Become a Python Data Analyst

Python
数据分析师修炼之道

[美] 阿尔瓦罗·富恩特斯 著　　刘　璋 译

清华大学出版社

Python 数据分析师修炼之道

[美] 阿尔瓦罗·富恩特斯 著

刘 璋 译

清华大学出版社

北 京

内 容 简 介

本书详细阐述了与 Python 数据分析相关的基本解决方案，主要包括 Anaconda 和 Jupyter Notebook、NumPy 向量计算、数据分析库 pandas、可视化和数据分析、Python 统计计算、预测分析模型等内容。此外，本书还提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

本书既可作为高等院校计算机及相关专业的教材和教学参考书，也可作为相关开发人员的自学教材和参考手册。

北京市版权局著作权合同登记号 图字：01-2019-1262

图书在版编目（CIP）数据

Python 数据分析师修炼之道/（美）阿尔瓦罗·富恩特斯（Alvaro Fuentes）著；刘璋译. —北京：清华大学出版社，2019
书名原文：Become a Python Data Analyst
ISBN 978-7-302-53016-9

Ⅰ. ①P… Ⅱ. ①阿… ②刘… 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（2019）第 093947 号

责任编辑：贾小红
封面设计：刘　超
版式设计：魏　远
责任校对：马子杰
责任印制：沈　露

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
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　　张**：8.5　　**字　　数**：168 千字
版　　次：2019 年 6 月第 1 版　　**印　　次**：2019 年 6 月第 1 次印刷
定　　价：69.00 元

产品编号：082450-01

译 者 序

当今，Python 已经成为一种主流的编程语言，它易于读写，非常实用，从而赢得了广泛的群众基础，被无数程序员热烈追捧。Python 几乎在每个领域都表现得非常优秀，这是一门真正意义上的全栈语言。

此外，Python 也是数据分析人员和统计人员在处理大量数据集和复杂数据可视化方面最常见和最流行的语言之一。具体来说，开发人员往往需要在工作中应用统计技术或数据分析，或者需要与 Web 应用程序进行交互。特别是，Python 在机器学习中的地位，它的机器学习库和灵活性的结合使得 Python 非常适合开发复杂的模型并可以直接在应用中加以使用。

本书介绍了 Python 语言中的核心工具和库，以帮助读者与数据分析处理过程协同工作、准备相关数据以执行简单的统计学分析，进而构建具有实际意义的数据可视化结果。本书将讨论 Python 语言中的各种库，例如 NumPy、pandas、matplotlib、seaborn、SciPy 和 scikit-learn，并将其应用于实际数据分析和统计示例中。

在本书的翻译过程中，除刘璋外，王辉、刘晓雪、张博、刘祎、张华臻等人也参与了部分翻译工作，在此一并表示感谢。

由于译者水平有限，难免有疏漏和不妥之处，恳请广大读者批评指正。

译 者

前　言

Python 是高级数据分析师和统计人员所用的最常见和最流行的语言之一，可用于处理大型数据集和复杂的数据可视化任务。

本书介绍了 Python 语言中的核心工具和库，以帮助读者与数据分析处理过程协同工作、准备相关数据以执行简单的统计学分析，进而构建具有实际意义的数据可视化结果。本书将讨论 Python 语言中的各种库，如 NumPy、pandas、matplotlib、seaborn、SciPy 和 scikit-learn，并将其应用于实际数据分析和统计示例中。在阅读过程中，读者将会领略到如何高效地使用 Jupyter Noyebook，并借助于 NumPy 和 landas 库对数据进行操控。此外，还将利用 Python 库实现简单的预测模型、统计计算-分析和数据分析技术。

在阅读完本书后，读者在基于 Python 的数据分析方面将具备较为丰富的经验。

适用读者

本书面向初级数据分析师、数据工程师和 BI 专业人员，他们希望使用 Python 工具执行高效的数据分析。要理解本书所涉及的概念，读者应具备 Python 编程方面的一些背景知识。

本书内容

第 1 章：Anaconda 和 Jupyter Notebook。本章介绍了 Python 中一些较为重要的数据科学库，并对 Python 预测分析所用的主要对象、属性、方法和函数进行了整体描述。

第 2 章：NumPy 向量计算。本章讨论 NumPy 库，这也是 Python 项目中几乎全部科学计算所使用的库。学习如何使用 NumPy 数组，对于 Python 数据科学来说十分重要。

第 3 章：数据分析库 pandas。本章将整体介绍 pandas 库。对于 Python 编程语言来说，pandas 库提供了高性能、易于使用的数据结构和分析工具，因而受到了数据科学家以及 Python 社区开发者的喜爱。本章将通过相关示例展示如何利用 pandas 执行描述性分析。

第 4 章：可视化和数据分析。本章将考查数据科学的可视化效果。Python 针对不同的

功能提供了多种可视化选项。本章将学习两种最为流行的库，即 matplotlib 和 seaborn，并面向真实数据集执行探索性数据分析。

第 5 章：Python 统计计算。本章解释了如何利用 Python 执行统计计算，并据此考查包含青少年饮酒信息的数据集。

第 6 章：预测分析模型。本章简要介绍了预测分析，并通过构建一个模型对青少年的饮酒习惯进行预测。

资源下载

本书将引领读者整体了解 Python 中的数据分析过程、Python 数据科学栈中的主要库，并讨论如何使用各种 Python 工具有效地分析、可视化和处理数据。

读者可访问 http://www.packtpub.com 并通过个人账户下载示例代码文件。另外，在 http://www.packtpub.com/support 中注册成功后，我们将以电子邮件的方式将相关文件发与读者。

读者可根据下列步骤下载代码文件。

（1）访问 www.packtpub.com，利用电子邮件地址和密码登录，或注册。

（2）选择 SUPPORT 选项卡。

（3）单击 Code Downloads & Errata。

（4）在 Serach 文本框中输入书名。

当文件下载完毕后，确保使用下列最新版本软件解压文件夹。

- Windows 系统下的 WinRAR/7-Zip。
- Mac 系统下的 Zipeg/iZip/UnRarX。
- Linux 系统下的 7-Zip/PeaZip。

另外，读者还可访问 GitHub 获取本书的代码包，对应网址为 https://github.com/PacktPublishing/Become-a-Python-Data-Analyst。此外，读者还可访问 https://github.com/PacktPublishing/，以了解丰富的代码和视频资源。

下载彩色图像

另外，我们还进一步提供了本书所用截图/图表的彩色图像，读者可访问 http://www.

packtpub.com/sites/default/files/downloads/BecomeaPythonDataAnalyst_ColorImages.pdf 进行下载。

本书约定

本书通过不同的文本风格区分相应的信息类型。下面通过一些示例对此类风格以及具体含义的解释予以展示。

代码块如下所示。

```
# The largest heading
## The second largest heading
###### The smallest heading
```

当某个代码块希望引起读者的足够重视时，一般会采用黑体表示，如下所示。

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,Voicemail

(u100)
exten => s,102,Voicemail(b100)
exten =>

i,1,Voicemail(s0)
```

 图标则表示较为重要的说明事项。

 图标则表示提示信息和操作技巧。

读者反馈和客户支持

欢迎读者对本书的建议或意见予以反馈。对此，读者可向 feedback@packtpub.com 发送邮件，并以书名作为邮件标题。若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。若读者针对某项技术具有专家级的见解，抑或计划撰写书籍或完善某部著作的出版工作，则可访问 www.packtpub.com/authors。

勘误表

尽管我们在最大程度上做到尽善尽美，但错误依然在所难免。如果读者发现谬误之处，无论是文字错误抑或是代码错误，还望不吝赐教。对此，读者可访问 http://www.packtpub.com/submit-errata，选取对应书籍，然后单击 Errata Submission Form 超链接，并输入相关问题的详细内容。

版权须知

目　录

第 1 章　Anaconda 和 Jupyter Notebook

本书主要介绍基于 Python 的数据分析的基本概念。在第 1 章中，我们将学习如何安装 Anaconda，其中包含了本书所用的全部软件。此外，本章还将引入 Jupyter Notebook，这也是全部工作的计算环境。相应地，我们将通过具体解决方案帮助读者快速掌握相关工具。

本章主要涉及以下内容。

- Anaconda 及其所处理的问题。
- 如何在计算机设备上安装、启用 Anaconda。
- 通过 Jupyter Notebook 执行计算和分析任务。
- Jupyter Notebook 中一些有用的命令和快捷操作。

1.1　Anaconda

针对开发人员和数据科学家，Anaconda 是 Python 提供的一个免费、易于安装的包管理和环境管理工具，进而使得科学计算、数据科学、统计分析和机器学习中的包管理和部署更加简单。Anaconda 由 Continuum Analytics 推出，读者可访问 https://www.anaconda.com/download/免费下载。

Anaconda 是一个工具箱，即执行 Python 数据分析任务时的一个工具集。另外，读者也可免费下载独立的工具，但在后续操作中，获取整个工具箱则肯定更加方便。这也是 Anaconda 的用武之地——在查找各种工具并将其安装至系统的过程中，这将会节省大量的时间。除此之外，在单独安装 Python 包时，Anaconda 还负责处理所产生的包依赖关系，以及其他的潜在冲突和问题。

当访问 https://www.anaconda.com/download/时，可以看到针对不同操作系统的各种下载选项，读者需要根据自己的操作系统选取相应的安装程序。在下载页面中，读者将会看到两个安装程序，即 Python 3.7 和 Python 2.7，本书将采用 Python 3.7，如图 1.1 所示。

下载 Anaconda 软件的最新版本，并将其保存至 Downloads 文件夹中。

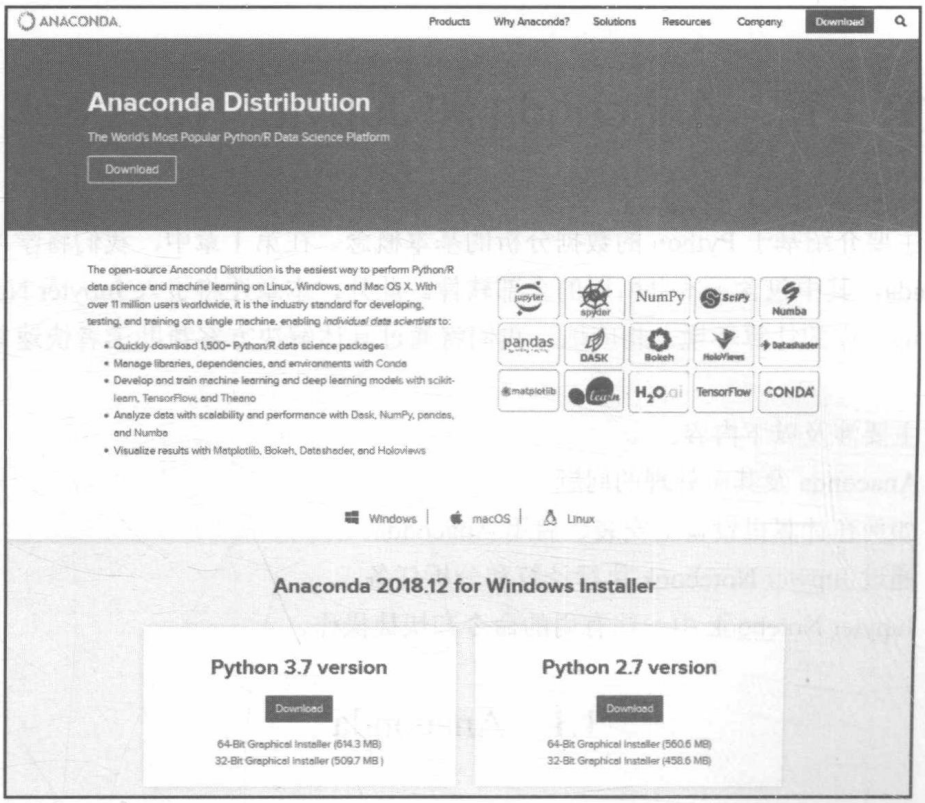

图 1.1

注意：

鉴于本书中的示例运行于 Windows 环境下，因而这里选择针对 Windows 的 64 位安装程序。macOS 和 Linux 的安装过程也基本类似。

Anaconda 的安装过程较为简单，且与其他软件的安装过程并无太多不同。双击.exe 文件，并在当前系统中安装 Anaconda 软件。相关步骤简单明了，另外，在软件的安装过程中，还会显示相应的提示信息。具体安装步骤如下。

（1）单击第一个安装程序对话框中的 Next 按钮。

（2）在浏览了软件的相关条款和条件后，单击许可协议中的 I Agree 按钮。

（3）在选项中选择 Just Me 并单击 Next 按钮。

（4）选取默认的安装目标文件夹并单击 Next 按钮。

（5）随后将询问环境变量，以及是否需要将 Anaconda 注册为默认的 Python。选中后单击 Install 按钮。

（6）安装结束后，单击安装程序对话框中的 Finish 按钮。

1.2　Jupyter Notebook

Jupyter Notebook 是一个 Web 应用程序，可创建、共享文档。该文档中包含了实时代码、等式、可视化结果以及解释性文本内容，其用途包括数据清理和转换、数值模拟、统计建模、机器学习等。Jupyter Notebook 类似于一个画布（Canvas）或环境，可使用编程语言（在当前示例中是 Python）执行计算并以非常方便的方式显示结果。

如果读者正在进行某种分析工作，那么 Jupyter Notebook 是非常方便的——其间通常会包含解释性文本、产生结果的代码和可视化结果，这些都显示在 Jupyter Notebook 中。据此，使用任何编程语言，特别是 Python，它都是一种非常方便的分析工作方法。Jupyter 项目诞生于 2014 年 IPython 项目。现在，它已经发展到支持多种编程语言的交互式数据科学和科学计算工具，因此可以将 Jupyter Notebook 与许多其他编程语言一起使用（多达 20 种语言）。Jupyter 这一名称来自 Julia、Python 和 R，这也是 Jupyter Notebook 最初支持的 3 种编程语言。

1.2.1　创建自己的 Jupyter Notebook

当启动 Anaconda 并开启 Jupyter Notebook 时，可从安装程序列表中单击 Anaconda Prompt。Anaconda Prompt 可视为一个终端（Terminal），用户可在其中输入相关命令。下面首先在桌面生成一个名为 PythonDataScience 的文件夹，该目录将存储在 Jupyter Notebook 中为本书编写和运行的所有 Python 代码。

一旦打开终端，可输入 cd Desktop/PythonDataScience 命令并按 Enter 键访问 PythonDataScience。为了启用该目录中的 Jupyter Notebook 应用程序，可输入 jupyter notebook 命令并按 Enter 键。这将启动当前应用程序，并可在浏览器的选项卡中看到该应用程序的主界面，如图 1.2 所示。

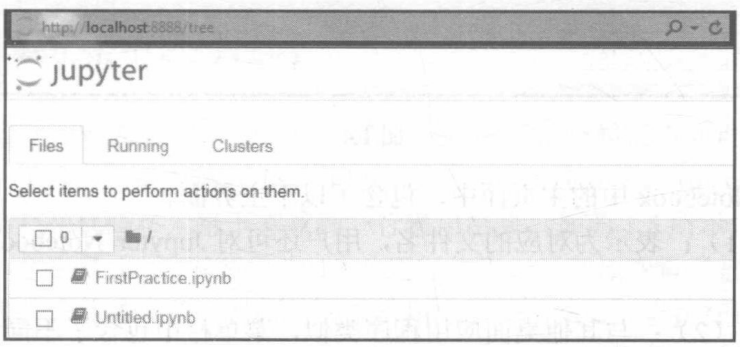

图 1.2

图 1.2 中包含了 3 个选项卡。在 Files 选项卡中，可以看到文件夹内包含的所有文件；在 Running 选项卡中，可以看到处于运行状态的程序，如 Terminal 或 Notebook；在 Clusters 选项卡中，则显示了与并行计算相关的细节内容，但本书将不会涉及这一特性。

Files 选项卡则是本书所用的主选项卡。当创建新的 Jupyter Notebook 时，可执行 New | Python 3 Notebook 命令，如图 1.3 所示。

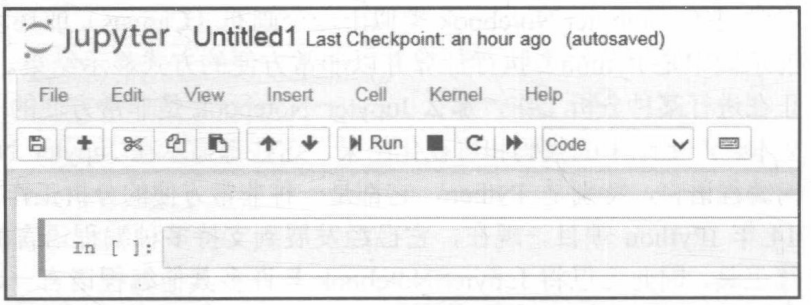

图 1.3

这将打开新的文件，即开始编码并运行 Python 代码的 Jupyter Notebook。

1.2.2 Jupyter Notebook 用户界面

Jupyter Notebook 包含了一些较为有用的界面，并可在操作过程中显示一些重要的信息和提示。此处访问 Help 命令，单击 User Interface Tour，并快速浏览一下 Jupyter Notebook 中的界面，如图 1.4 所示。

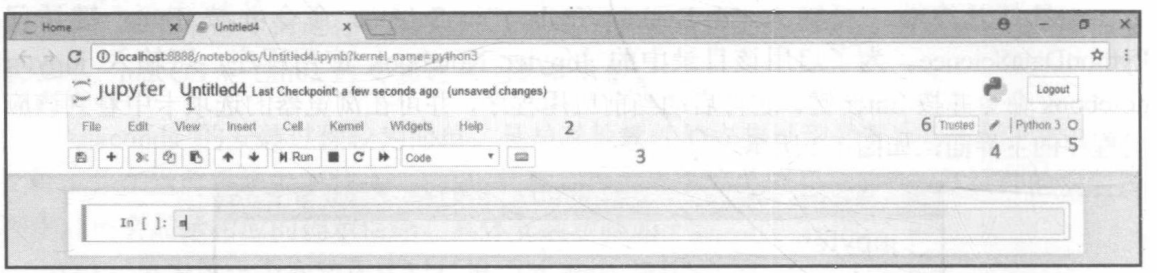

图 1.4

在 Jupyter Notebook 中的主页面中，包含了以下主界面。

- 标题（1）：表示为对应的文件名，用户还可对 Jupyter Notebook 的文件名进行修改。
- 菜单栏（2）：与其他桌面应用程序类似，菜单栏中包含了不同的操作。
- 工具栏（3）：位于菜单栏下方，其中包含了一些小图标，进而执行某些常见的

操作，如保存文件、分割单元、粘贴单元、移动单元等。

- 模式指示器（4）：位于菜单栏的右侧。Jupyter Notebook 包含了两种模式，即 Edit 模式和 Command 模式。其中，Command 模式中涵盖了许多可用的键盘快捷操作。在该模式中，指示器区域并不会显示任何图标，且需要对文件自身进行操作，如保存文件、复制和粘贴单元等。Edit 模式则允许用户在某个单元中编写代码或文本。当采用 Edit 模式时，将会在指示器区域看到一个铅笔图标。

注意：

Jupyter Notebook 由两种单元类型构成，即代码单元和文本单元。当在 Edit 模型下，所选单元的边框呈现为绿色。当从 Edit 模式返回 Command 模式时，可按 Esc 键或 Ctrl+M 快捷键。此外，还存在多种可用的键盘快捷方式，读者可访问 Help 命令查看快捷操作列表。

- 内核指示器（5）：显示系统的计算进程的状态。当中断进程中的计算时，可使用工具栏中的 stop 按钮。
- 消息区域（6）：该区域将显示相关消息，如 saving the file、interrupting the kernel 等。在消息区域中，用户可看到所执行的操作。

1.3　使用 Jupyter Notebook

下面打开新的 Jupyter Notebook，生成新的 Python 3 Jupyter Notebook，并将其命名为 FirstPractice。如前所述，Jupyter Notebook 由单元构成，其中包含了两种单元类型，即代码单元和文本单元。每次打开 Jupyter Notebook 时，将会显示代码单元，用户可于其中执行任何 Python 语句。

1.3.1　在代码单元格中运行代码

本节将尝试运行一些简单的代码语句，并学习如何在代码和文本间调整单元类型。对此，可在第一个代码单元格中输入 1+1 并执行该代码。当通过单击 run cell 按钮运行单元格中代码时，将会在代码单元格下方看到一行输出结果，如图 1.5 所示。

接下来生成一个变量 a，将其赋值为 10 并运行代码。虽然该变量已被创建，但考虑到尚未编写任何代码计算该变量，因而无法看到相应的输出结果。当执行这一条语句时，如果使用到该变量，例如，将该变量加 1，运行代码后将会看到如图 1.6 所示的结果。

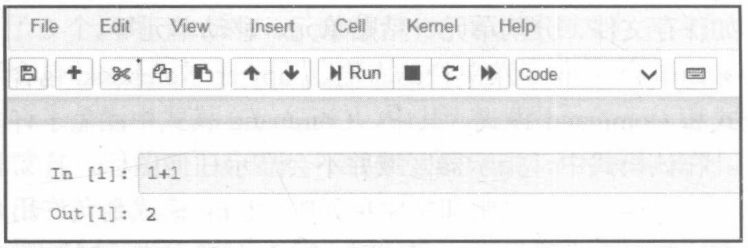

图 1.5

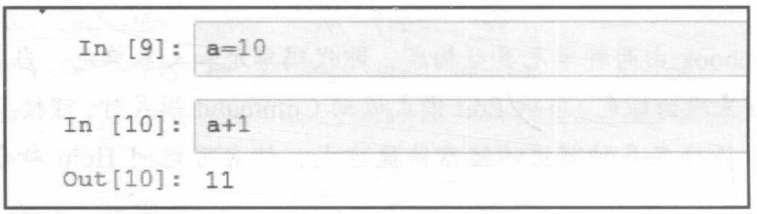

图 1.6

下面考查基于变量 i 的 for 语法示例，如图 1.7 所示。

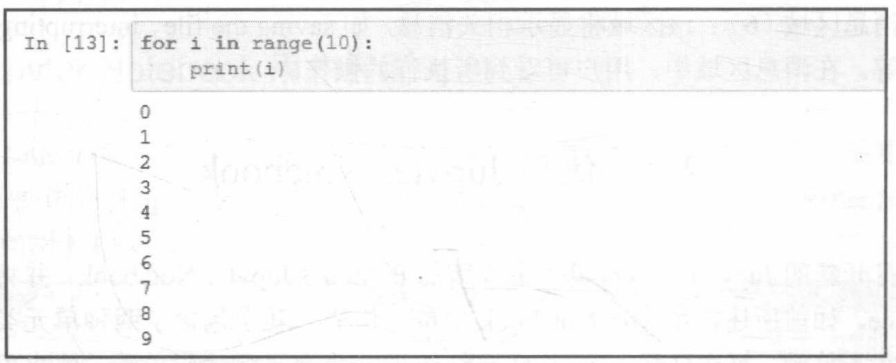

图 1.7

当对应值位于 range(10)时，上述代码将通知 Jupyter Notebook 输出 i 值，对应结果如图 1.7 所示。

1.3.2　在文本单元格中运行 markdown 语法

之前曾谈到，单元格的默认类型是代码单元格，并可于其中编写 Python 表达式。除此之外，另一种类型则是文本单元格，并可用于编写文本内容，在当前输出结果下方的单元格中，可尝试输入 This is regular text。当执行 Cell | Cell Type | Markdown 命令时，即可通知 Jupyter Notebook，当前内容并非 Python 代码，而是文本内容。运行代码后，将会