

2. Anaconda

[Anaconda](<http://www.continuum.io/downloads>)是一个很好用的Python IDE，它集成了很多科学计算需要使用的**python**第三方工具包。

Anaconda 和 Jupyter notebook已成为数据分析的标准环境。Anaconda是包管理器和环境管理器，Jupyter notebook 可以将数据分析的代码、图像和文档全部组合到一个web文档中。

- 1. Anaconda是什么？
- 2. 如何安装？
- 3. 如何管理包？
- 4. 如何管理环境？
- 5. pip安装使用详解

2.1 Anaconda是什么

Anaconda在英文中是“蟒蛇”，麻辣鸡（Nicki Minaj妮琪·米娜）有首歌就叫《Anaconda》，表示像蟒蛇一样性感妖娆的身体。

为什么需要 Anaconda？有以下3个原因：

1) Anaconda 附带了一大批常用数据科学包，它附带了 conda、Python 和 150 多个科学包及其依赖项。因此你可以立即开始处理数据。

2) 管理包Anaconda 是在 conda（一个包管理器和环境管理器）上发展出来的。在数据分析中，你会用到很多第三方的包，而conda（包管理器）可以很好的帮助你在计算机上安装和管理这些包，包括安装、卸载和更新包。

3) 管理环境为什么需要管理环境呢？比如你在A项目中用了 Python 2，而新的项目B老大要求使用Python 3，而同时安装两个Python版本可能会造成许多混乱和错误。这时候 conda就可以帮助你为不同的项目建立不同的运行环境。还有很多项目使用的包版本不同，比如不同的pandas版本，不可能同时安装两个 Numpy 版本，你要做的应该是，为每个 Numpy 版本创建一个环境，然后项目的对应环境中工作。这时候conda就可以帮你做到。

2.2 如何安装Anaconda？

Anaconda 可用于多个平台（Windows、Mac OS X 和 Linux）。你可以在下面地址上找到安装程序和安装说明。根据你的操作系统是32位还是64位选择对应的版本下载。官网地址：[Anaconda](#)

Anaconda 的下载文件比较大（约 500 MB），因为它附带了 Python 中最常用的数据科学包。如果计算机上已经安装了 Python，安装不会对你有任何影响。实际上，脚本和程序使用的默认 Python 是 Anaconda 附带的 Python。

注意：如果你是 windows 10 系统，注意在安装 Anaconda 软件的时候，右击安装软件→选择以管理员的身份运行。

安装方法

2.3 如何管理包？

完成安装后，如果你是在 windows 上操作，按下面图打开 Anaconda Prompt（或者 windows dos /Linux、Mac 下的终端）

2.3.0 设置国内镜像

如果需要安装很多 packages，你会发现 conda 下载的速度经常很慢，因为 Anaconda.org 的服务器在国外。所幸的是，清华 TUNA 镜像源有 Anaconda 仓库的镜像，我们将其加入 conda 的配置即可：

添加 Anaconda 的 TUNA 镜像

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/
```

TUNA 的 help 中镜像地址加有引号，需要去掉

设置搜索时显示通道地址

```
conda config --set show_channel_urls yes
```

执行完上述命令后，会生成 ~/.condarc (Linux/Mac) 或 C:\Users\USER_NAME\condarc 文件，记录着我们对 conda 的配置，直接手动创建、编辑该文件是相同的效果。

[anaconda 清华镜像站](#)

2.3.1 更新软件

第一次安装好 [Anaconda](#) 以后，可以在命令行输入以下命令使 [Anaconda](#) 保持最新：

```
conda update conda
```

```
conda update anaconda
```

2.3.2 查看已经安装的packages

在命令行下输入：

```
conda list
```

2.3.3 安装卸载

```
conda install package_names  
conda remove package_names
```

上面命令中的package_names是指你要卸载包的名称，例如你想卸载pandas包：

2.3.4 更新包

在终端中键入：

```
conda update package_name
```

如果想更新环境中的所有包（这样做常常很有用），使用：

```
conda update --all
```

2.3.5 查询包名

如果不知道要找的包的确切名称，可以尝试使用以下命令进行搜索

```
conda search search_term
```

例如，我知道我想安装numpy，但我不清楚确切的包名称。我可以这样尝试：

```
conda search num
```

2.3.6 查看 conda 的信息：

```
conda info
```

conda 是一种很强大的工具，具体用法可以参照它的[文档](#)。
也可以参考它的 [cheat sheet](#) 来快速查看它的用法。

2.4.如何管理环境？

conda 可以为你不同的项目建立不同的运行环境。

2.4.1 安装nb_conda用于notebook自动关联nb_conda的环境。

```
conda install nb_conda
```

2.4.2 创建环境

在终端中使用：

```
conda create -n env_name package_names
```

上面的命令中，env_name 是设置环境的名称（-n 是指该命令后面的env_name是你要创建环境的名称），package_names 是你要安装在创建环境中的包名称。例如，要创建环境名称为 py3 的环境并在其中安装 numpy，在终端中输入

```
conda create -n py3 pandas。
```

2.4.3 指定要安装在环境中的 Python 版本

当你同时使用 Python 2.x 和 Python 3.x 中的代码时这很有用。要创建具有特定 Python 版本的环境，例如创建环境名称为py3，并安装最新版本的Python3在终端中输入：

```
conda create -n py3 python=3
```

或也可以这样创建环境名称为py2，并安装最新版本的Python2：

```
conda create -n py2 python=2
```

因为做的项目不同，有时候会用到Python2，还有时候会用到Python3。所以在自己的计算机上创建了这两个环境，并分别取了这样的环境名称：py2,py3。这样我可以根据不同的项目轻松使用不同版本的python。如果你要安装特定版本（例如 Python 3.6），请使用

```
conda create -n py python=3.6
```

2.4.4 进入环境

Windows 上，你可以使用以下命令进入

```
activate my_env
```

在 OSX/Linux 上使用这样

```
source activate my_env
```

进入环境后，你会在终端提示符中看到环境名称。可以用conda list 查看环境中默认安装的几个包：

在环境中安装包的命令与前面一样：conda install package_name。不过，这次你安装的特定包仅在你进入环境后才可用。

2.4.5 离开环境

在 Windows 上，终端中输入：

```
deactivate
```

在 OSX/Linux 上 输入：

```
source deactivate、
```

2.4.6 共享环境

共享环境非常有用，它能让其他人安装你的代码中使用的所有包，并确保这些包的版本正确。比如你开发了一个药店数据分析系统，你要提交给项目部署系统的王二狗来部署你的项目，但是王二狗并不知道你当时开发时使用的是哪个python版本，以及使用了哪些包和包的版本。这怎么办呢？

你可以在你当前的环境中终端中使用

```
conda env export > environment.yaml
```

将你当前的环境保存到文件中包保存为YAML文件（包括Python版本和所有包的名称）。

命令的第一部分 `conda env export` 用于输出环境中的所有包的名称（包括 Python 版本），在“notebook工作文件夹”下（及你在终端中上图的路径）可以看到导出的环境文件。

在 GitHub 上共享代码时，最好同样创建环境文件并将其包括在代码库中。这能让其他人更轻松地安装你的代码的所有依赖项。导出的环境文件，在其他电脑环境中如何使用呢？

首先在conda中进入你的环境，比如`activate py3`然后在使用以下命令更新你的环境：#其中-f表示你要导出文件在本地的路径，所以/path/to/environment.yml要换成你本地的实际路径

```
conda env update -f=/path/to/environment.yml
```

2.4.7 列出环境

有时候会忘记自己创建的环境名称，这时候用就可以列出你创建的所有环境。

```
conda env list
```

你会看到环境的列表，而且你当前所在环境的旁边会有一个星号。默认的环境（即当你不在选定环境中时使用的环境）名为 `root`。

2.4.8 删除环境

如果你不再使用某个环境，可以删除指定的环境（在这里环境名为 `env_name`）。

```
conda env remove -n env_name
```

最后重新再强调下，不要被上面的命令吓到。虽然上述命令多，给了谁都记不住的。后面你跟着我在知乎上多做项目，用的多了自然记住了。你只需要跟着上面步骤操作下，并理解了每一步是干什么的就可以了。后面遇到要做的事情，忘记了回头查这个文档就可以了。

2.5 pip使用

pip 是一个安装和管理 Python包的工具，下载相关python包很是简单。不仅把需要的包下载下

来，而且会把相关依赖包，给下载下来，这是pip的优势所在。

anaconda 已经安装了这个工具，直接使用即可。

2.5.1 下载包

我们可以通过这个网站查看自己所需的Python包及相关依赖的包 [windows依赖包](#)

然后通过

```
pip install opencv_python-3.4.1-cp36-cp36m-win_amd64.whl
```

2.5.2 在线安装

可直接在线将包下载下来并安装

```
pip install sklearn
```

2.5.3 卸载安装包

```
pip uninstall sklearn
```