Preface

序言

What Is Data Science?

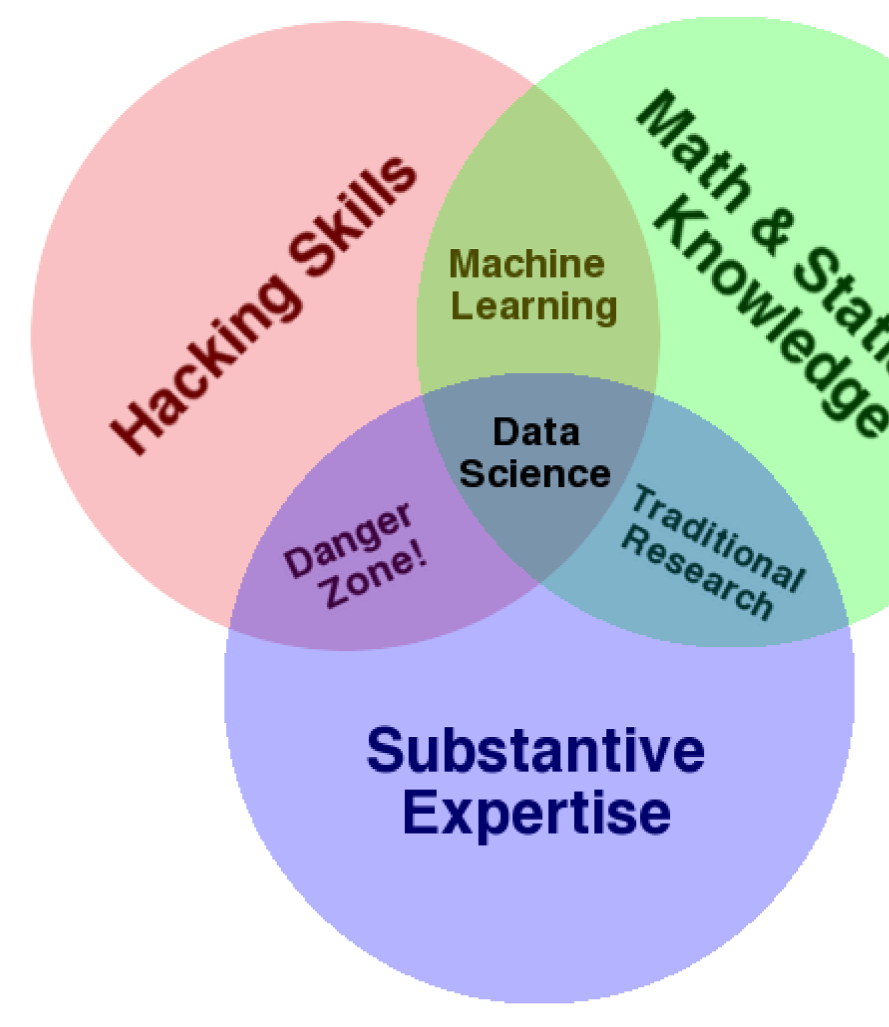
什么是数据科学?

> This is a book about doing data science with Python, which immediately begs the question: what is *data science*? It's a surprisingly hard definition to nail down, especially given how ubiquitous the term has become. Vocal critics have variously dismissed the term as a superfluous label (after all, what science doesn't involve data?) or a simple buzzword that only exists to salt resumes and catch the eye of overzealous tech recruiters.

这是一本介绍使用Python完成数据科学工作的书，那么立刻就会带来一个问题：什么是*数据科学*？这是一个十分难以定义的概念，尤其是这几年这个术语几乎随处可见。批评的声音认为这是一个多余的标签（毕竟，哪样科学不包含数据呢？）或者这只是一个为了博取关注而产生的流行词汇。

> In my mind, these critiques miss something important. Data science, despite its hype-laden veneer, is perhaps the best label we have for the cross-disciplinary set of skills that are becoming increasingly important in many applications across industry and academia. This cross-disciplinary piece is key: in my mind, the best extisting definition of data science is illustrated by Drew Conway's Data Science Venn Diagram, first published on his blog in September 2010:

在作者看来，这些批评忽略了一些重要的东西。数据科学，除了部分炒作的成分外，可能是目前我们能够找到的最合适的词汇来表达这种跨学科领域的技术了，特别是越来越多的工业和学术应用都在使用它。这里的关键是跨学科领域：作者认为，最好表达数据科学的定义的方式是2010年9月Drew Conway在他的博客里面发表的下面这张图：



(Source: [Drew Conway](http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram). Used by permission.)

(来源: [Drew Conway](http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram)。授权使用。)

> While some of the intersection labels are a bit tongue-in-cheek, this diagram captures the essence of what I think people mean when they say "data science": it is fundamentally an *interdisciplinary* subject. Data science comprises three distinct and overlapping areas: the skills of a *statistician* who knows how to model and summarize datasets (which are growing ever larger); the skills of a *computer scientist* who can design and use algorithms to efficiently store, process, and visualize this data; and the *domain expertise*—what we might think of as "classical" training in a subject—necessary both to formulate the right questions and to put their answers in context.

虽然图中，圆形重叠部分的标签看起来很有些嘲讽的意味，但这张图把握了当人们使用“数据科学”这个术语时候的精髓：最根本来说，数据科学是一门交叉学科。数据科学有三个领域交叉而成：需要*统计学家*来对数据集（正在变得越来越巨大）进行建模和统计；需要*计算机科学家*来使用算法有效地存储、处理和展现这些数据；还需要*领域专家*（通常在传统意义上我们就是这么做的）来在相关垂直领域整理出正确的问题和相应的解决方法。

> With this in mind, I would encourage you to think of data science not as a new domain of knowledge to learn, but a new set of skills that you can apply within your current area of expertise. Whether you are reporting election results, forecasting stock returns, optimizing online ad clicks, identifying microorganisms in microscope photos, seeking new classes of astronomical objects, or working with data in any other field, the goal of this book is to give you the ability to ask and answer new questions about your chosen subject area.

根据上述解释，读者与其将数据科学当成是一个新的知识领域来学习，还不如将你已有的专业知识融会贯通，发展出新的数据科学技巧。无论你是在统计选举结果、预测股市回报、优化在线广告点击、在显微镜图像中识别微小组织、寻找一类新的天文物体、或者是其他任何与数据相关的工作，本书的目标就是为你提供一种新的能力来提出和解答该领域的相关问题。

Who Is This Book For?

谁适合读这本书?

> In my teaching both at the University of Washington and at various tech-focused conferences and meetups, one of the most common questions I have heard is this: "how should I learn Python?" The people asking are generally technically minded students, developers, or researchers, often with an already strong background in writing code and using computational and numerical tools. Most of these folks don't want to learn Python *per se*, but want to learn the language with the aim of using it as a tool for data-intensive and computational science. While a large patchwork of videos, blog posts, and tutorials for this audience is available online, I've long been frustrated by the lack of a single good answer to this question; that is what inspired this book.

在作者华盛顿大学教学经历和在其他论坛会议演讲的过程中，最多被问到的问题之一就是：“我应该怎样学习Python？”提问者包括想在技术上深造的学生、开发人员或者研究人员，而且他们往往已经具备很强大的代码编写和使用数值计算工具的背景。他们其实并不是渴望学习Python语言*本身*，而只是想要学习Python语言有关数据方面或科学数值计算方面的内容。 网络上已经有数之不尽的视频、博客和教程，很难找到一个关于这个问题的唯一答案。是什么促使作者写这本书。

> The book is not meant to be an introduction to Python or to programming in general; I assume the reader has familiarity with the Python language, including defining functions, assigning variables, calling methods of objects, controlling the flow of a program, and other basic tasks.NumPy, Pandas, Matplotlib, Scikit-Learn, and related tools–to effectively store, manipulate, and gain insight from data.

本书不会作为Python语言的通用介绍；作者假定读者对于Python语言已经比较熟悉，包括函数定义，变量赋值，对象方法调用，程序流程控制，和其他基本的任何。使用Numpy、Pandas、Matplotlib、Scikit-Learn和相关的工具来存储、处理和展示数据。

Why Python?

为什么要用**Python**?

> Python has emerged over the last couple decades as a first-class tool for scientific computing tasks, including the analysis and visualization of large datasets. This may have come as a surprise to early proponents of the Python language: the language itself was not specifically designed with data analysis or scientific computing in mind. The usefulness of Python for data science stems primarily from the large and active ecosystem of third-party packages: *NumPy* for manipulation of homogeneous array-based data, *Pandas* for manipulation of heterogeneous and labeled data, *SciPy* for common scientific computing tasks, *Matplotlib* for publication-quality visualizations, *IPython* for interactive execution and sharing of code, *Scikit-Learn* for machine learning, and many more tools that will be mentioned in the following pages.

Python在最近20年已经发展成为科学计算包括分析和展示大型数据集的最流行工具。这对于Python语言的早期支持者来说是一个惊喜：因为这门语言本身并不是专门为了数据分析和科学计算来设计的。 Python在数据科学中的蓬勃发展主要来源于其大量活跃的第三方包：*Numpy*用于处理同类的数组结构数据；*Pandas*用于处理不同种类和标签化的数据；*SciPy*用于通用的科学运算任务；*Matplotlib*用于可打印标准的图表展示；*IPython*用于交互式执行和共享代码；*Scikit-Learn*用于机器学习，这些工具将在后续的章节中介绍。

> If you are looking for a guide to the Python language itself, I would suggest the sister project to this book, "A Whirlwind Tour of the Python Language". This short report provides a tour of the essential features of the Python language, aimed at data scientists who already are familiar with one or more other programming languages.

如果你需要的是Python语言本身的指引，作者推荐本项目的兄弟项目"A Whirlwind Tour of the Python Language"（译者注：Python旋风之旅中文版已经全部翻译完成）。这个项目提供了Python语言最基本特性的一个简单介绍，针对已经掌握了一门或更多其他编程语言数据科学家。

Python 2 vs Python 3

Python 2 还是 3

> This book uses the syntax of Python 3, which contains language enhancements that are not compatible with the 2.x series of Python. Though Python 3.0 was first released in 2008, adoption has been relatively slow, particularly in the scientific and web development communities. This is primarily because it took some time for many of the essential third-party packages and toolkits to be made compatible with the new language internals. Since early 2014, however, stable releases of the most important tools in the data science ecosystem have been fully compatible with both Python 2 and 3, and so this book will use the newer Python 3 syntax. However, the vast majority of code snippets in this book will also work without modification in Python 2: in cases where a Py2-incompatible syntax is used, I will make every effort to note it explicitly.

本书采用Python 3的语法编写，内含一些2.x版本不具备的语言增强特性。虽然Python 3.0在2008年就已经发布，但是转换并不迅速，尤其在科学和Web开发社区中。这主要是因为很多核心的第三方库和工具需要时间才能兼容新的语言版本特性。自2014年初起，大多数重要的数据科学生态工具都已经发布了兼容Python 2和3的稳定版本，因此本书采用新的Python 3语法。然而，本书很大一部分代码片段都能不需修改地运行在Python 2环境：在使用了Py2不兼容的语法的地方，作者会尽力标明。

译者注：Python 2将于2020年1月1日停止维护，因此强烈建议读者不要继续使用Python 2环境编写代码。

Outline of the Book

大纲

> Each chapter of this book focuses on a particular package or tool that contributes a fundamental piece of the Python Data Sciece story.
>
> 1. IPython and Jupyter: these packages provide the computational environment in which many Python-using data scientists work.
> 2. NumPy: this library provides the `ndarray` for efficient storage and manipulation of dense data arrays in Python.
> 3. Pandas: this library provides the `DataFrame` for efficient storage and manipulation of labeled/columnar data in Python.
> 4. Matplotlib: this library provides capabilities for a flexible range of data visualizations in Python.
> 5. Scikit-Learn: this library provides efficient & clean Python implementations of the most important and established machine learning algorithms.

本书的每一章都聚焦于一个特定的包或工具，它对数据科学某个方面都有重要的应用和帮助。

1. IPython 和 Jupyter: 这两个包提供了使用Python的数据科学家最喜爱的计算环境。
2. NumPy: 这个包提供了 `ndarray` 对象用于有效的存储和处理数组中的非稀疏数据。
3. Pandas: 这个包提供了 `DataFrame` 对象用于有效的存储和处理标签化的基于列结构的数据。
4. Matplotlib: 这个包提供了最灵活的数据图表展示功能。
5. Scikit-Learn: 这个包提供了很多重要的机器学习算法以及有效和简洁的Python实现。

> The PyData world is certainly much larger than these five packages, and is growing every day. With this in mind, I make every attempt through these pages to provide references to other interesting efforts, projects, and packages that are pushing the boundaries of what can be done in Python. Nevertheless, these five are currently fundamental to much of the work being done in the Python data science space, and I expect they will remain important even as the ecosystem continues growing around them.

Python的数据科学领域肯定远远不止这5个包，而且每天都在不断增长。作者会在每个章节都尽量提供其他有趣的项目和包的推荐。无论如何，这五个包目前是Python数据科学领域最基础的内容，作者期待他们会在未来依然保持其重要性，甚至在生态持续发展的情况下。

Using Code Examples

使用代码例子

> Supplemental material (code examples, figures, etc.) is available for download at http://github.com/jakevdp/PythonDataScienceHandbook/. This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

本书附带的资源（代码示例，图表等）可以在 http://github.com/wangyingsm/Python-Data-Science-Handbook/ 下载。本书的代码例子是为了帮助你理解内容。在通常意义下，本书附带的代码可以被使用在你的程序和文档中。你不需要联系作者获得授权，除非你在修改或重构代码非常重要的部分。例如，使用本书的代码编写你的程序不需要获得作者授权；销售和分发本书的代码不需要获得作者的授权；引用本书或书中的代码例子回答问题不需要获得作者的授权。将本书大部分的代码例子组织在你产品的文档中确实需要获得作者的授权。

> We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example:
>
> > *The Python Data Science Handbook* by Jake VanderPlas (O'Reilly). Copyright 2016 Jake VanderPlas, 978-1-491-91205-8.

虽然不是必须的，但是如果你在引用时声明了标题、作者、出版社和ISBN的话，作者表示感激。

> If you feel your use of code examples falls outside fair use or the per‐ mission given above, feel free to contact us at permissions@oreilly.com.

如果你认为你对于代码例子的使用超出了上述的授权范围，请联系 permissions@oreilly.com。

Installation Considerations

安装

> Installing Python and the suite of libraries that enable scientific computing is straightforward . This section will outline some of the considerations when setting up your computer.

安装Python和科学计算库的套件是很直接的。本节简单介绍一下配置你的计算机的方法。

> Though there are various ways to install Python, the one I would suggest for use in data science is the Anaconda distribution, which works similarly whether you use Windows, Linux, or Mac OS X. The Anaconda distribution comes in two flavors:
>
> - Miniconda gives you the Python interpreter itself, along with a command-line tool called `conda` which operates as a cross-platform package manager geared toward Python packages, similar in spirit to the apt or yum tools that Linux users might be familiar with.
> - Anaconda includes both Python and conda, and additionally bundles a suite of other pre-installed packages geared toward scientific computing. Because of the size of this bundle, expect the installation to consume several gigabytes of disk space.

虽然有很多种方式安装Python，作者推荐使用Anaconda发行版安装，就像你的操作系统使用Windows、Linux或Mac OS X一样。Anaconda发行版有两种模式：

- Miniconda 带有Python解释器，还有一个命令行工具 `conda` 的包管理器，就像你在Linux操作系统发行版中常用的apt或yum工具一样。
- Anaconda 安装Python解释器和conda，还会预装科学计算套件，因为这个发行版包括了很多的第三方库，因此可能会占用你磁盘几G的空间。

> Any of the packages included with Anaconda can also be installed manually on top of Miniconda; for this reason I suggest starting with Miniconda.

任何包括在Anaconda发行版中的包都可以在Miniconda的基础上安装；因此，作者建议使用Miniconda。

译者注：如果磁盘空间不紧张，网络带宽也好的情况下，强烈建议使用Anaconda。

> To get started, download and install the Miniconda package–make sure to choose a version with Python 3–and then install the core packages used in this book:

```
[~]$ conda install numpy pandas scikit-learn matplotlib seaborn jupyter
```

在正式开始之前，下载和安装Miniconda，确认选择的是Python 3的版本，然后使用命令行安装本书需要用到的核心包：

```
$ conda install numpy pandas scikit-learn matplotlib seaborn jupyter
```

> Throughout the text, we will also make use of other more specialized tools in Python's scientific ecosystem; installation is usually as easy as typing **`conda install packagename`**. For more information on conda, including information about creating and using conda environments (which I would *highly* recommend), refer to conda's online documentation.

由上我们可知，我们可以使用conda命令安装Python生态中的任何工具，只需要简单的运行 **`conda install 包名称`** 即可。更多关于conda的信息，包括创建和使用conda环境（作者*强烈*推荐阅读），请参见conda在线文档。