

2

Install and Use Anaconda

安装使用 Anaconda

整套鸢尾花书都离不开 Anaconda 这个工具库



依在我看来，世间万物皆数学。

But in my opinion, all things in nature occur mathematically.

—— 勒内·笛卡尔 (René Descartes) | 法国哲学家、数学家、物理学家 | 1596 ~ 1650



XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX



2.1 集成开发环境

Python 有很多常用的 IDE (集成开发环境), 比如:

- ▶ JupyterLab: 基于 Web 的交互式开发环境, 支持多种编程语言, 包括 Python, 可以快速编写、测试和共享代码, 非常适合数据科学和机器学习领域。作者认为, JupyterLab 和 Jupyter Notebook 非常适合大家做探究式学习。目前, 《编程不难》、《可视之美》两册的配套的代码多是 Jupyter 笔记。这个话题后文将详细介绍如何使用 JupyterLab。
- ▶ Spyder: 基于 Qt 开发的 Python IDE, 提供了一个集成的开发环境, 包括编辑器、调试器和控制台, 非常适合科学计算和数据分析。虽然“鸢尾花书”剩余几册的代码都是在 Spyder 中完成, 建议初学者还是在 JupyterLab 中分段运行代码。对于 MATLAB 转 Python 的读者来说, Spyder 可能是最容易上手的 IDE。在所有的 Python IDE 中, Spyder 最像 MATLAB。
- ▶ PyCharm: JetBrains 公司开发的跨平台 Python IDE, 提供了许多功能, 包括代码智能提示、代码自动完成、调试和单元测试等。建议有 Python 开发经验的读者使用 PyCharm 运行本书代码。



什么是集成开发环境?

集成开发环境 (Integrated Development Environment, 简称 IDE) 是一种用于软件开发的工具。它通常包括一个代码编辑器、一个调试器和一个构建工具, 以及其他功能, 例如自动补全、语法高亮、代码重构等。IDE 的目的是提供一个集成的工作环境, 使开发人员能够更高效地编写、调试和测试代码。使用 IDE 可以极大地提高开发效率。例如, 它可以帮助开发人员在编写代码时自动补全函数名称、参数等, 减少打错代码的风险; 它可以提供一些调试工具来检测和修复代码中的错误, 使得开发人员更容易发现问题; 它可以通过自动构建工具来编译和构建代码, 减少手动操作的繁琐过程。总之, IDE 是一种开发人员必备的工具, 可以让开发人员更加专注于编写高质量的代码。

表 1. 比较三个常用的 IDE

维度	JupyterLab	Spyder	PyCharm
适用场景	数据科学、机器学习、交互式	科学计算、数据分析	通用编程、开发
编辑器	基于 Web 的文本编辑器	Qt 构建的文本编辑器	IntelliJ IDEA 编辑器
调试器	内置的交互式调试器	内置的调试器	内置的调试器
插件支持	丰富的插件生态系统	插件支持较少	丰富的插件生态系统
社区支持	由 Jupyter 项目支持	由 Spyder 社区支持	由 JetBrains 公司支持
扩展性	支持自定义和扩展	可以自定义外观和行为	支持自定义和扩展
学习曲线	平缓	友好	稍微陡峭
收费与否	免费	免费	有免费和付费版本
平台支持	支持 Windows、Mac 和 Linux	支持 Windows、Mac 和 Linux	支持 Windows、Mac 和 Linux

Anaconda 可谓“科学计算全家桶”, 包含科学计算领域可能用到的大部分 Python 工具, 包括 Python 解释器、常用的第三方库、包管理器、IDE 等。前文提到的 JupyterLab、Spyder、PyCharm 这三个 IDE 都在 Anaconda 中。



什么是 Anaconda?

Anaconda 是一个流行的 Python 发行版, 由 Anaconda, Inc. 开发和维护, 旨在为数据科学、机器学习和科学计算提供一个全面的工具包。Anaconda 集成了许多常用的 Python 库和工具, 如 NumPy、SciPy、Pandas、Matplotlib、Scikit-learn、Jupyter Notebook 等。它还包括一个名为 conda 的软件包管理器, 可以帮助用户安装、更新和管理 Python 库和依赖项。Anaconda 还提供了一个名为 Anaconda Navigator 的图形用户界面, 用户可以通过这个界面轻松地管理他们的 Python 环境、安装和卸载库、启动 Jupyter

Notebook 等操作。除了 Python 环境和库之外，Anaconda 还包括许多其他工具和应用程序，如 Spyder、PyCharm、VS Code、R 语言环境等等，使得它成为数据科学家和研究人员的首选工具之一。Anaconda 可以安装在多个平台上，包括 Windows、Linux 和 Mac OS X。

2.2 如何安装 Anaconda?

安装 Anaconda

下文手把手教大家如何在 Windows 上安装、测试 Anaconda，有经验的读者可以跳过。

对于 Mac 用户，大家可以参考如下链接安装 Anaconda：

<https://docs.anaconda.com/anaconda/install/mac-os/>

要是想特别安装某个版本的 Python，请参考：

https://pythonhowto.readthedocs.io/zh_CN/latest/install.html

注意，Anaconda 安装后大概占用 5G 空间。有 Python 开发经验的读者，可以根据需求自行分别安装 JupyterLab、Spyder、PyCharm。

在 Windows 上安装 Anaconda 可以按照以下步骤进行：

a) 下载。在 Anaconda 官网 (<https://www.anaconda.com/>) 下载适合大家操作系统的 Anaconda 版本，选择对应的 Python 版本（一般建议选择最新版 Python3.x），并下载对应的安装程序。注意，Anaconda 不断推出新版本，大家下载的版本号肯定和下图的版本号不同。建议大家从官网下载最新版本安装程序。



图 1. 安装程序图标

b) 运行安装程序：下载完毕后，双击下载文件运行安装程序。在安装程序打开后，点击“Next”进入下一步。

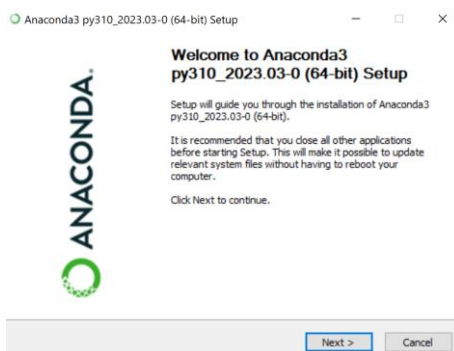


图 2. 运行安装程序

c) 阅读协议：阅读协议并同意“I Agree”，然后点击“Next”。

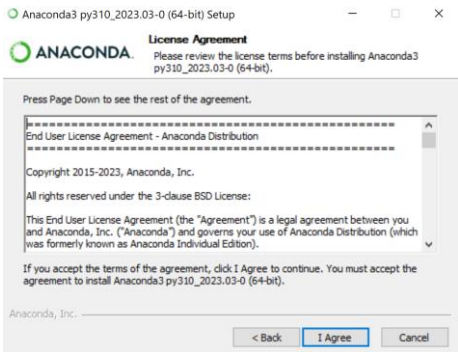


图 3. 阅读协议

d) 安装类型：推荐默认“Just Me”；对于多用户 PC，可以选择“All Users”；然后点击“Next”。

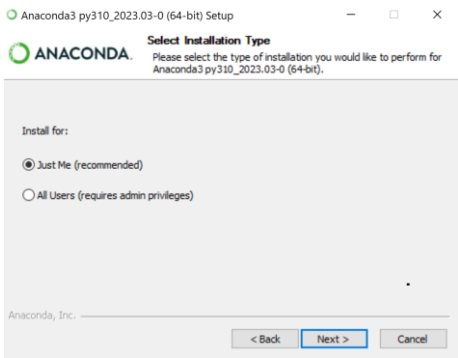


图 4. 安装类型

e) 安装路径：可以指定 Anaconda 的安装路径 (建议零基础读者选择默认路径)，然后点击“Next”。

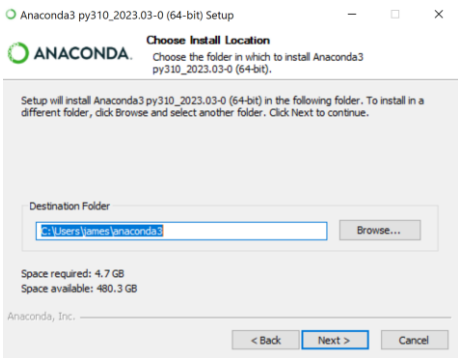


图 5. 安装路径

f) 配置环境变量：选择是否将 Anaconda 添加到系统环境变量中，建议勾选该选项，这样就可以在命令行中使用 Anaconda 的工具了。然后点击“Install”进行安装。

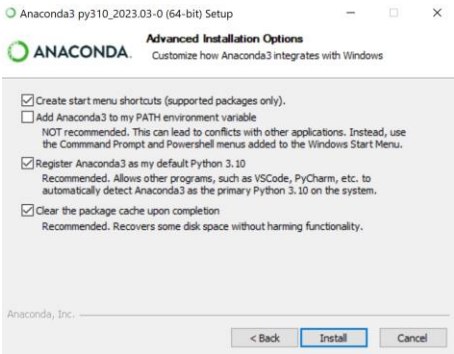


图 6. 安装选择

g) 等待安装完成：安装过程可能持续 10 分钟左右。等待安装完成后，会弹出“Installation Complete”对话框，点击“Next”。如果这步持续时间过长 (超过一小时)，建议强制停止安装，删除安装包。关机再开机，重新下载安装包从头开始再尝试安装。

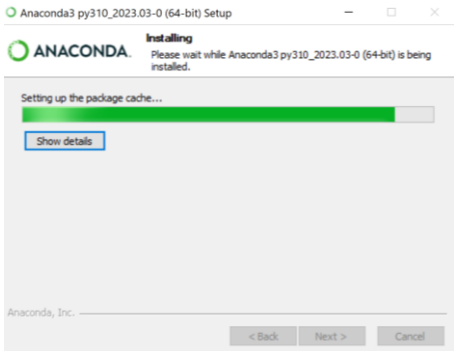


图 7. 等待安装完成

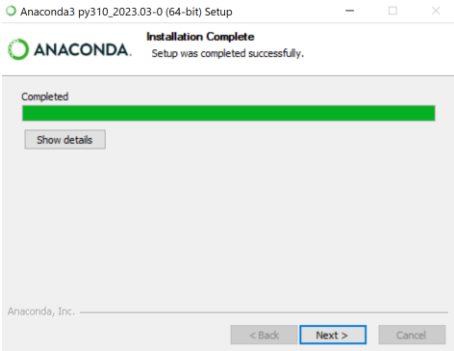


图 8. 安装完成

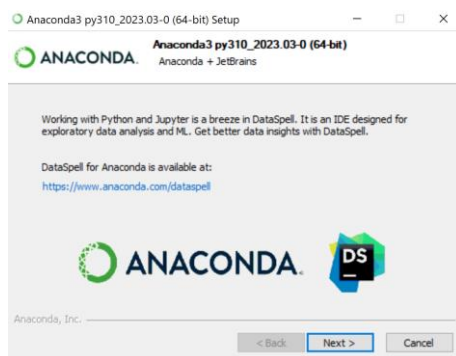


图 9. 广告时间，点 Next

h) 完成安装：点击“Finish”完成 Anaconda 的安装。之后会跳出两个网页，不需要理会，关闭即可。

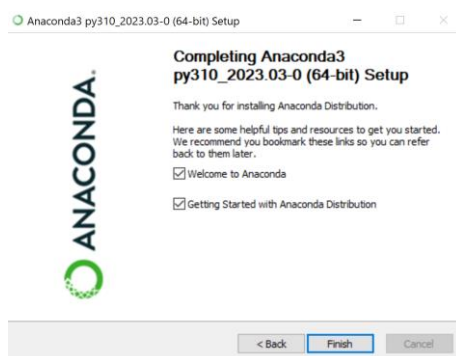


图 10. 确认完成

安装完成后，可以在“开始菜单”中找到 Anaconda 的安装目录，并启动“Anaconda Navigator”来使用 Anaconda 的工具和功能。同时，也可以在命令行中使用 Anaconda 的工具和命令，例如使用“conda”命令来管理 Python 的虚拟环境和安装依赖包等。

2.3 测试 JupyterLab

要打开并测试 JupyterLab，可以按照以下步骤进行：

a) 找到并打开 Anaconda Navigator (需要 1 分钟左右，稍安勿躁)，点击 JupyterLab 对应的 Launch。马上一个网页将会跳出来，建议大家默认使用 Chrome 浏览器，Firefox 或 Edge 也都可以。

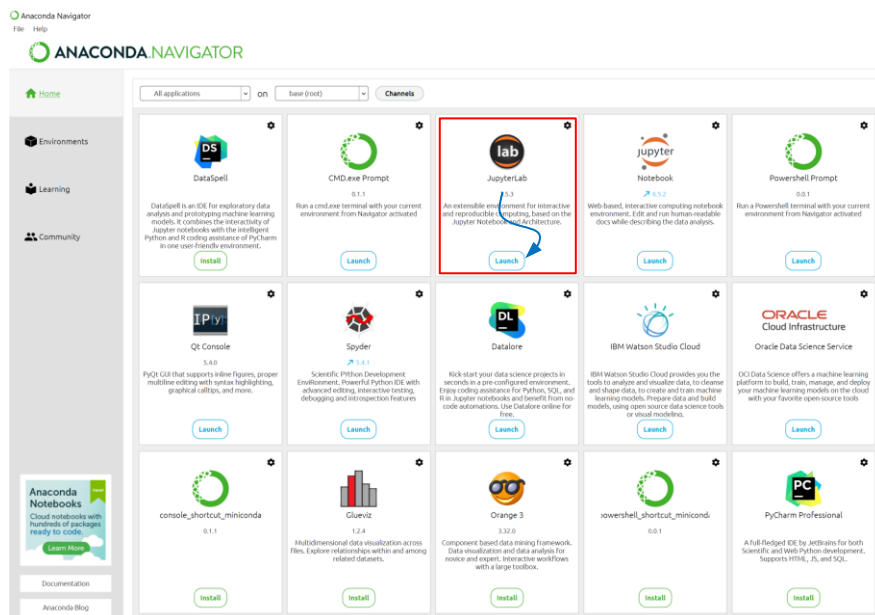


图 11. Anaconda Navigator 界面

b) 进入 JupyterLab 界面，点击 Notebook (Python 3)，创建 Jupyter Notebook。

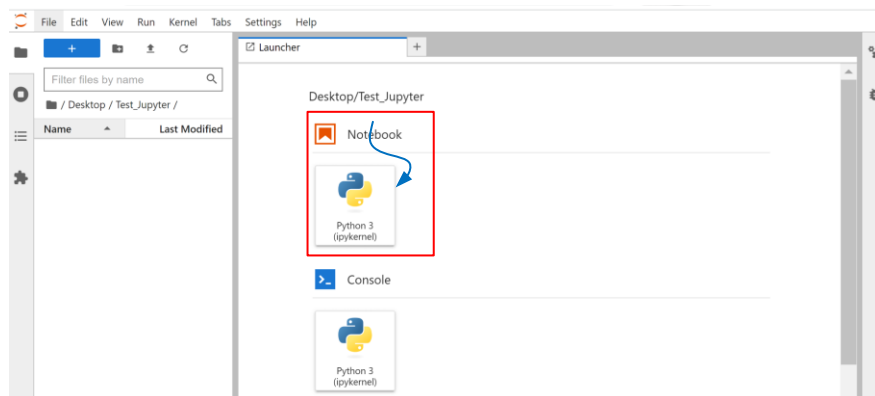


图 12. JupyterLab 界面

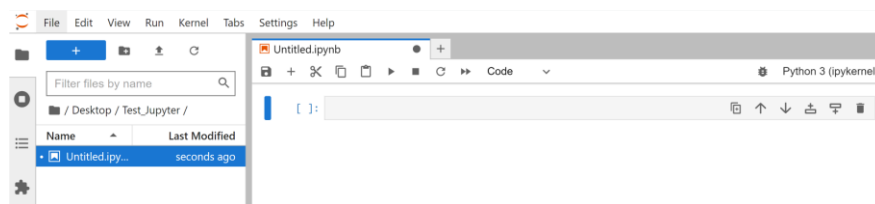


图 13. 创建 Jupyter Notebook

c) 在下面窗口中输入， $1 + 2$ ，然后点击“Ctrl + Enter”快捷键，运行并得到 3 这个结果。大家也可以尝试“Shift + Enter”快捷键，运行代码同时生成新区域，大家自己可以先玩一会。下一节将专门讲解如何使用 JupyterLab。

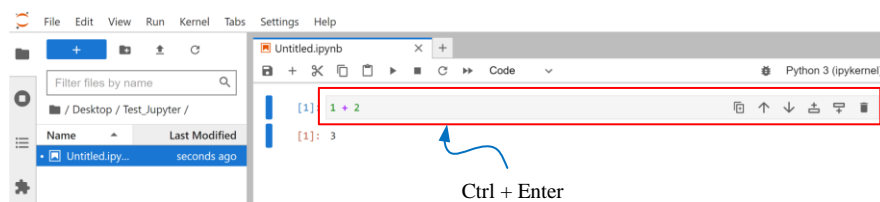


图 14. 运算

2.4 查看 Python 库版本号

在安装 Anaconda 时，各种常用的 Python 工具已经顺道安装完成。有些时候，我们需要查看 Python 各种库的版本号，下面介绍几种方法。

大家可以进入 Anaconda.Navigtor，点击 Environments，如果有不同环境的话，选择特定的环境，在右侧可以看到已经安装库的版本号。

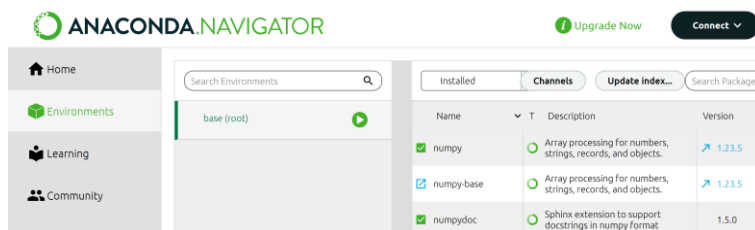


图 15. 在 Anaconda.Navigtor 查看 Python 库版本号

在电脑中搜索 Anaconda Prompt，然后键入 `conda list`，也可以调取所有安装 Python 库的版本号。

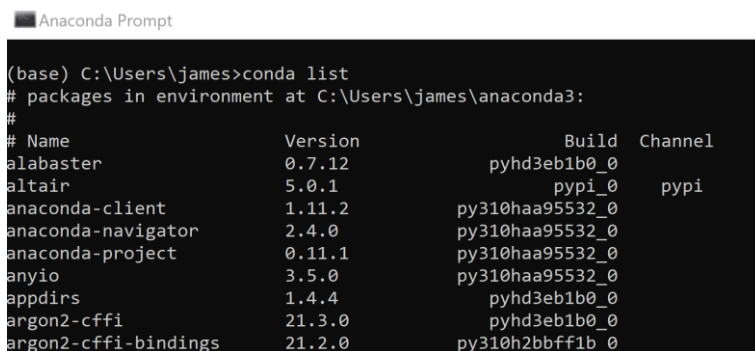


图 16. 在 Anaconda Prompt 查看 Python 库版本号

此外，在图 15 中左键点击绿色三角，左键单击 Open Terminal 也可以打开 Anaconda Prompt。

在 Anaconda Prompt，我们可以用 `pip show` 库名，比如 `pip show numpy`，来调取某个特定 Python 库信息。


```
(base) C:\Users\james>pip show numpy
Name: numpy
Version: 1.23.5
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD
Location: c:\users\james\anaconda3\lib\site-packages
Requires:
Required-by: altair, astropy, basemap, bokeh, Bottleneck, Cartopy, contourpy, daal4py, datashader, datashape, gensim, gl
zeh, h5py, holoviews, hvplot, imagecodecs, imageio, imbalanced-learn, matplotlib, mkl-fft, mkl-random, moviepy, numba, n
umexpr, pandas, patsy, pyarrow, pydeck, pyerfa, PyWavelets, scikit-image, scikit-learn, scipy, seaborn, shapely, statsmo
dels, streamlit, tables, tifffile, transformers, xarray
```

图 17. 在 Anaconda Prompt 查看某个特定 Python 库版本号

在 JupyterLab 中，我们也可以用 `!pip list` 查看已经安装的所有 Python 库版本号。在 JupyterLab 中，感叹号 (exclamation mark) `!` 用于执行操作系统命令或外部程序。比如，`!dir` 可以调取当前文件的目录。`!pip install` 包名，还可以用 `pip` 安装特定 Python 库。

```
[2]: !pip list
```

Package	Version
alabaster	0.7.12
altair	5.0.1
anaconda-client	1.11.2
anaconda-navigator	2.4.0
anaconda-project	0.11.1
anyio	3.5.0
appdirs	1.4.4

图 18. 在 JupyterLab 中查看 Python 库版本号

此外，对于特定 Python 包，我们还可以用如下语句查看其版本号。

```
[1]: import numpy
print(numpy.__version__)
1.23.5
```

图 19. 在 JupyterLab 中查看某个特定 Python 库版本号

在《编程不难》《可视之美》两本书中，大家会经常看到图 20 这种代码实例，书中会对代码中关键语句编号并讲解。为了

虽然这些代码都可以在配套代码文件中找到，但是依然强烈建议大家在 JupyterLab 中自己敲一遍。对于编程零基础读者，特别推荐大家逐行注释。

本书第 4 章专门介绍如何注释代码。

下面，我们就聊聊图 20 这段代码。

a 这句话就是注释。简单来说，代码中的注释是给人看的，机器对其视而不见。在 Python 中，`#` 符号用于创建单行注释。注释是用于解释代码的文本，它不会被 Python 解释器执行，因此不会影响程序的运行。

即便如此，编程时注释并不是可有可无的部分。我们可以使用注释来解释代码的目的、功能或特殊注意事项。毫不夸张地说，自己写完的代码，过不了一个月可能就会忘了某些具体语句或逻辑，而代码注释就完美解决这一问题。代码注释当然对于其他开发人员阅读和理解代码非常有帮助。

此外，在调试或测试代码时，我们也可以使用临时注释来暂时禁用或跳过某些代码行。

此外，在自定义函数时，我们也可以添加多行注释，来生成代码文档。本书第 8 章会专门介绍自定义函数。

b 导入 SciPy 库。SciPy 是一个用于科学计算和数据分析的开源 Python 库，它包含了许多用于数学、科学和工程计算的功能和工具。

本书第 26 章专门介绍 SciPy 库。

在 Python 中，import 是一个关键字，用于导入其他 Python 库/包/模块。

本书第 4 章将专门介绍如何使用 import。在 JupyterLab 中，只有成功导入某个库或模块后，才能调用其中函数。

注意，SciPy (S 和 P 大写) 是这个 Python 库的名字，而在 JupyterLab 中，导入这个库时，scipy 为全小写无空格。

c 中，print() 是 Python 的内置函数，用来打印，p 小写。

'scipy: %s' 是一个包含占位符的字符串，其中%s 是一个占位符，表示后面将被替换成一个字符串的值。在 Python 中，字符串 (string) 是一种数据类型，用于表示纯文本数据。

本书第 5 章将专门介绍包括字符串在内的常用数据类型。

scipy.__version__ 是 SciPy 库的一个属性，它包含了当前导入的 SciPy 版本的字符串。通过 scipy.__version__，我们可以获取计算机中当前 SciPy 库的版本信息。

d 在 Python 中用于导入 NumPy 库的语句。NumPy 是 Python 中用于科学计算和数值操作的一个强大的开源库。它提供了多维数组 (NumPy array) 和一系列用于操作这些数组的函数。NumPy 广泛用于数据分析、科学计算、机器学习等领域。

本书第 13 ~ 18 章介绍 NumPy 库常用工具。

e 导入 Matplotlib 库的语句。Matplotlib 是一个用于创建各种类型的图形和可视化的 Python 库。

本书第 10 ~ 12 章专门介绍常用可视化工具。

f 导入 Pandas 库。Pandas 是 Python 中用于数据分析和数据操作的高性能库。Pandas 提供了两种主要数据结构：Series 和 DataFrame，用于处理和操作各种类型的数据，包括表格数据、时间序列数据等等。

本书第 19 ~ 24 章介绍 Pandas 库常用工具。

g 导入 Statsmodels 库。Statsmodels 是一个 Python 库，用于执行统计分析和建立统计模型，包括线性回归、时间序列分析、假设检验和许多其他统计方法。

本书第 27 章专门介绍 Statsmodels 库。

h 导入 Scikit-Learn 库。Scikit-Learn，也称 sklearn，是一个强大的开源机器学习库，提供了用于各种机器学习任务的工具和算法。它包括分类、回归、聚类、降维、模型选择、模型评估等各种机器学习任务的实现。Scikit-Learn 还包括用于数据预处理和特征工程的功能。

本书第 28 ~ 33 章介绍 Scikit-Learn 库常用工具。

```

a # 检查常用Python库版本号
b import scipy
c print('scipy: %s' % scipy.__version__)

d import numpy
  print('numpy: %s' % numpy.__version__)

e import matplotlib
  print('matplotlib: %s' %
    matplotlib.__version__)

f import pandas
  print('pandas: %s' % pandas.__version__)

g import statsmodels
  print('statsmodels: %s' %
    statsmodels.__version__)

h import sklearn
  print('sklearn: %s' % sklearn.__version__)

```

图 20. 查看常用 Python 库版本号

2.5 安装、更新、卸载 Python 库

即便安装 Anaconda 时，各种常用 Python 库已经安装好；但是，在使用时，我们经常会安装其他库，亦或是更新已经安装的库。

以 pandas-datareader 为例，本书后续会利用 pandas-datareader 下载金融数据。在安装 Anaconda 时，这个库没有被安装，需要我们自行安装。

使用 pip 安装。pip 是 Python 的包管理器，它是最常用的安装库的方法。打开 Anaconda Prompt，然后运行 pip install pandas-datareader 命令来安装库。

```

Anaconda Prompt
(base) C:\Users\james>pip install pandas-datareader

```

图 21. 安装 pandas-datareader

如果使用的是 Anaconda Python 环境，有时也可以使用 conda 包管理器来安装库，比如 conda install library_name。具体采用 pip 还是 conda，建议大家在安装任何第三方库之前，首先查看这个库的技术文档，了解库的版本、更新情况、使用说明、常见案例。比如，pandas-datareader 的技术文档：

<https://pandas-datareader.readthedocs.io/en/latest/>

在这个网页首页，我们看到推荐 pip install pandas-datareader 安装 pandas-datareader。

如果大家有多个 Anaconda 环境，安装特定库时需要选择特定环境。如图 22 所示，当前 Anaconda 有两个环境，如果我们想在 demo 环境安装 Streamlit 的话，左键点击其绿色三角，在菜单中左键点击 Open Terminal，然后调出对应环境的 Anaconda Prompt。然后利用 pip install streamlit 安装 Streamlit。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

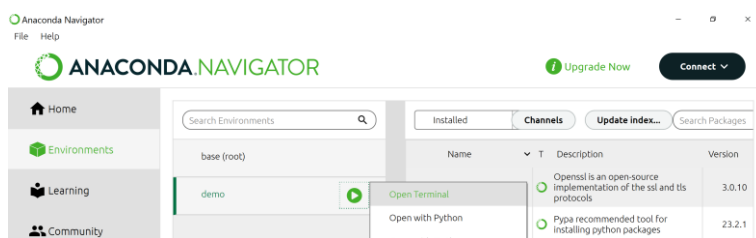


图 22. 在特定 Anaconda 环境安装 Python 库

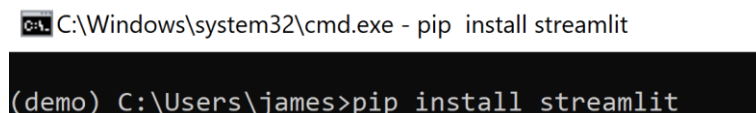


图 23. 安装 Streamlit

顺便提一嘴，在 Anaconda Navigator 中可以很轻松创建全新 Python 环境。如图 24 所示，大家只需要点击左下角加号 Create，在弹出的对话框中输入环境名称，大家还可以选择不同 Python 版本号。如果大家使用 R 语言的话，还可以创建 R 语言环境。

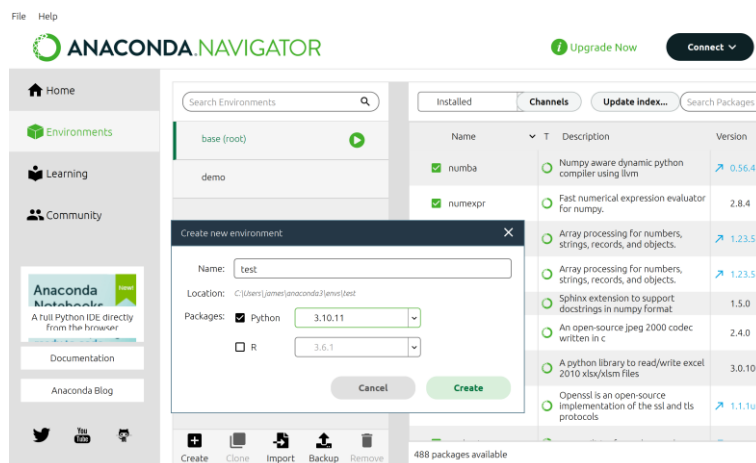


图 24. 在 Anaconda Navigator 中创建新环境

Streamlit 是一个用于创建 Web 应用程序的 Python 库，它可以让数据科学家、工程师和分析师轻松地将数据应用程序转化为交互式 Web 应用程序，无需深入的前端开发经验。我们将在本书最后两章利用 streamlit 搭建数学动画、数据分析、机器学习 App。请大家在安装 Streamlit 前，查看其技术文档：

<https://docs.streamlit.io/library/get-started/installation>

给大家一个任务，请大家首先安装 pandas-datareader，然后再安装 streamlit 库。

有时，我们也可以从库的源代码安装库，下载或克隆压缩文件，利用类似 `python setup.py install` 命令安装。这种方法不推荐初学者使用。

此外，我们也可以在 JupyterLab 中，用 `!pip install library_name` 方法安装特定库。这种方法也不推荐初学者使用。

想要卸载特定 Python 库也很容易，大家在 Anaconda Prompt 中键入 `pip uninstall library_name`。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

想要更新某个 Python 库，可以使用 `pip install library_name --upgrade`。如图 25 所示，大家也可以在 Anaconda Navigator 查看某个 Python 库是否有更新。如果出现蓝色箭头，这说明该库有新版本。

注意，由于 Python 库由不同第三方开发者开发、维护，更新库时要小心兼容性问题。这就是为什么我们有时需要不同 Anaconda 环境，以便控制不同库的版本。

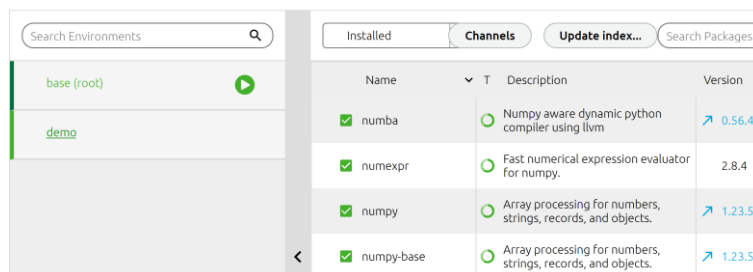


图 25. 在 Anaconda Navigator 中查看 Python 库是否有更新

库的健康情况

Python 第三方库都是由社区开发者开发、维护，在使用一些生僻的 Python 库之前，建议大家了解一下这个库的健康情况。

大家可以查看库在 GitHub 或其他代码托管平台上的维护更新情况，比如提最近提交日期、版本历史、日志更新，以及提交频率等信息。此外，大家也可以看看 GitHub 上库的安装使用、标星 (star)、问题 (issue) 等是否活跃。某个 Python 库的技术文档质量、更新情况也可以作为衡量其健康程度的指标。

此外，最简单的办法就是通过 Snyk Advisor 打分来评估 Python 库的健康情况：

<https://snyk.io/advisor/python/scoring>

图 26 所示为 Streamlit 库在 2023 年 9 月份的评分。一般来说，评分在 85 分左右的 Python 库可以一试。评分如果在 95 分上下，说明 Python 库的健康程度很好。

吐槽一下，pandas-datareader 这个 Python 库的维护就很差，2023 年 9 月份在 Snyk Advisor 评分仅为 62 分，刚及格。也就是说这个库凑合能用，但是出现 bug 后果自负。在撰写本书时，作者还能用 pandas-datareader 从 FRED (Federal Reserve Economic Data) 下载金融数据。为了避免 pandas-datareader 失效，作者对本书中用到的金融数据都做了备份，大家可以在本书配套代码中找到。万一下载失败，可以用 `pandas.read_csv()` 函数导入 CSV 数据。很期待开发者能尽快更新库，并解决 Yahoo 金融数据下载问题。

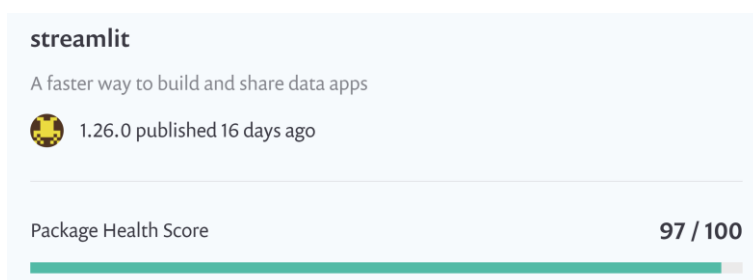


图 26. Snyk Advisor 对 Streamlit 库的评分，2023 年 9 月



请大家完成以下题目。

Q1. 安装 Anaconda。

Q2. 从 Anaconda Navigator 进入 JupyterLab 并完成测试。编写并执行图 20。再次提醒，快捷键组合点击“Ctrl + Enter”完成当前代码块运算。

Q3: 打开 Anaconda Navigator，查看已安装 Python 库的版本。

Q4: 从 Anaconda Navigator 进入 Anaconda Prompt，然后安装 pandas-datareader 和 Streamlit。

* 这些题目很基础，本书不给答案。



下一章，我们将深入了解鸢尾花书自主探究学习的利器——JupyterLab。