# CONDA Installation of Python 3.5 w/ 2nd 2.7 Environment

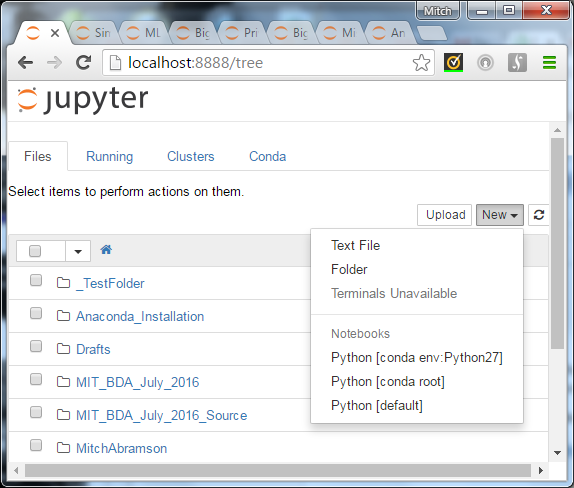
Includes Data Science Packages Used in “MIT Big Data and Social Analytics Online Certificate Program” --July 2016

|  |
| --- |
| **Warning!** -- This content added to this file on 1/1/2017.  The environment in this document worked for about 3 months (9/2016 – 12/2016). Then randomly, Spyder stopped working. Research suggests that using “conda update –all” may have resulted in some incompatibilities in the Python 3.5 environment which was also the root environment and that this may have disabled Spyder. Worse, running “conda update –all” on that environment (as part of the debugging effort) destroyed the environment (as per this [post](https://github.com/conda/conda/issues/4160)). Use this command with caution and/or look into ways to test updates in pieces and check them if you wish to avoid the pain the author just went through to fix his environment.  The intent of this document was to provide a doc that combines some learning, some instruction, and some automation to re-create the Anaconda 4.1 setup built to work in Python 3.5 with additional environments to replicate what was in the Python 2.7 envrionment for the “MIT Big Data and Social Analytics” class of 2016. It may still be possible to use this document that way for Anaconda 4.1, but changes since this document was written have introduced risk that some packages may not play together if/when seemingly unrelated packages are updated.  More up-to-date content is planned for the future. |

Content from Installation of 9/2016:

On my Windows 7 laptop (w 16 Gigs of RAM), I currently have three Notebook creation options as shown in the screenshot below. Python “[default]” and “[conda Root]” are really both pointers to the same Python 3.5 environment (creating Python 3.5 Noteboks) while “conda env: Python27” uses its own separate environment to create Python 2.7 nobebooks.

Note: The setup shown in this document, if followed exactly, will result in Spyder and other Non-Jupyter python tools running Python 3.5 by default while Jupyter will allow the creation, editing, and running of notebooks in both Python 3.5 and 2.7:



If you are using different specs for your machine and a different OS or version, your results may vary from mine and/or you may encounter things I can’t anticipate. Windows7 users hopefully can use this instruction set as is. Mac and Windows 8 or higher users hopefully will find this useful but may have to investigate minor system differences. Note that content below labeled “Before you Begin” (for example) was originally figured out with the help of an online topic written for Mac users even though I was originally trying to figure out what was going wrong for me on Windows 7.

The shortest path to replicate my environment is this:

1. <https://www.continuum.io/downloads> - Download Windows installer for your machine for Anaconda / Python 3.5 from this link.
   1. **Before you Begin:** Make sure the install path you are planning to use does not have any spaces in the names of any folders. By default, it may try to set you up in C:\Anaconda3 – this is fine. But if it puts you in a subfolder of “My Documents” or some other Windows profile folder with spaces in it, don’t let it. Move your path. FYI – I created a path for my installation as follows: c:\ProgramFilesCoders\Anaconda3
2. Then follow the instructions given below to add a 2.7 environment and to do essential updates to replicate the “MIT Big Data and Social Analytics” class environment

**Note:** This process should give you all of the packages from class installed under the Python 2.7 environment and only some of the packages from our class under you primary 3.5 environment. The ones that are left out of 3.5 are left off because they may not be compatible. If you install an incompatible package, cleaning up an environment can get messy, so if the few missing packages are not yet needed specifically with your 3.5 code, it might make sense to wait until upgrades are published. This notebook can be loaded in either your 3.5 or 2.7 environment with hyperlinks to research that contributed to this decision:



To “download” this file – open it in Notepad and save it to your hard drive. Then you can upload it into your finished Anaconda environment once you create it.

**Steps to set up (and understand) your Anaconda Python environment(s):**

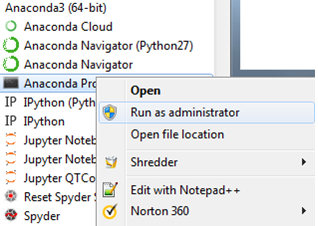
Some additional commands are provided in these steps that will informationally give you a better understanding of your new environment.

1. Recommended but not required: Take the Anaconda test drive at: <http://conda.pydata.org/docs/test-drive.html>

But I recommend you deviate from given instructions as follows:

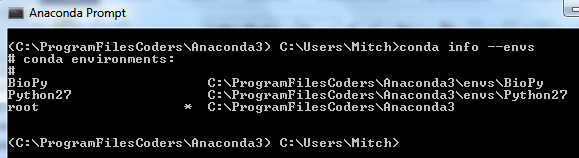
* 1. Make sure to remove any extra environments you create except the one for “biopython” which is small and useful for testing
  2. I skipped the exercise to install a free trial of a pay-for package from Continuum. Rational for this: it is better to test this when you are actually ready to begin your free trial so there are no issues with time running out before you get to use it.

1. If you take the test drive as indicated above, as you go through the steps below you will see a biopython environment listed with the name you gave it during the test drive. If not, you will not see one (having just a root and Python 27 environment once you create it).
2. In case you get into any trouble during this process, keep this URL handy. Posting to the Anaconda forums can get you help: <https://groups.google.com/a/continuum.io/forum/#!forum/anaconda>
3. Once you complete default installation of Anaconda3, launch an Anaconda command prompt from your programs list (in Start Menu) as shown here. Recommended: right-click on the icon and “run as administrator” so commands execute w/ less system warnings.



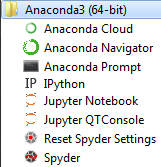
1. Use this command to view all existing environments in your Anaconda installation. Note: an “environment” is created to allow you to run isolated python worlds without them conflicting with each other, the solution to being able to create both Python 2.7 and 3.5 notebooks on your computer.

Command: conda info --envs



Note: When you do the above step, you will probably see just “Root”. If you have multiple environments (as above), the one with “\*” is currently active.

1. If you wish to verify all is as it should be before all the work that follows, check your program files folder and just verify all of the icons are there. You should now have these icons (for Anaconda3):



Note: After adding Python27 using file provided in this walkthrough, you may see more icons. This is a harmless side-effect of creating a secondary environment with almost all of the packages of root in it.

1. If you simply create the Python27 environment (using the “create” command), you get only a small subset of the packages you will find in your root installation. To address this, you will create your Python 27 environment by importing it from a file that was created from my Python 2.7 environment after I already jumped through hoops for you to install a lot of packages. You will be using this command but replace the path shown with the location and filename where you download this file:



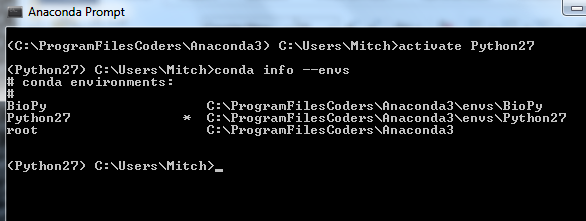
Command:

conda env create -f C:\Anaconda3\Python27\_EnvPackageInst.txt

1. Now you need to activate your python 27 environment:

Command: Activate Python27

1. Verify it is active by re-doing the info command: conda info –envs



1. You should see something like what is shown above with “\*” near Python27
2. Now you need to execute each of these commands to add packages to your environment to replicate the Python 27 class environment we worked in:

Commands (execute these one by one in Anaconda Prompt):

* Channel command gets enhanced packages from Allan Kent that should be a little better than what we had in class
* Other PIP commands will get you the packages we used in class

Commands provided in this text file:



1. In my environment, I chose to have the latest and greatest of each of these packages. To do this, I used the update command:

Command: conda update –all

1. Now switch back to Python 3.5 and add just those packages that are compatible with 3.5.

Commands (to make sure we are in root and to check it):

deactivate Python27

activate root

conda info --envs

1. A list of these commands is included in the text file below. Add these commands one by one just like you did with Python 2.7 (you will notice the list for 3.5 is shorter):



1. To see a list of what packages are in either of your environments, first activate that environment and then use:

Command: conda list

1. Did you encounter any problems with this process? Help is available through the Continuum.io forums at:

<https://groups.google.com/a/continuum.io/forum/#!forum/anaconda>

1. Optional: Something not in the test drive – so including it here just in case you need it
   1. Need to add another Python environment that you can create notebooks in?
   2. The default create env command does not add the environment to your notebook kernels menu (you won’t see a biopython notebook menu item for example even if you built that environment during the Anaconda test drive mentioned in step 1).
   3. To create an additional environment and hook it up to the Kernels/Notebook menu on Jupyter, you need to use create env syntax like this example (this example assumes that you want Python 3.3 as an additional environment for some reason):

conda create –n Python33 python=3.3 ipykernel

(the ipykernel argument adds it to the kernel/notebook menu on Jupyter)

1. To launch your Jupyter environment, you can either type “Jupyter notebook” from an anaconda command prompt, or you can use the “Jupyter Notebook” icon in your Anaconda3 programs menu.
2. By default, when your Jupyter environment launches, you will probably find that “root” = the highest level of your “Documents” or “My Documents” folder on Windows. To change this:
   1. File you need to edit is in this path for Windows 7: C:\Users\<your username>\.jupyter\
   2. Above path may vary for other Windows releases so google Users path for your particular release if it is not found as shown above
   3. File to edit: jupyter\_notebook\_config.py
   4. Search the file for this line and if it is commented out, remove the “#” at start so it is not commented out:

c.NotebookApp.notebook\_dir =

* 1. Change path to your desired starting path making sure to escape each \ to a double \\ when you do so.
  2. Your edit will look something like this (but all on one line):

c.NotebookApp.notebook\_dir = C:\\PathOfYourChoosing\\NotebooksRoot

1. Test your environment and if all works as it should – celebrate! You have come to the end of this process.

Some good things to know:

1. Best practice: add a markdown cell to the top of each notebook indicating the Python Version and name of the environment it came from. I use a convention like “Python 3.5 [root]”. This can help with environment debugging down the road (what environment was this code built in?) and also help you share the Notebook with other users (what version of Python do they need when selecting environments on their system?)
2. Terminal menu will still be grayed out even if you follow all instructions in this guide to the letter. Forums seem to indicate terminal menu does not work in Windows, but hopefully there will be a fix for that in the future. In the interim, for local PC installations like the one described in this document, the Anaconda Prompt is sufficient to do anything you would have done through the terminal.

Summary of Useful Links Contained in this document:

1. Anaconda Help Forums: <https://groups.google.com/a/continuum.io/forum/#!forum/anaconda>
2. Anaconda Test Drive (learn basics): <http://conda.pydata.org/docs/test-drive.html>
3. Download Page for Anaconda Software: <https://www.continuum.io/downloads>