotlib 等，具体每个库的功能将在第 2 章中介绍。在上述提到的语言中，只有 R 语言和 Python 语言是开源的，由很多人共同维护，对于新的需求可以很快地付诸实践。

3. Python 的胶水语言特性

Python 的底层可以用 C 语言来实现，一些底层用 C 语言写的算法封装在 Python 包中能显著提高性能。例如 NumPy 底层是用 C 语言实现的，所以对于很多运算，它的速度都比用 R 语言等语言实现的要快。

第2章

Python——从了解Python开始

2.1 Python 的发展史

1989 年的圣诞节，荷兰数学家、计算机学家 Guido von Rossum 为了打发无聊的假期，着手设计了一门新的脚本解释型编程语言。他希望这门语言能够像 Shell 语言一样方便，同时又能像 C 语言一样可以调用众多系统接口。Guido 将这种介于 C 与 Shell 之间的语言命名为 Python，这个名称来源于他最爱的电视剧。1991 年，Python 的第 1 个公开发行版问世。Python 的后续版本不断发行，其中最重大的升级出现在 2000 年 10 月发行的 Python 2.0 和 2008 年 12 月发行的 Python 3.0 版本中。在 Python 2.0 中增加了许多新特性，包括垃圾回收机制和对 Unicode 的支持；在 Python 3.0 中去掉了 2.x 系列版本中冗余的关键字，使 Python 更加规范、简洁，并进一步完善了对 Unicode 的支持。值得注意的是，Python 3.x 系列版本不支持向下兼容。Python 2.x 系列的最新版本为 2010 年 7 月发行的 2.7 版本，官方将在 2020 年停止对该版本的支持。

自 1991 年至今，Python 经过了大大小小多次升级变革，发展成为简洁、人气颇高

的编程语言，受到了众多编程人员的青睐，这与 Python 社区的支持和贡献是分不开的。社区人员贡献的大量模块能够支持 Python 方便地完成包括机器学习、图像处理、科学计算等在内的多种多样的任务，这也吸引了越来越多的编程人员成为 Python 社区的一员。

2.2 Python 及 Pandas、scikit-learn、Matplotlib 的安装

2.2.1 Windows 环境下 Python 的安装

在 Windows 系统下安装 Python 的过程非常简单，只需要到官网上[①]下载相应的安装程序即可。网页会自动识别计算机的操作系统，并在最醒目的位置提供该操作系统对应的最高版本安装程序的下载链接。需要注意的是，安装程序并未默认选中“将 Python 3.6 加入到系统环境变量 PATH 中”这一选项，如果在安装时未选中此选项，需要在安装完毕后手动将安装路径加入到环境变量 PATH 中，否则系统无法找到 Python 命令。

2.2.2 Mac 环境下 Python 的安装

Mac 系统需要使用 Python，因此该系统中已经预装了某个版本的 Python。但在通常情况下，开发者需要一个更新的 Python 版本，此时需要注意保留系统中原有的 Python 版本，否则可能会影响系统的稳定性。在 Mac 系统下安装 Python 有两种常用方法，一种是使用 homebrew 安装；另一种是使用官网的 installer 安装。在使用 homebrew 安装时，如果安装 Python 2.x 版本，可以直接在终端中输入：

```
brew install python
```

如果是安装 Python 3.x 版本，需要输入：

```
brew install python3
```

如果需要查看上述 Python 版本，可以输入：

① https://www.Python.org/downloads/

```
brew info python
```

在使用 homebrew 安装 Python 时，无法选择 Python 在 2. x 及 3. x 系列下的具体版本，版本可能也不是最新的。除此之外，对 Mac 系统不熟的用户可能会出现一些意想不到的问题，因此这里推荐使用官网的 installer 进行安装。和 Windows 系统下 Python 的安装类似，用户首先需要去官网下载相应版本的 installer(mac OS 64-bit/32-bit 版)，然后按照向导提示进行安装即可。

2.2.3 Pandas、scikit-learn 和 Matplotlib 的安装

和其他第三方包相同，本书用到的 3 个主要包 Pandas、scikit-learn 和 Matplotlib 都可以使用 pip 进行安装。pip 是 Python 的第三方包管理器，在此我们不做详细的介绍。这里使用 pip 进行安装，如果系统中已经安装了 pip，则直接在终端依次输入以下命令即可完成安装：

```
pip install pandas
pip install scikit - learn
pip install matplotlib
```

自 3.4 版本开始，在安装 Python 的同时也会安装 pip。如果用户使用的是较低版本的 Python，则需要手动安装 pip，但将 Python 升级到最新版本也许是一个更好的选择。

2.2.4 使用科学计算发行版 Python 进行快速安装

除了安装官方的标准 Python 版本以及手动安装所需的各 Python 包以外，还有一种更加简单的 Python 安装方法——使用第三方科学计算发行版 Python。这类发行版一般会将一个标准版本的 Python 和众多的包集成在一起，免去手动安装科学计算库的步骤，安装和使用都较为方便。现在流行的几款科学计算发行版 Python 如下。

Anaconda[①]：Anaconda 包括一个标准版本的 Python(目前有 2. 7、3. 5 和 3. 6 3 个版本可以选择)、一个 Python 包管理器 conda 和 100 多个科学计算功能 Python

① https://www.continuum.io/anaconda-overview

包。Anaconda 包括 Jupyter、Spyder 和 Visual Studio 等多个开源开发环境，还支持 Sublime Text 2 和 PyCharm。Anaconda 目前发行了 Windows、Mac、Linux 几个平台的版本，因此无论对于哪个平台的用户都是很好的选择。

WinPython[①]：WinPython 是 Windows 系统上的一个 Python 科学计算发行版，和 Anaconda 类似，它也包含一个标准 Python 版本、一个 Python 包管理器 WPPM（WinPython Package Manager）和众多科学计算功能 Python 包，内置 Spyder、Jupyter 和 IDLE 等编辑器。WinPython 的最大特点是便携（Portable），它是一个绿色软件，不会写入 Windows 注册表，所有的文件都位于一个文件夹中，将这个文件夹放置到移动存储设备中甚至是其他设备上也能够运行。

2.3 Python 基础知识

本节将会用一段功能较为简单的程序来简要介绍 Python 语言的基础知识，对 Python 语言有一定了解的读者可以跳过此节，而基础较弱的读者如果无法看懂本节所介绍的知识点，可以阅读更多的 Python 基础教程，在开始打好坚实的 Python 语言基础将会为接下来的数据分析实战做良好的铺垫。Code 2-1 是一段简单的 Python 小程序，用于计算斐波那契数列的前 10 项，并将结果存入文件中。

Code 2-1 Python 代码实例：求斐波那契数列

```
1:  #Fibonacci sequence
2:  '''
3:  斐波那契数列
4:  输入：项数 n
5:  输出：前 n 项
6:  '''
7:  import os
8:
9:  def fibo(num):
10:     numbers = [1,1]
11:     for i in range(num - 2):
12:         numbers.append(numbers[i] + numbers[i + 1])
13:     return numbers
```

① http://winPython.github.io

试读结束：需要全本请在线购买： www.ertongbook.com