数据分析与决策技术丛书

HZ BOOKS 华章IT

[PACKT] PUBLISHING

Python Data Analysis Cookbook

Python数据分析实战

[印] 伊凡·伊德里斯（Ivan Idris） 著

冯博 严嘉阳 译

通过140多个实例，详细讲解用Python进行数据分析的各种实用技术及最佳实践，并提供一个包含各种工具的Docker镜像

机械工业出版社
China Machine Press

数据分析与决策
技术丛书

Python Data Analysis Cookbook

Python数据分析实战

[印] 伊凡·伊德里斯（Ivan Idris） 著

冯博 严嘉阳 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Python 数据分析实战 /（印）伊凡·伊德里斯（Ivan Idris）著；冯博，严嘉阳译 . —北京：机械工业出版社，2017.8
（数据分析与决策技术丛书）
书名原文：Python Data Analysis Cookbook

ISBN 978-7-111-57640-2

I. P… II. ①伊… ②冯… ③严… III. 软件工具－程序设计 IV. TP311.561

中国版本图书馆 CIP 数据核字（2017）第 191413 号

本书版权登记号：图字：01-2016-8648

Python 数据分析实战

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：李秋荣
印 刷：北京市荣盛彩色印刷有限公司 版 次：2017 年 8 月第 1 版第 1 次印刷
开 本：186mm×240mm 1/16 印 张：21.5
书 号：ISBN 978-7-111-57640-2 定 价：79.00 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 88379649 68995259 读者信箱：hzit@hzbook.com

The Translator's Words 译者序

Python 语言诞生已经超过 25 年，距离 Python 3 发布也已经快 10 年了。经过大浪淘沙，Python 却依旧势头强劲，长期稳居编程语言市场占有率前十，甚至前五。各个领域都能看到 Python 的身影，从 Web 开发到数据挖掘，从网络爬虫到图像分析等。这从一个侧面也印证了一门编程语言要想“成功”，除了优良的语言本身特性之外，强大的生态圈也必不可少。

正式使用 Python 参与一个 Web 开发项目之后，译者就被 Python 语言本身之优美、框架之强大所吸引。随着不断了解，发现 Python 在数据分析领域是当仁不让的翘楚。虽然一些语言也偏向于数据分析，例如 R 语言，但是 Python 却和工程实践结合得更紧密，一门语言就可以让读者开发包含强大数据分析的后台 Server。

本书是《 Python Data Analysis 》的后续，如果说前一本书更偏向于介绍各种工具库和数据分析技术的使用，例如用于科学计算的库 SciPy/NumPy，用于操作数据的库 pandas，用于机器学习的库 scikit-learn，用于绘制图形的库 matplotlib，那么本书则更侧重于将这些技术应用于实际领域，解决实际的问题。作者 Ivan Idris 曾是 Java 和数据库应用开发者，后专注于 Python 和数据分析领域，致力于编写干净、可测试的代码。他还是《 Python Machine Learning By Example 》《 NumPy Cookbook 》等书的作者，在工程实践和书籍撰写方面都非常有经验。虽然译者预先已有心理准备，但是当真正开始翻译时，才被作者领域涉及之多、知识之渊博所折服。读者不但能从本书中找到 Python 用于数据分析的典型案例，例如信号处理、聚类等，甚至还能找到财务数据用于股票市场的分析。可以说这本书涵盖了 Python 在数据分析领域的方方面面。不但如此，作者还创建了一个包含各种工具的 Docker 镜像，方便读者使用。诚然，这也给翻译带来了不小的难度。有些领域过于专业，虽然查阅了大量资料，但限于译者本身水平所限，可能仍然存在错误，希望读者不吝指教。

最后，这是一本 Cookbook（食谱式手册）成为我翻译本书的另一个原因。各种 Cookbook 一直是译者学习各种技术的参考书，一方面它更偏向于实践，另一方面它更直接，包含各种实例，本书也是如此。大量代码片段和图例一定能帮助读者快速掌握用 Python 进行数据分析的各种技术。

前　言 *Preface*

数据分析是 Python 的杀手锏。

——匿名

本书是《 Python Data Analysis 》的后续。那么在《 Python Data Analysis 》已经足够优秀（我愿意这么认为）的情况下，这本书有哪些新的内容吗？本书是针对那些有一定经验的 Python 程序员写的。一年时间过去了，因此，我们将使用在《 Python Data Analysis 》中没有用到的一些更新版本的软件和软件库。另外，经过深入反思和调研，我做出了以下的总结：

- 为了减轻自己的负担，同时提高代码的可重复使用率，我需要一个工具箱，我将这个工具箱命名为 dautil 并将它发布到了 Pypi 上（可以通过 pip/easy_install 安装）。
- 通过反省，我深信需要简化获取和安装所需要的软件的过程，因此我通过 DockerHub 发布了一个包含了我们需要用到的软件的 Docker 容器（pydacbk），在本书的第 1 章和线上章节中你将了解更多关于如何安装的细节。这个 Docker 容器还是不够理想，因为它的体积已经相当大，所以我需要做出一些艰难的决定。因为这个容器并不是本书的一部分，所以如果你有任何问题可以直接与我联系，但是请记住我不会对镜像做很大的修改。
- 本书会使用 IPython Notebook，这个工具已经成为数据分析时的标准工具。在线上章节以及我写的其他书中，我已经给出了一些和 IPython Notebook 相关的建议。
- 除了极少数案例外，本书中我主要使用的是 Python 3，因为 2020 年后官方将不再支持 Python 2。

为什么需要这本书

有人会说你并不需要书籍，你只需要去做一个感兴趣的项目，然后在做项目的同时就会搞明白那些东西。但是尽管接触到大量的资源，这个过程可能还是会令你感到沮丧。打个比方，如果想烹调一碗美味的汤，你可以去向朋友和家人寻求帮助，上网搜索或者收看烹饪

节目，但是朋友和家人不会一直在你身边，网络上的内容也是良莠不齐。以我的浅见，出版社、审稿人和作者都在这本书上花费了大量的时间和精力，如果你不能从中有所收获我会感到很诧异。

数据分析、数据科学、大数据——有什么了不起的

你应该看过将数据科学用数学 / 统计学、计算机科学以及专业领域的知识进行描述的维恩图（Venn diagram）。数据分析是永恒的，它出现在数据科学之前，甚至是计算机科学之前。你可以用笔和纸或者更先进的便携计算器进行数据分析。

数据分析体现在很多方面，比如说以做出决策或提出新的假设和问题为目的进行数据分析。数据科学以及大数据的热潮、高待遇以及经济回报让我想起了当数据存储和商业智能还是时髦词的年代。商业智能和数据存储的终极目标是构建应用于管理的可视化图表。这涉及很多政治和组织方面的利益，但是从技术的角度来看，这主要还是和数据库相关。数据科学则不是以数据库为中心，而是很大程度上依赖于机器学习。由于数据的量在不断地变多，机器学习变得越来越不可或缺。数据大量增长的背后是人口的快速增长以及新技术的层出不穷，比如说社交媒体和移动设备的出现。事实上，数据增长可能是我们唯一可以肯定的将一直持续的趋势。构建可视化图表和应用机器学习的区别就类似于搜索引擎的演进。

搜索引擎（如果可以这么称呼）最初只是手动创建的组织良好的链接集合。而最终，纯自动的方式取代了前者。当下，更多的数据将会被创建（而不是被销毁），我们可以预见自动化数据分析领域的增长。

Python 数据分析的简要历程

各个 Python 软件库的历史十分有趣，但我不是一个历史学家，所以下面的记录主要从我的视角来写：

- 1989 年：Guido Van Rossum 在荷兰的 CWI 实现了 Python 的第一个版本，当时是作为一个圣诞节的“兴趣”项目。
- 1995 年：Jim Hugunin 创建了 Numeric——Numpy 的前身。
- 1999 年：Pearu Peterson 写了 f2py 作为连接 Fortran 和 Python 的桥梁。
- 2000 年：Python 2.0 发布。
- 2001 年：SciPy 库发布，同期创建的还有与 Numeric 竞争的库 Numarray。Fernando Perez 发布了 IPython，它最初是以“午后黑客”（afternoon hack）的名义发布的。NLTK（自然语言工具包）发布且用于研究项目。
- 2002 年：John Hunter 创建了 Matplotlib 库。

- 2005 年：Travis Oliphant 发布了 NumPy，NumPy 最初是受 Numarray 启发而对 Numeric 进行扩展的库。
- 2006 年：NumPy 1.0 发布，第 1 版 SQLAlchemy 发布。
- 2007 年：David Cournapeau 将 scikit-learn 作为 Google Summer of Code 的项目，Cython 在 Pyrex 的基础上开始开发，Cython 后来集中用在了 pandas 和 scikit-learn 上以提升性能。
- 2008 年：Wes McKinney 开始开发 pandas，Python 3.0 发布。
- 2011 年：IPython 0.12 发行版本中引入了 IPython Notebook，Packt 出版社出版了《NumPy 1.5 Beginners Guide》。
- 2012 年：Packt 出版社出版了《NumPy Cookbook》。
- 2013 年：Packt 出版社出版了第 2 版的《NumPy Beginners Guide》。
- 2014 年：Fernando Perez 宣布了 Jupyter 项目，致力于开发与语言无关的 Notebook，Packt 出版社出版了《Learning NumPy Array》和《Python Data Analysis》。
- 2015 年：Packt 出版社出版了第 3 版的《NumPy Beginners Guide》以及第 2 版的《NumPy Cookbook》。

对未来的猜想

未来将会是一片光明，难以计数的数据将会存在于云上，软件运行在各种具有直观的自定义界面的设备上。（我知道会有年轻人不厌其烦地夸赞他们的手机是多么厉害，以及终有一天我们将通过拖放式操作在平板电脑上进行编程。）Python 社区里有人担忧他们的技术会与未来格格不入。而且当你在 Python 上投入得越多，这种担忧会越强烈。

要弄清楚我们能做什么，就需要知道 Python 有何独特之处。有的学派认为 Python 是一门胶水语言（glue language），其融合了 C、Fortran、R、Java 以及其他一些语言的特性，因此我们只需要更好的胶水。这可以理解为从其他的语言那里去“借”来一些特性。从我个人角度来说，我喜欢 Python 的工作方式，包括它的灵活性、数据结构，以及它拥有的相当数量的库和特性。我认为代码的未来在于更加美味的语法糖以及即时编译器。因此我们应该能够继续编写 Python 代码，因为它能自动地将代码转成并发（机器）代码。它以我们察觉不到的一些机制管理着低层级的细节，并给 CPU、GPU 或者云计算发送数据和指令。代码需要能够和我们使用的各种后端存储进行通信。理想情况下，所有的这些“魔法”会像自动垃圾回收一样便捷，这听起来就像不可能实现的“一键完成”的梦，但我觉得这值得我们去追求。

本书主要内容

第 1 章　非常重要，建议不要跳过。这一章会介绍 Anaconda、Docker、单元测试、日志

以及一些在进行可重复的数据分析时不可或缺的部分。

第 2 章　会演示如何进行数据可视化以及常见的陷阱。

第 3 章　会讨论两个变量间的统计概率分布及其相关性。

第 4 章　讨论异常和其他常见的数据问题。数据几乎从来没有完美过，因此需要进行大量的分析来处理数据中的缺陷。

第 5 章　本章的重点不在数学上，而是关注一些技术话题，比如数据库、网络抓取以及大数据。

第 6 章　介绍时间序列数据，这类数据的数据量巨大。因此需要独特的技术来处理。通常我们关注的是数据的趋势、季节性和周期性。

第 7 章　关注股票投资，这是因为股价的数据量巨大。这是唯一和金融有关的章节，即使你对股票不感兴趣，这一章也值得阅读，因为有些内容是和数据分析相关的。

第 8 章　将帮你去应对洪水般的文本和社交媒体信息。

第 9 章　涵盖集成学习、分类和回归算法，以及分层聚类。

第 10 章　评估第 9 章的分类器、回归器、集成学习与降维。

第 11 章　将多次使用 OpenCV 来分析图像。

第 12 章　涉及软件性能，本章将讨论各种提升软件性能的方法，包括缓存和即时编译器。

附录 A　包含了本书中用到的技术概念的一个简单的词汇表，以帮助读者更好地查询相关信息。

附录 B　包含一些函数的简单参考，这会在你临时无法查看文档时提供一些额外的帮助。

附录 C　包含演示文档、文档链接，以及一些免费提供的 IPython Notebook 和数据的资源列表，这个附录将作为在线章节提供。

附录 D　对本书中用到的许多工具，比如 IPython Notebook、Docker 以及 Unix shell 命令给出一个简短的提示列表，可能不会面面俱到。同样这个章节也是作为在线章节提供。

阅读准备

首先需要安装 Python 3 发行版，我推荐完整版的 Anacoda 版，因为它自带需要使用到的大部分软件，我用 Python 3.4 及以下包测试了代码：

- joblib 0.8.4
- IPython 3.2.1
- NetworkX 1.9.1
- NLTK 3.0.2

- ❑ Numexpr 2.3.1
- ❑ pandas 0.16.2
- ❑ SciPy 0.16.0
- ❑ Seaborn 0.6.0
- ❑ sqlalchemy 0.9.9
- ❑ statsmodels 0.6.1
- ❑ matplotlib 1.5.0
- ❑ NumPy 1.10.1
- ❑ scikit-learn 0.17
- ❑ dautil 0.0.1a29

在一些小节中，可能需要安装一些额外的软件，这些都会在需要使用软件的时候进行解释说明。

读者人群

本书重在动手，轻于理论。你需要比 Python 初学者掌握更多的知识，比如线性代数、微积分、机器学习和统计。你最好读过《 Python Data Analysis 》，但这并不是必需的，我同样推荐以下这些书：

- ❑《 Building Machine Learning Systems with Python 》，Willi Richert 和 Luis Pedro Coelho 著，2013。
- ❑《 Learning NumPy Array 》，Ivan Idris 著，2014。
- ❑《 Learning scikit-learn: Machine Learning in Python 》，Guillermo Moncecchi 著，2013。
- ❑《 Learning SciPy for Numerical and Scientific Computing 》，Francisco J. Blanco-Silva 著，2013。
- ❑《 Matplotlib for Python Developers 》，Sandro Tosi 著，2009。
- ❑《 NumPy Beginner's Guide，Third Edition 》，Ivan Idris 著，2015。
- ❑《 NumPy Cookbook，Second Edition 》，Ivan Idris 著，2015。
- ❑《 Parallel Programming with Python 》，Jan Palach 著，2014。
- ❑《 Python Data Visualization Cookbook 》，Igor Milovanović 著，2013。
- ❑《 Python for Finance 》，Yuxing Yan 著，2014。
- ❑《 Python Text Processing with NLTK 2.0 Cookbook 》，Jacob Perkins 著，2010。

说明

在本书中，你会看到一些经常出现的标题（准备工作、操作步骤、工作原理、更多信息、参见）。

为了清晰地组织章节，我们使用了如下的小节标题：

1. 准备工作

这部分告诉你本小节的目的是什么，以及描述如何安装这个示例中需要用到的软件和一些初步的设置。

2. 操作步骤

这部分包含了完成一个小节的步骤。

3. 工作原理

这部分通常包含了对上一个部分中发生的内容的细节解释。

4. 更多信息

这部分会给出一些关于这个示例的额外信息，帮助读者了解更多关于这个示例的知识。

5. 参见

这部分提供了关于这个示例的其他有用信息的链接。

本书约定

在本书中，你会发现一些用于区分不同类型信息的文本样式。以下是一些样式的例子以及对它们含义的解释。

如下是一个代码段：

```
population = dawb.download(indicator=[dawb.get_name('pop_grow'), dawb.
get_name('gdp_pcap'),
                                      dawb.get_name('primary_
education')],
                           country=countries['iso2c'], start=2014,
end=2014)

population = dawb.rename_columns(population)
```

当本书希望你关注某一段代码的时候，相关的行或者部分将会被加粗：

```
plt.figure()
plt.title('Rainy Weather vs Wind Speed')
categorical = df
categorical['RAIN'] = categorical['RAIN'] > 0
ax = sns.violinplot(x="RAIN", y="WIND_SPEED",
                           data=categorical)
```

任何命令行的输入或输出都按如下方式书写：

```
$ conda install -c scitools cartopy
```

表示警告或重要注释。

下载配套软件包

你可以从 http://www.packtpub.com 通过个人账号下载示例代码文件。如果你通过其他途径购买了本书，可以访问 http://www.packtpub.com/support 然后注册，我们会将文件直接电邮给你。

你也可以访问华章图书官网 http://www.hzbook.com，通过注册并登录个人账号，下载本书的源代码。

本书中使用到的代码同样存放在 Github 上，地址 https://github.com/PacktPublishing/Python-DataAnalysisCookbook，同样我们还有很多其他各类图书以及视频中的代码在 https://github.com/PacktPublishing/ 上提供，去发现它们吧。

Contents 目　录

第 1 章 Chapter 1

为可重复的数据分析奠定基础

1.1 简介

可重复的数据分析（reproducible data analysis）是良好科学的一块基石。在科学技术飞速发展的今天，可重复性是一个热门的话题。可重复性目的是为了减少与他人之间的理解障碍。这么说可能有点奇怪或者不必要，但是可重复性的分析对于让别人认可你的工作必不可少。如果有很多人都能证实你的分析结果，那么对你的职业生涯肯定会带来积极的影响，然而，可重复性的分析是困难的。它具有重要的经济影响，对此你可以阅读 Freedman LP, Cockburn IM, Simcoe TS（2015）《The Economics of Reproducibility in Preclinical Research》. PLoS Biol 13 (6): e1002165. doi:10.1371/journal.pbio.1002165 来了解。

所以可重复性无论对于社会还是你都非常重要，但是它如何适用于 Python 用户呢？我们可以通过如下的方法来减少与他人间的理解障碍：

- 提供我们使用的软件和硬件信息，包括版本。
- 共享虚拟环境。
- 记录程序的行为。
- 为代码写单元测试，这相当于是有序文档。
- 共享配置文件。
- 为随机数生成器设置种子，使程序行为尽可能地一致。
- 标准化生成的报告、数据访问和代码风格。

我为本书创建了一个工具包 dautil，它可以通过 pip 或者从本书附带的代码包安装。如果你比较赶时间，直接运行 $python install_ch1.py 可以安装本章用到的大部分软件，包括

dautil。如果你不想安装任何软件，除了 Docker，我还创建了一个 Docker 镜像，你可以使用它（参考 1.5 节）。

1.2 安装 Anaconda

Anaconda 是一个针对数据分析和科学计算免费发布 Python 的版本，它有自己的包管理器 conda。这个版本包含超过 200 个 Python 包，使用起来非常方便。对于临时用户，Miniconda 可能是一个更好的选择，Miniconda 包含了 conda 包管理器和 Python。技术编辑比如我，则使用 Anaconda。但是不用担心，我也会为那些不使用 Anaconda 的读者提供相应的安装说明。在这一节，我们将安装 Anaconda 和 Miniconda，并创建一个虚拟环境。

1. 准备工作

安装 Anaconda 和 Miniconda 的过程非常相似。显然，Anaconda 需要更多的磁盘空间。按照 Anaconda 网站 http://conda.pydata.org/docs/install/quick.html 上的安装说明进行安装（检索于 2016 年 3 月）。首先，你必须下载适合你操作系统和 Python 版本的安装程序。有时，你可以在 GUI 和命令行安装程序之间进行选择。我使用的是 Python 3.4 的安装程序，尽管我系统的 Python 版本是 v2.7。这是可行的，因为 Anaconda 会自带 Python。在我的机器上，Anaconda 安装程序会在 home 目录下创建一个 anaconda 目录，并需要大约 900MB 空间。Miniconda 安装程序则在 home 目录下创建一个 miniconda 的目录。

2. 操作步骤

（1）现在假设你已经安装了 Anaconda 或 Miniconda，使用下面的命令可以列出软件包：

```
$ conda list
```

（2）为了可重复使用，我们还可以把软件包信息导出：

```
$ conda list --export
```

（3）之前的命令会在屏幕上打印软件包和对应版本，你也可以把这些信息保存到文件中，之后通过下面的命令来重新安装它们：

```
$ conda create -n ch1env --file <export file>
```

这条命令同时还会创建名为 ch1env 的环境。

（4）下面的命令会创建一个简单的名为 testenv 的环境：

```
$ conda create --name testenv python=3
```

（5）在 Linux 和 Mac OS X 上，通过下面这条命令可以切换到 testenv 这个环境上：

```
$ source activate testenv
```

（6）在 Windows 上，我们不需要 source。退出时的语法也是类似的：