

Installing and using Anaconda

Lucy Hackett

3/16/2022

Contents

What is Anaconda?	1
Installation	1
Using Anaconda	2
Creating, activating conda environments	2
Installing packages in conda environments	2
Appendix: Using jupyter notebooks	3

What is Anaconda?

Anaconda is a **package manager**. For all your favorite programming languages, you use packages. Probably, lots of packages. The issue is that packages are complicated- they change over time via different versions, they may not play well together, and we all know the frustration of trying to use someone else's code and having errors pop up because we don't have the right dependencies installed.

Enter: Anaconda.

Anaconda is a systematic way of installing packages, making sure they don't conflict with each other, and creating separate *environments* that allow you to work with different packages/versions with different projects. To give you an example, I work (for now) with python 3.7 on my computer, but for a class I needed to use a package that required python 3.5 or lower. Anaconda allows me to make a new environment where I could have python 3.5 and the necessary package installed, without messing up my python 3.7 installation that everything else depended on! Even beyond environments, Anaconda just simplifies the installation of many packages. Some packages have complicated dependencies that require tricky installation; Anaconda usually takes care of all that for you.

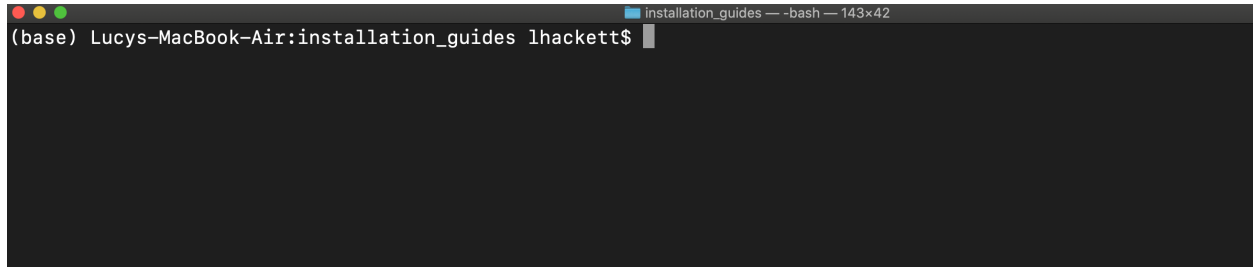
Installation

Anaconda comes in 2 versions: Anaconda, and miniconda. Anaconda is a beast- it includes the package manager and comes with a whole slew of packages installed for you directly. This can be nice, as you avoid installing lots of packages later. However, I tend to recommend miniconda, which includes just the package manager, python and a few other basic packages. Why? The main reason is space. Not all laptops have enough memory for all those packages, many of which you may never use. The rest of this guide will focus on installing and using miniconda, but the instructions and ideas are very similar for full Anaconda.

You can find the installation links [here](#); just find your OS, and follow the installation instructions (note: for mac, download the pkg installer)!

Using Anaconda

Now what? Let's create an environment for doing metrics in python. I'm going to walk you through creating an environment, activating it, and installing packages. First, we need to open up a command line. On Macs this command line is called Terminal (look for it in Applications) and uses the bash scripting language.¹ On PC's, you need to open the PowerShell, which allows you to run bash script.



Creating, activating conda environments

Above is a screenshot of my terminal. If you've used terminal before, you'll note that something has now changed - there this **(base)** thing in front of my directory. This is our current environment! I generally try to leave the base environment alone, so let's create a new environment by typing (in the terminal):

```
conda create --name env-name
```

where **env-name** is the new name of your environment. To activate this environment, we just type:

```
conda activate env-name
```

And you should see that **(base)** has changed to **(env-name)**. We are now "in" the env-name environment, and we *only have access to the packages installed in env-name*, which at the moment is none! So let's install.

Installing packages in conda environments

I am going to set up a simple environment that just has python 3.7, jupyter notebook, spyder, statsmodels, pandas and numpy for some basic metrics. Remember you can always install more packages later.

Making sure that our new environment is activated, we will run `conda install`. With this, we can choose to specify the version we want, or if we do not specify then the most recent version will attempt to be installed. So below I will specify the python version but let all the other packages be the most recent. I won't get into conda channels here, but note that googling "conda install PACKAGE-NAME" will usually turn up code you can copy paste into your terminal to install packages. You can install packages one at a time, or several at once:

```
conda install python=3.7 notebook spyder statsmodels pandas numpy
```

Voilà! Now our environment is set up with these packages. If you want to test this, try:

```
conda list
```

This should return a list of the installed packages. If you create a new environment and run this same command, you will notice that you get an empty list because there are no programs available!

To deactivate a conda environment, just type:

```
conda deactivate
```

Now you have conda, python and jupyter notebooks installed!

¹Also Linux, but I won't treat Linux separately here. If you are using Linux I'm going to assume you have the coding skills to do most of this yourself.

Appendix: Using jupyter notebooks

To use a jupyter notebook, first make sure your environment that has notebook installed is activated. Then type in the terminal:

```
jupyter notebook
```

And a notebook window should pop up in your browser.