Python setup

Jim Bagrow

STAT/CS 287 — Data Science 1

Data Science 1 - Python setup

Outline:

1. Where to get Python
2. Setting up a work environment
    1. writing code with Spyder
    2. installing packages with Anaconda
3. Running code
    1. IPython—Interactive Python
4. Getting help

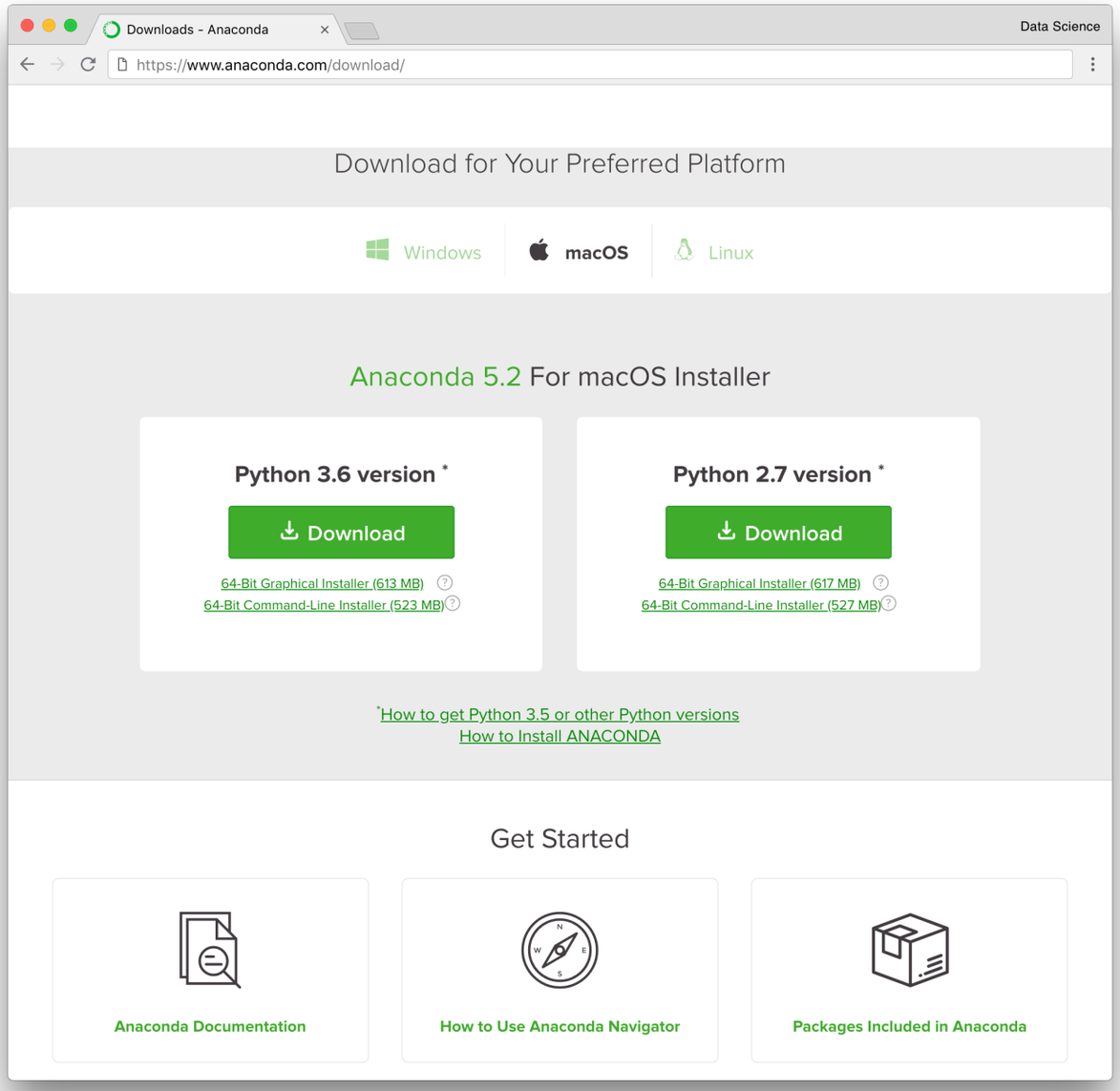
1. Where to get Python

Where to get Python

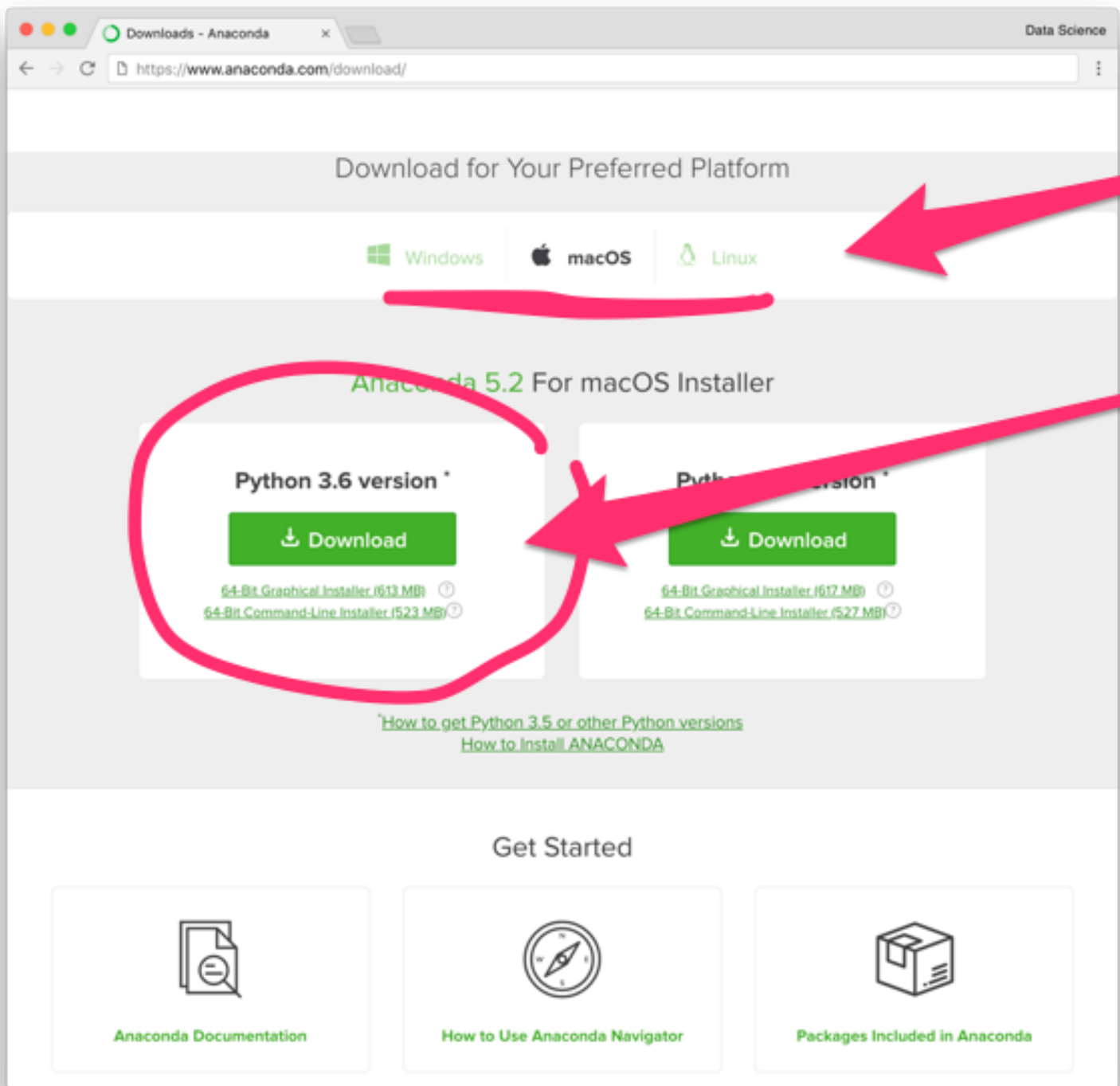
https://www.anaconda.com/downloads

ANACONDA.

Anaconda is a free, cross-platform Python distribution that includes crucial Python libraries such as Numpy and Scipy



Where to get Python



Select your platform

Select Python 3.x

(Python 2.x code will not be accepted.)

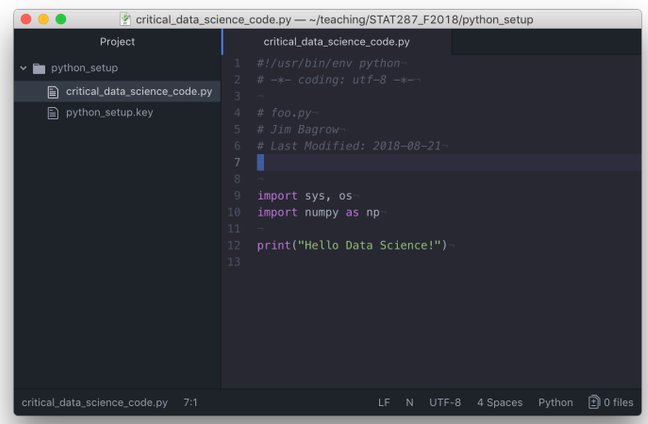
(Macs and Linux machines come with a version of Python already, but it's best to install Anaconda anyway in order to get all the packages set up for you.)

2. Setting up a work environment

Setting up a work environment

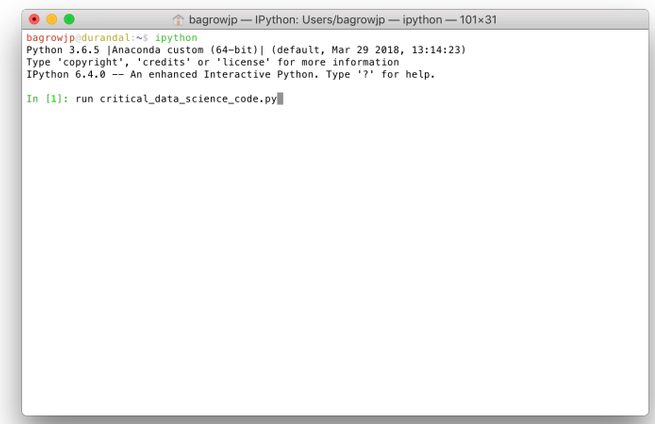
Python requires a tool to create Python code and a tool to execute Python code

Often these are **separate apps** such as a text editor and a terminal:



text editor

+



terminal

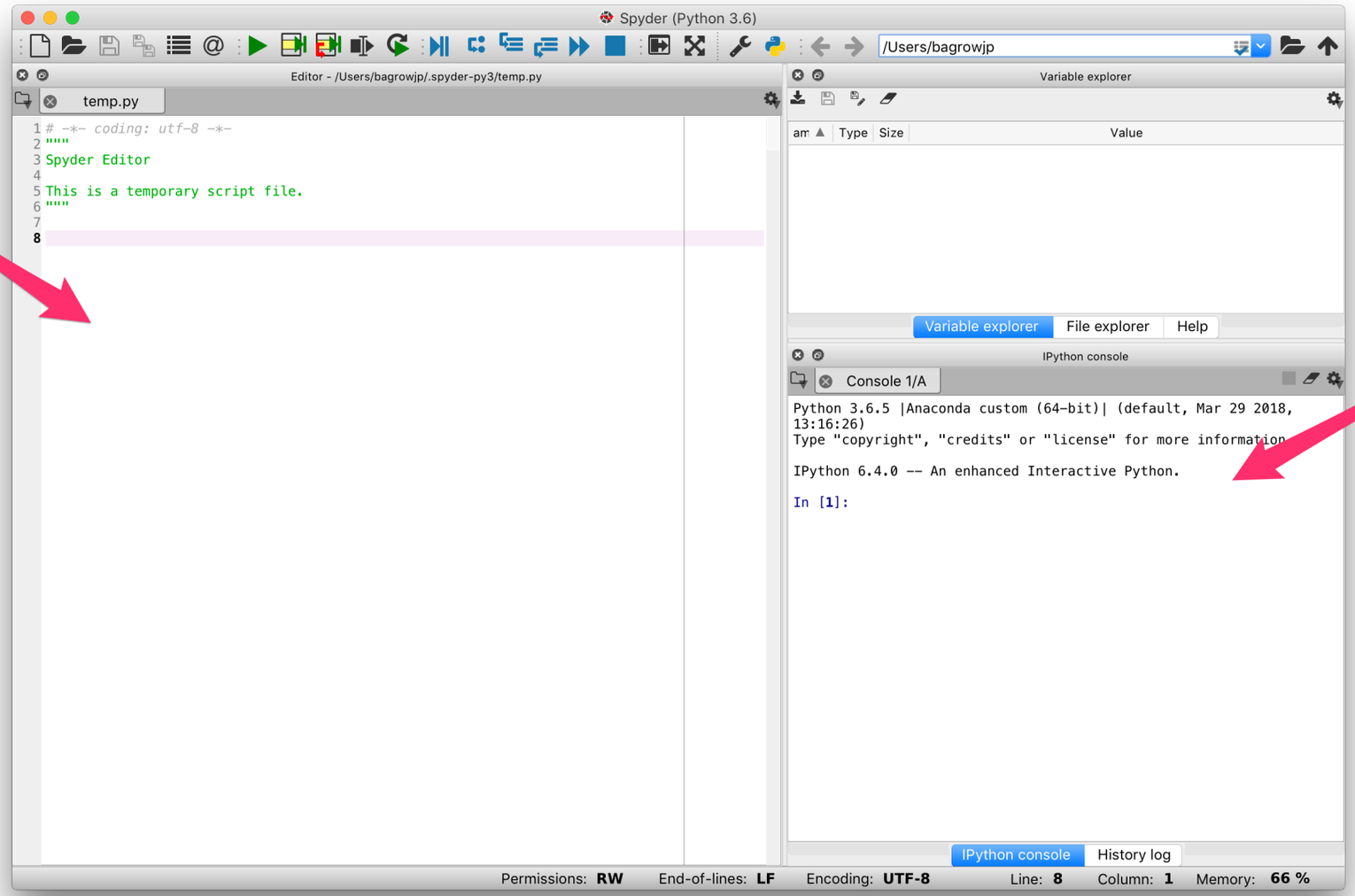
But apps exist that contain both →

Spyder - Python work environment

(Scientific PYthon Development EnviRonment)

text editor to create code

console (terminal) to run code



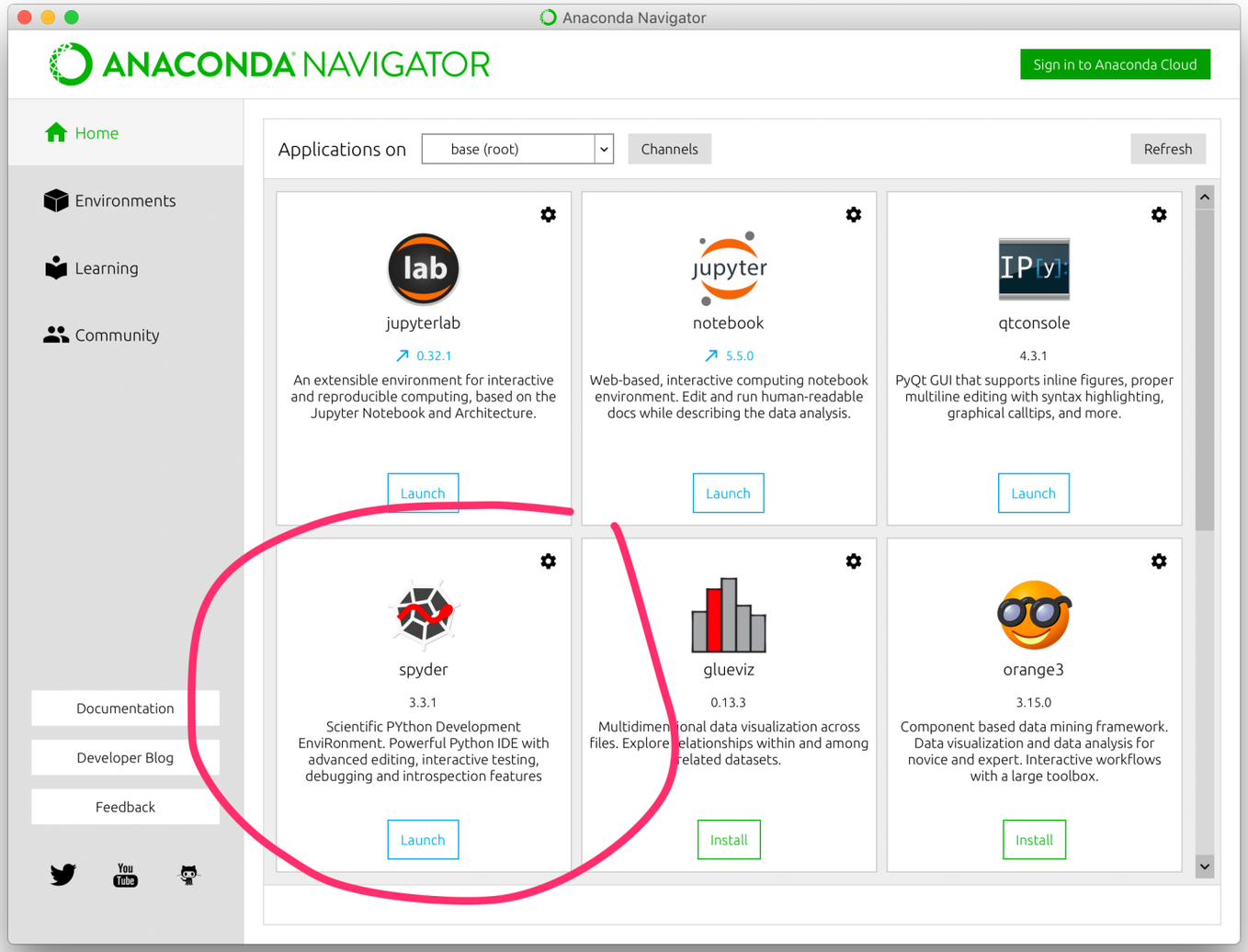
You are *STRONGLY* encouraged to do all your work within the free Spyder app. You can set up your own tools, but you are responsible for ensuring they function correctly (see Appendix).

Use Spyder!

Launching Spyder

1. Open the Anaconda Navigator app that installs with Anaconda Python

2. Click Launch on the Spyder pane

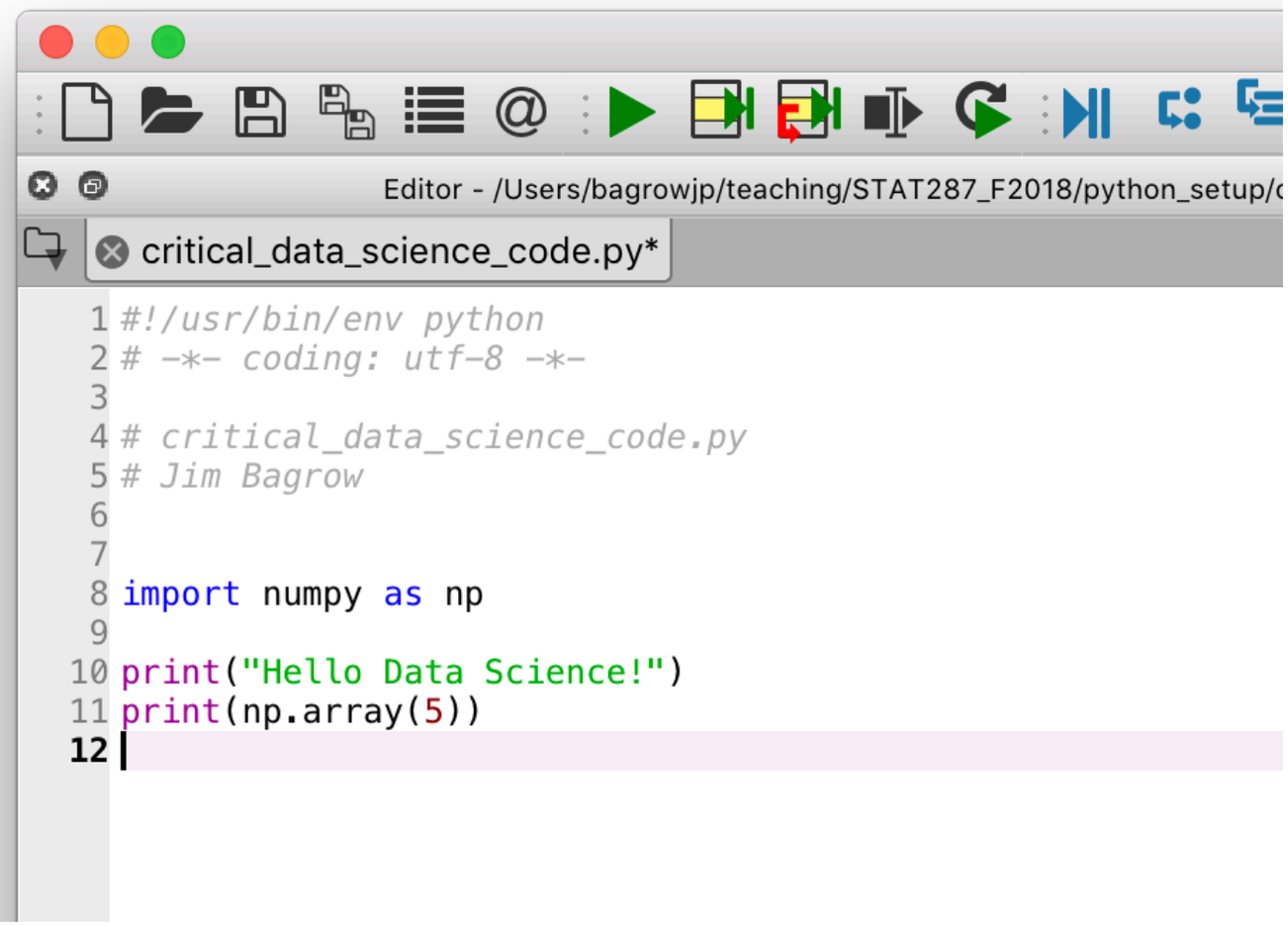


https://docs.anaconda.com/anaconda/navigator/

Writing code with Spyder

Write your code in the Spyder editor:

This creates a .py **script file**

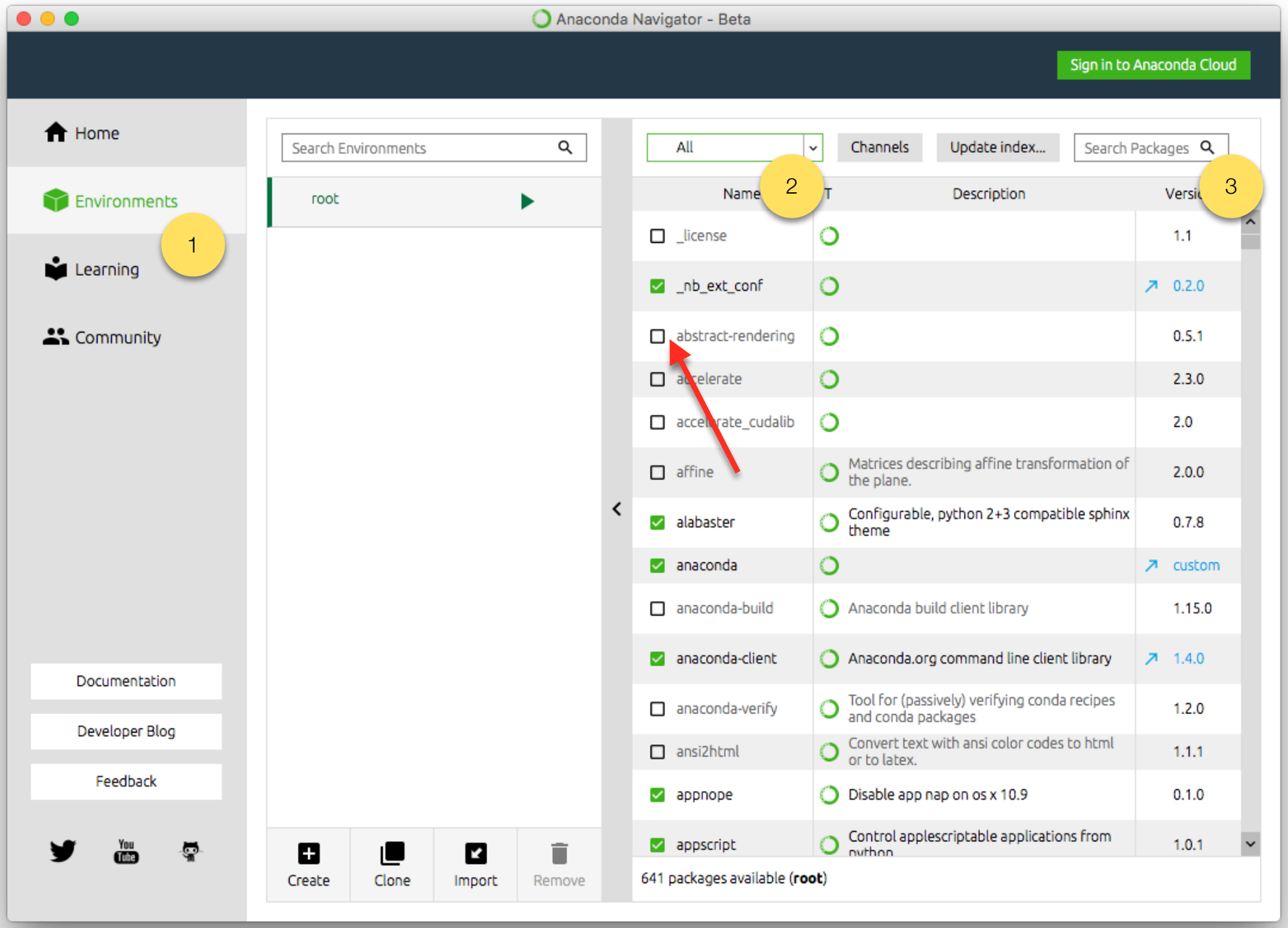


Installing packages with Anaconda

If necessary, to **install packages** from the Navigator:

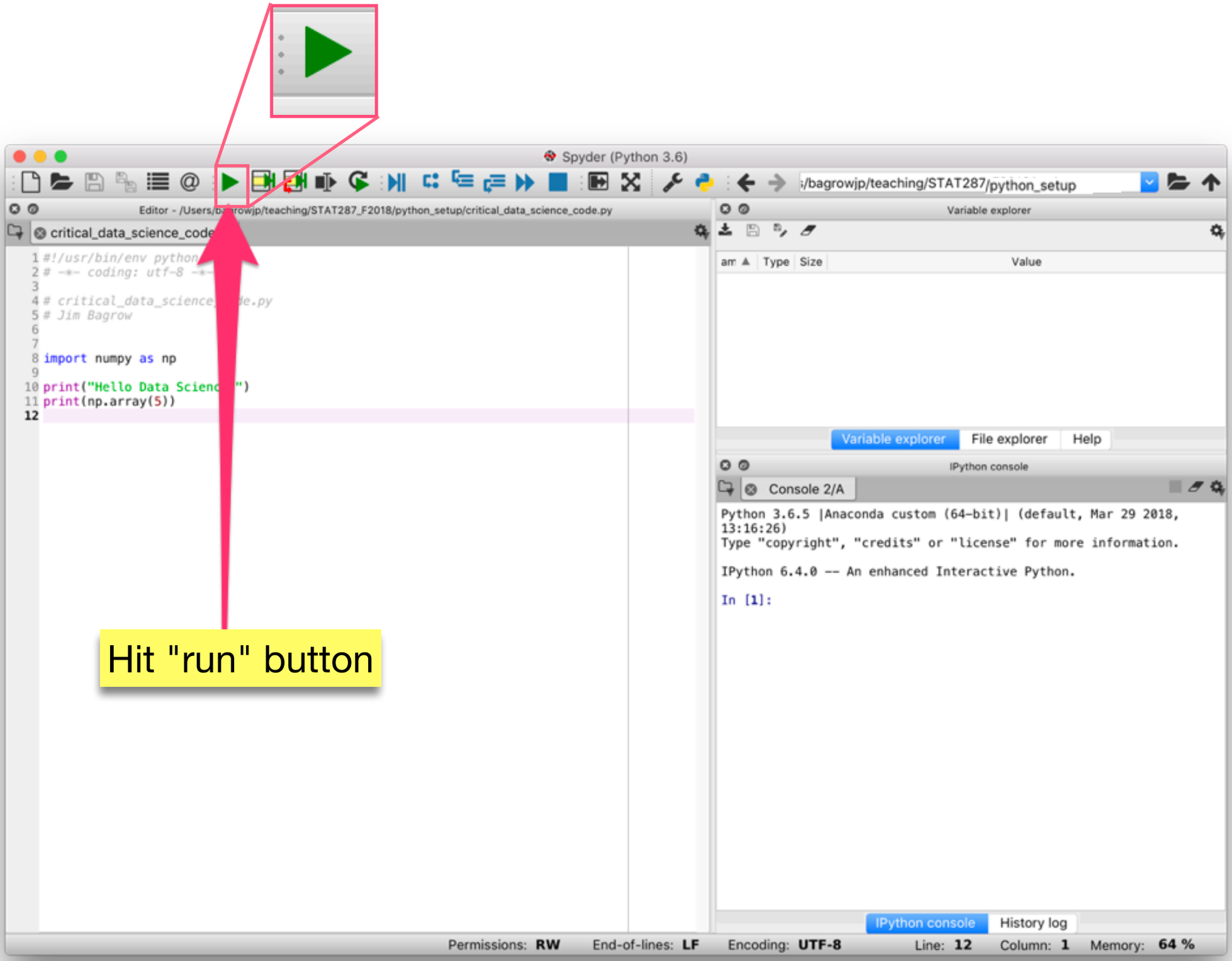
(1) Go to environments
(2) Select “All”
(3) Search for a package
→ Check the package's checkbox

Lastly, click the new apply button that appears and follow the resulting dialog box to finish the installation.



3. Running code

Running code with Spyder



Running code with Spyder

Spyder (Python 3.6)

Editor - /Users/bagrowjp/teaching/STAT287_F2018/python_setup/critical_data_science_code.py

critical_data_science_code.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# critical_data_science_code.py
# Jim Bagrow


import numpy as np

print("Hello Data Science!")
print(np.array(5))
```

Variable explorer

am	Type	Size	Value

Variable explorer | File explorer | Help

IPython console

Console 2/A

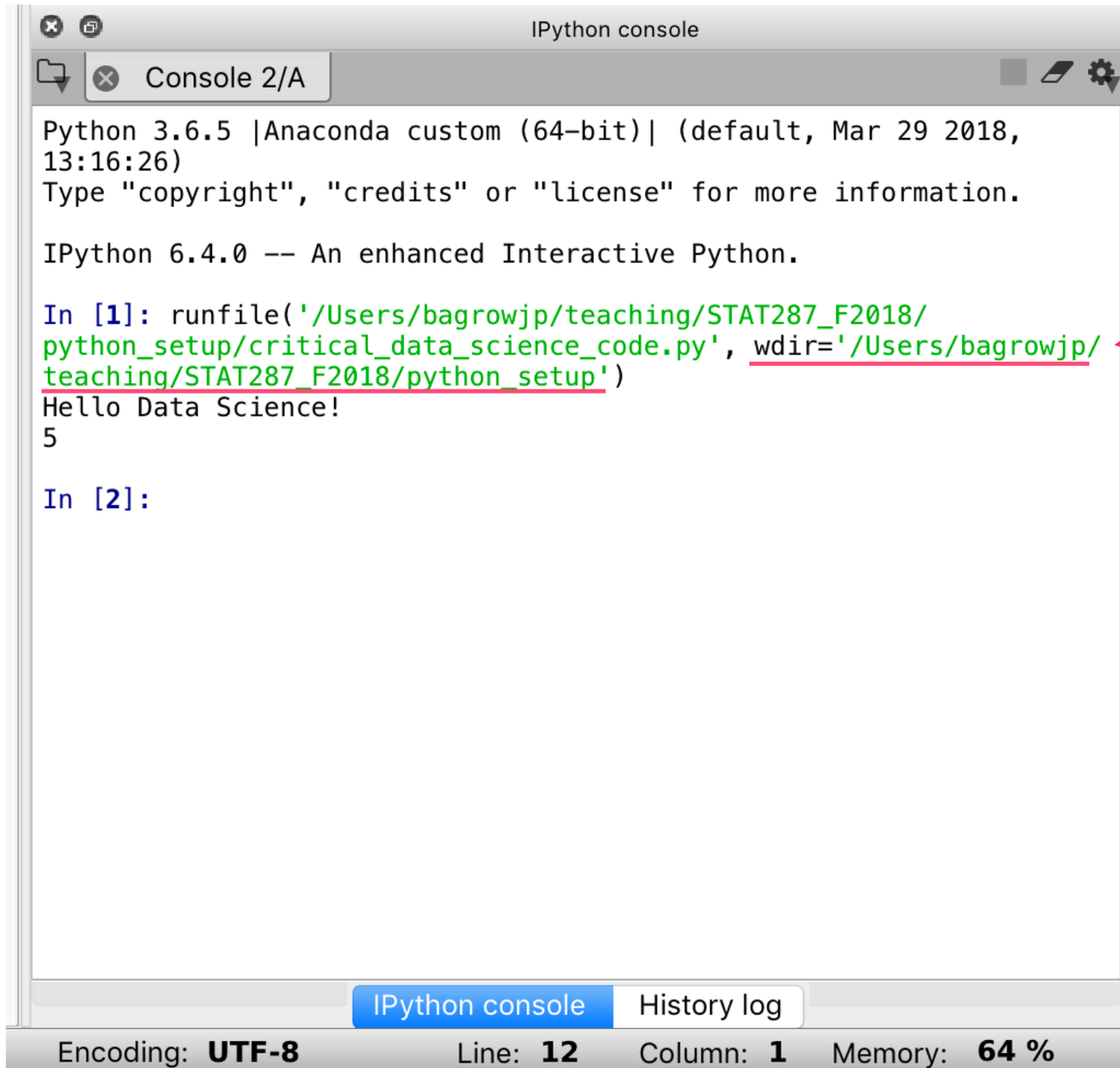
```
Python 3.6.5 |Anaconda custom (64-bit)| (default, Mar 29 2018,
13:16:26)
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/bagrowjp/teaching/STAT287_F2018/
python_setup/critical_data_science_code.py', wdir='/Users/bagrowjp/
teaching/STAT287_F2018/python_setup')
Hello Data Science!
5

In [2]:
```

IPython console | History log

Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 12 Column: 1 Memory: 64 %

and any output appears here!

Running code with Spyder

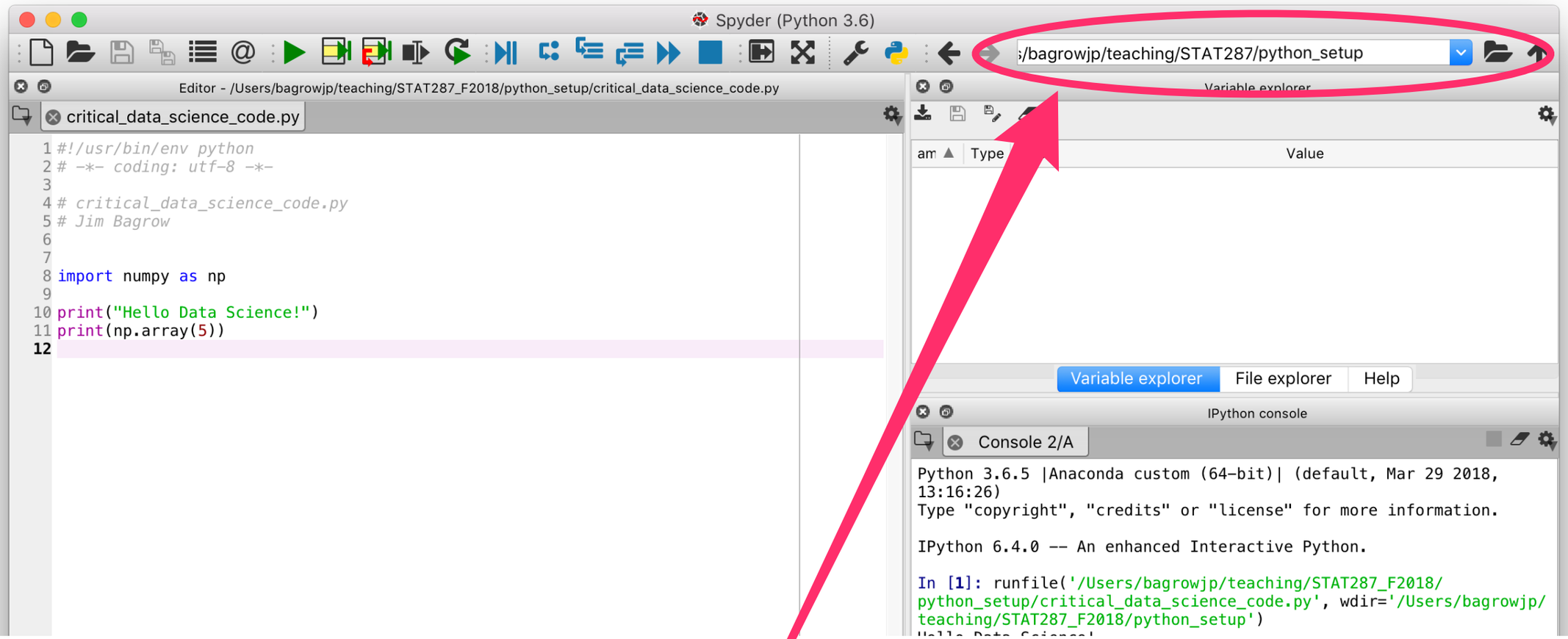


wdir = "working directory" the location where the code runs.

Make sure this is the location where you saved your .py script (this should be the enclosing folder of any assignments)

Any files you read or write inside your .py script must have locations specified *relative* to wdir

Running code with Spyder

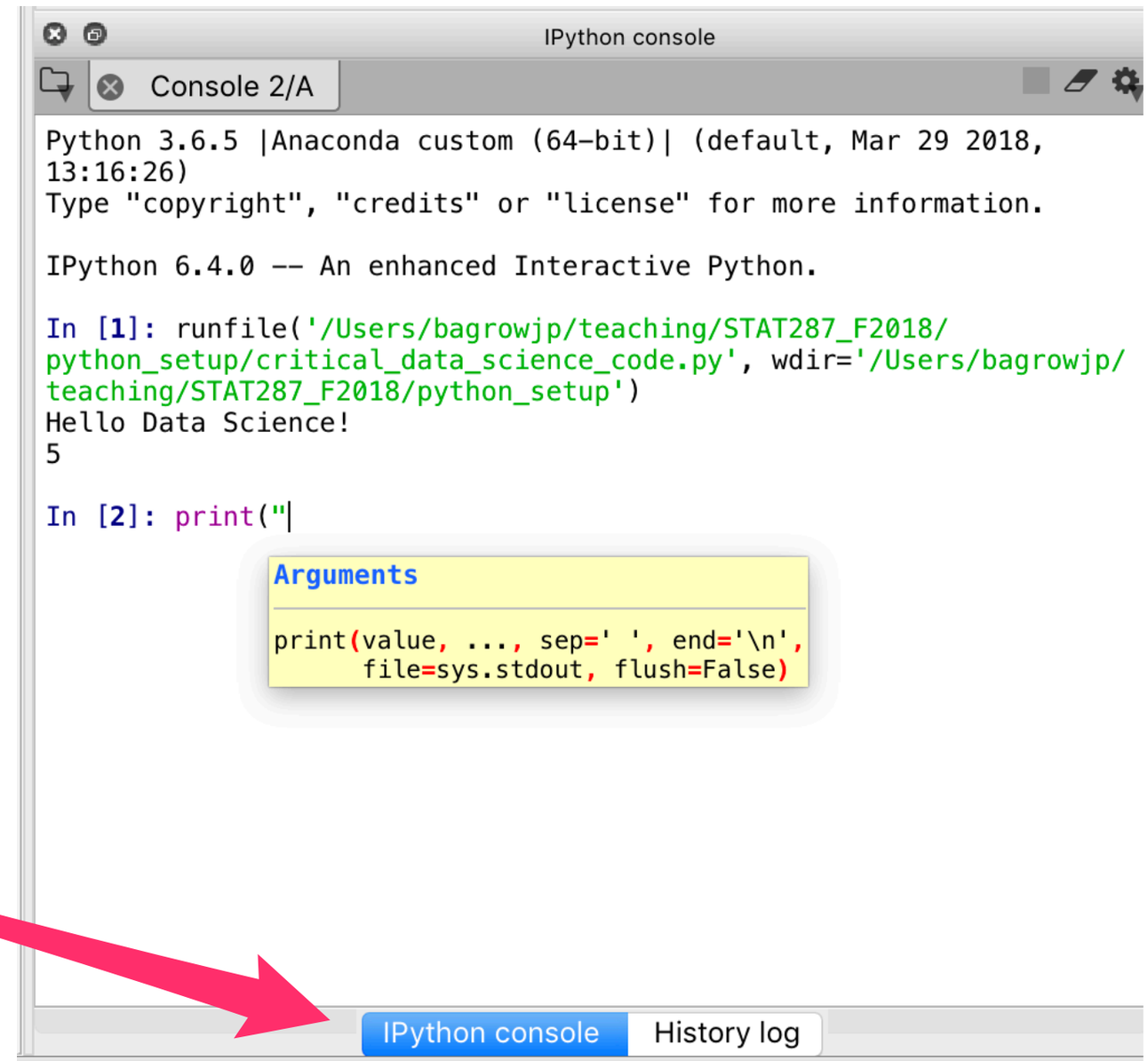


The current working directory is also listed here

Interactive Python

The console can also run individual python commands typed directly into it

This console is running **IPython**, a very handy command line tool for using Python *interactively*!

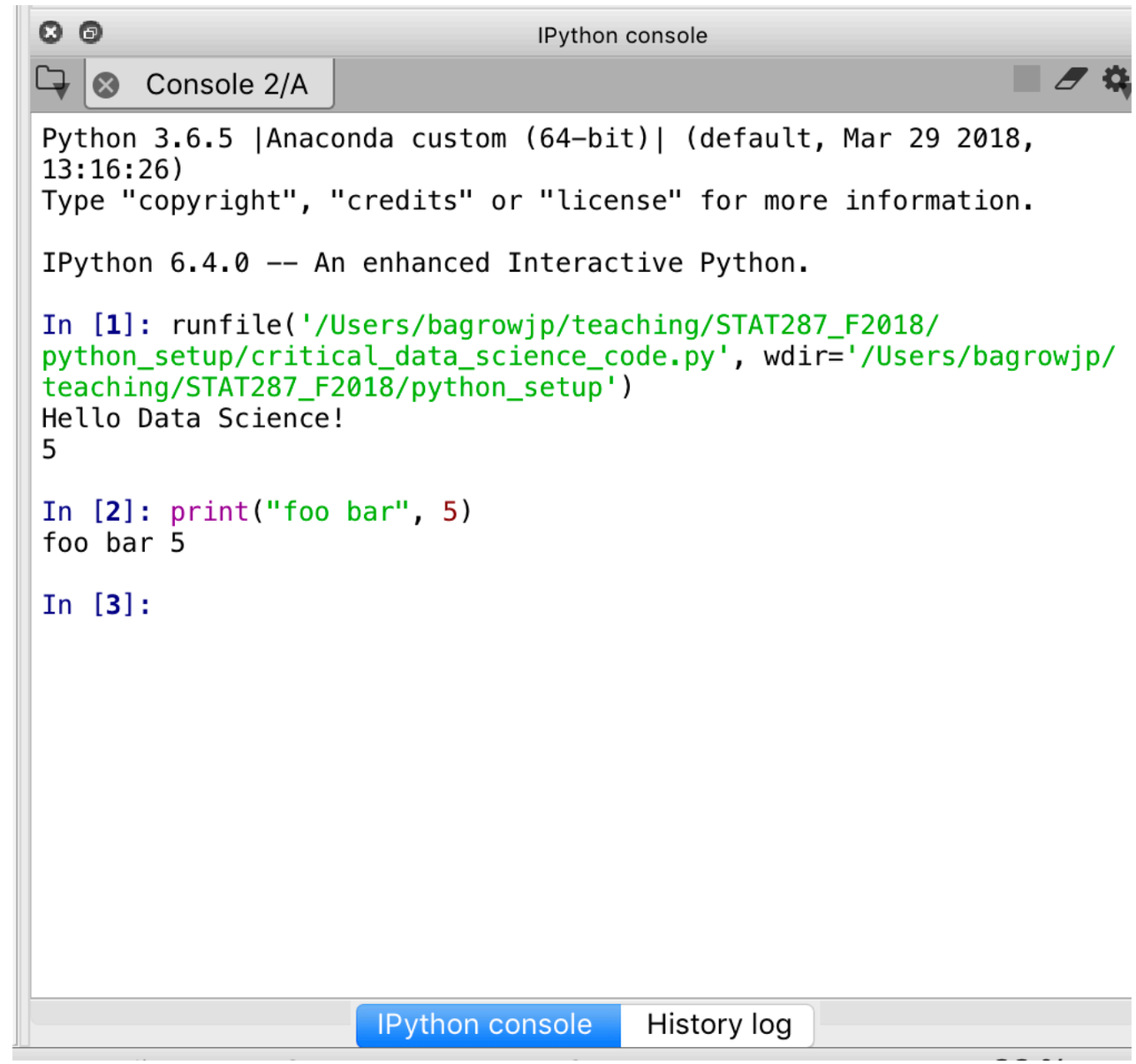


Interactive Python

The console can also run individual python commands typed directly into it

Just type your command and press enter/return to run it

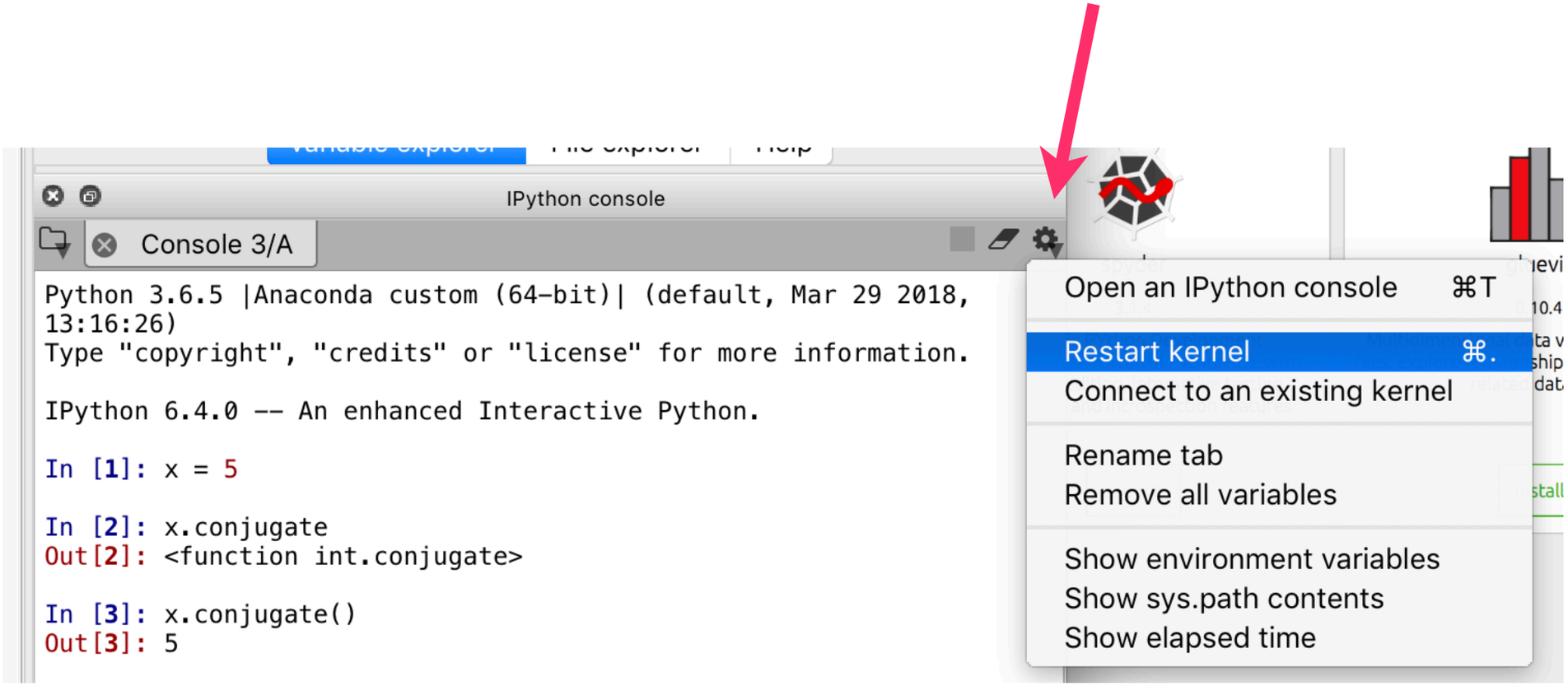
Running small snippets of your code interactively and examining the results is one of the **most important techniques** you have for writing correct code, especially as you build longer scripts



Side note: killing the console

Sometimes, you need to restart Python, such as if you have created an endless loop

The "gear" button in the upper right of the console lets you do this, just click and select "Restart kernel"

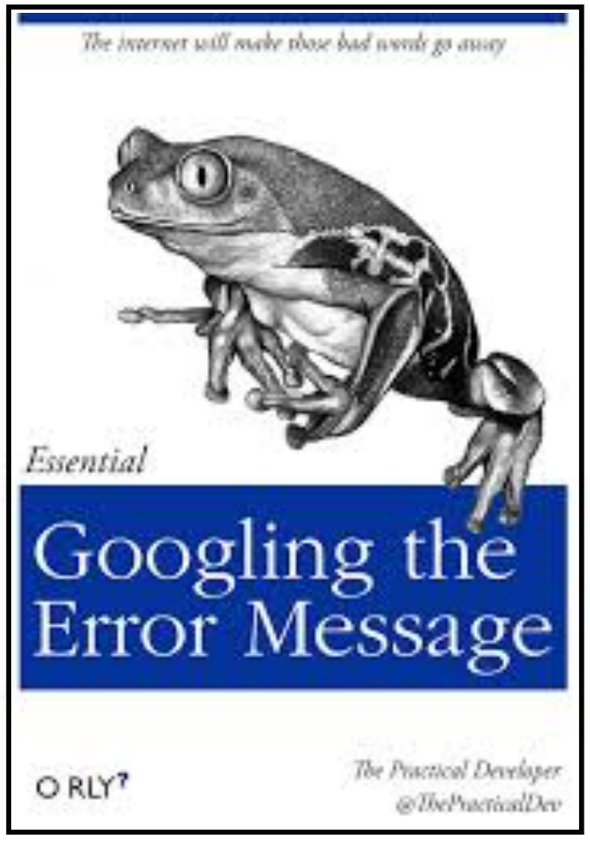


4. Getting help

(besides office hours, asking friends, etc.)

Getting help

Many online tutorials exist for Python, and websites such as Stack Overflow have tons of answered questions. *Googling the error message* is also helpful!

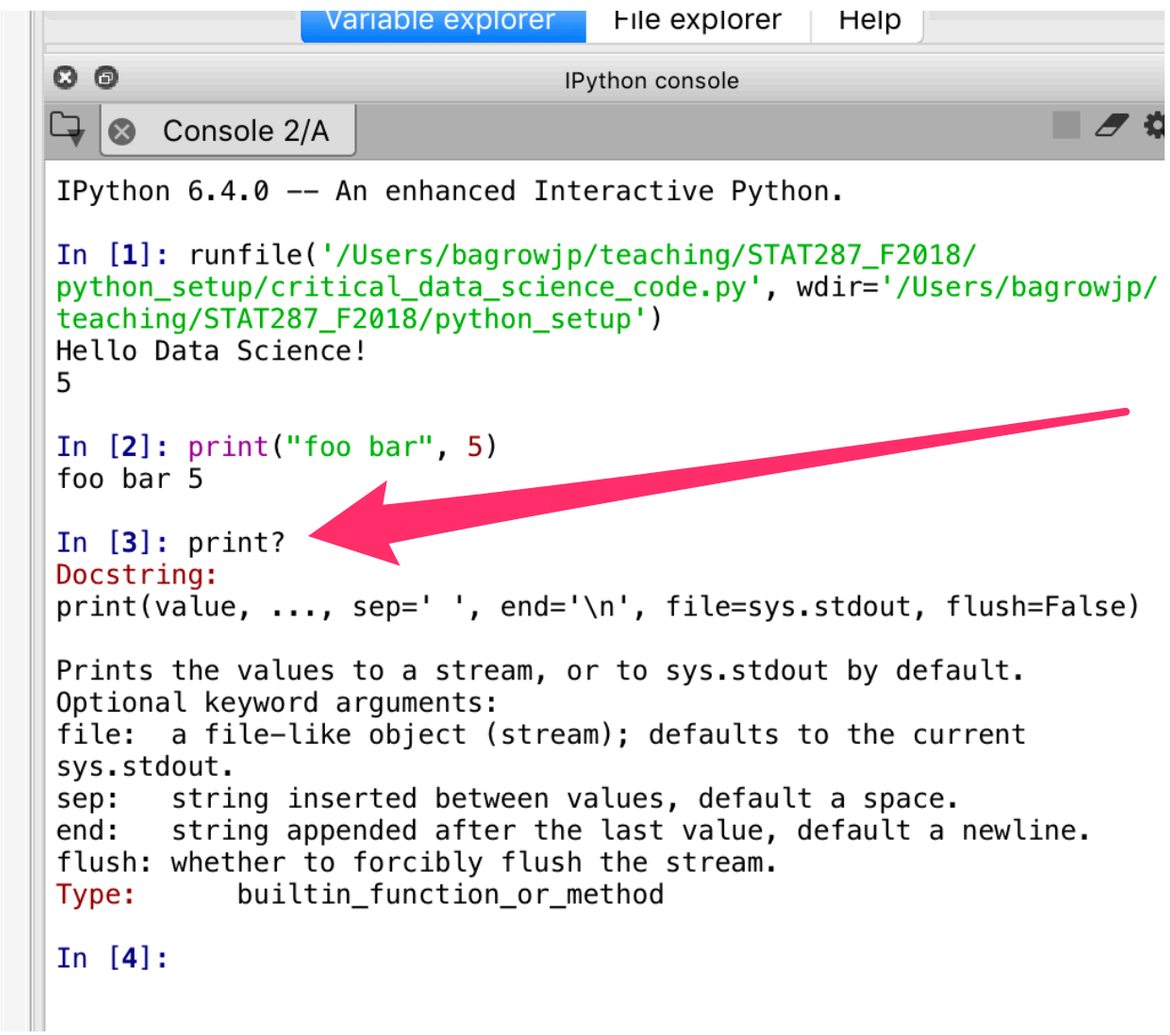


- Put only the relevant part of the error message into Google
- Remove local information, like the name of your laptop (for example)
- Use quotes "" to search for exact strings as needed

Getting help

Many online tutorials exist for Python, and websites such as Stack Overflow have tons of answered questions. *Googling the error message* is also helpful!

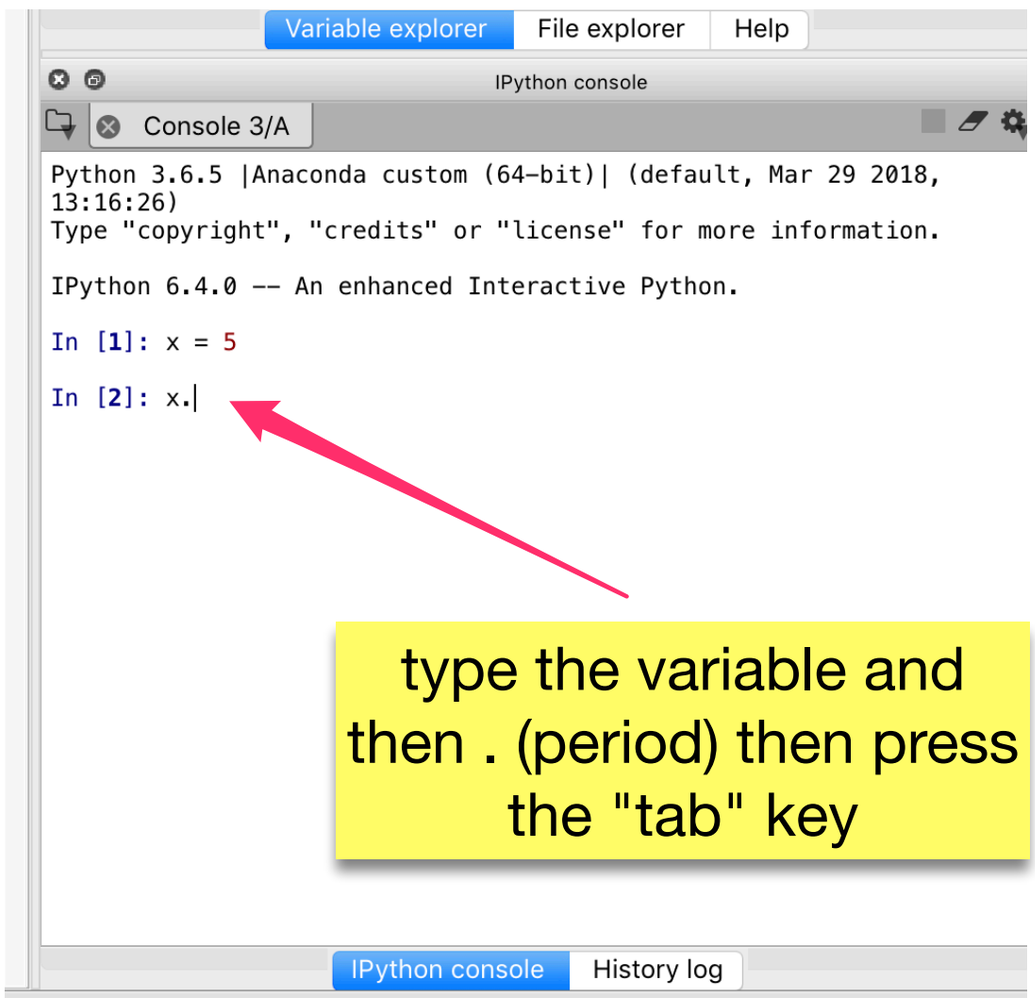
The IPython console provides documentation access as well:



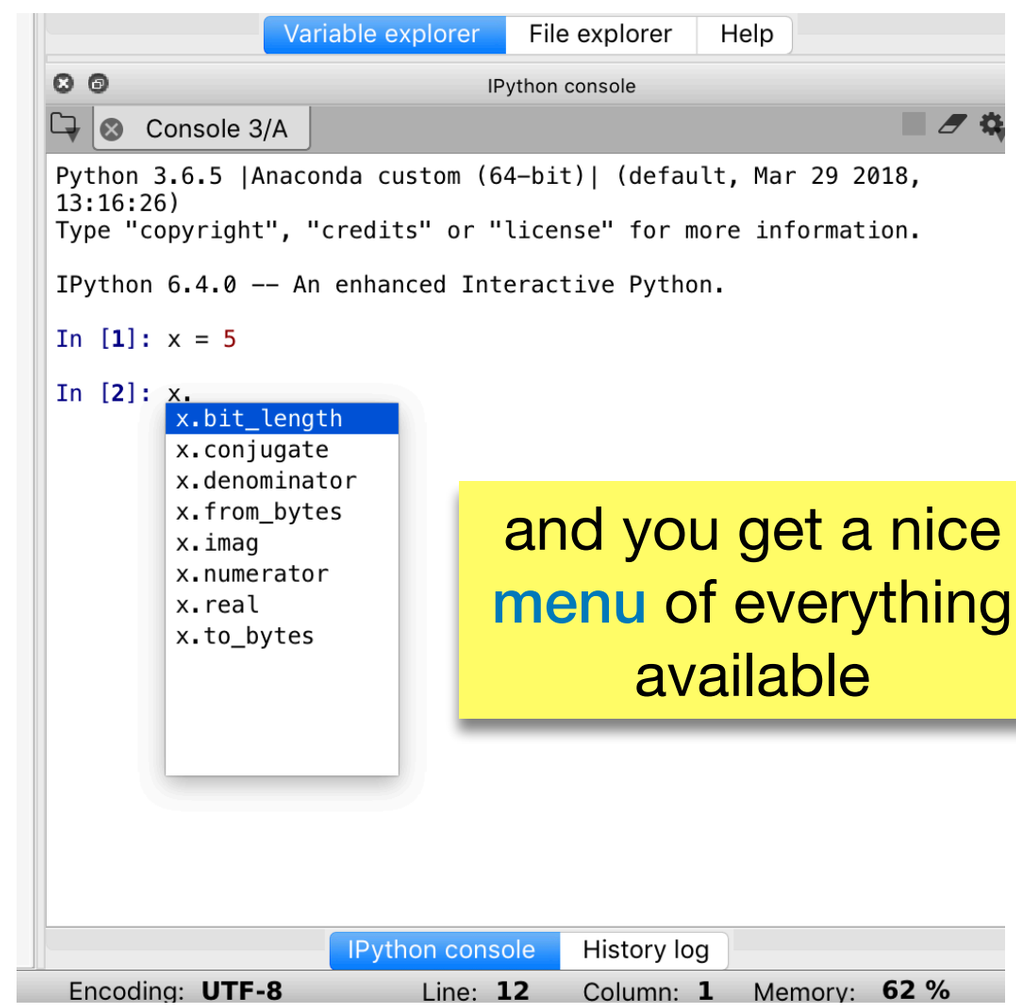
Just put a "?" after a variable, method, or function (ex: `print?`) and the **docstring** will be displayed

Getting help

IPython also helps you explore new code with **TAB COMPLETION**



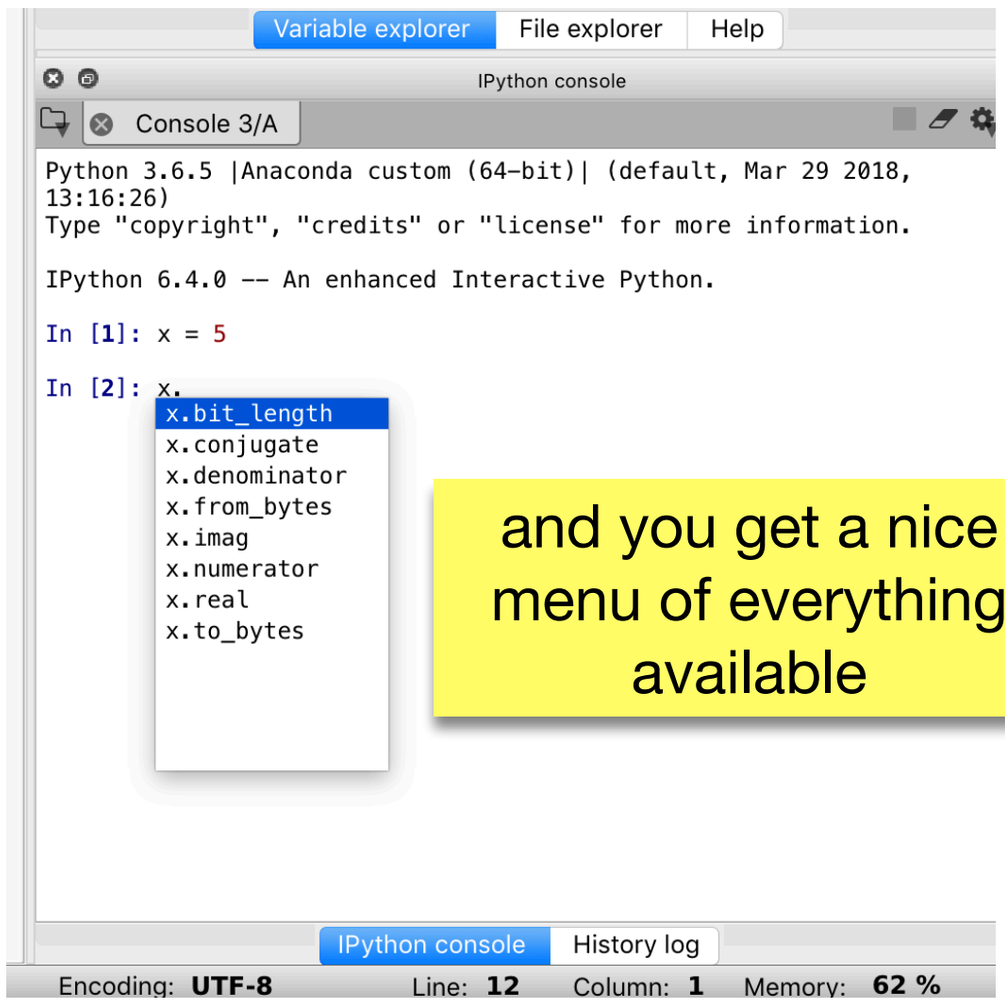
type the variable and then . (period) then press the "tab" key



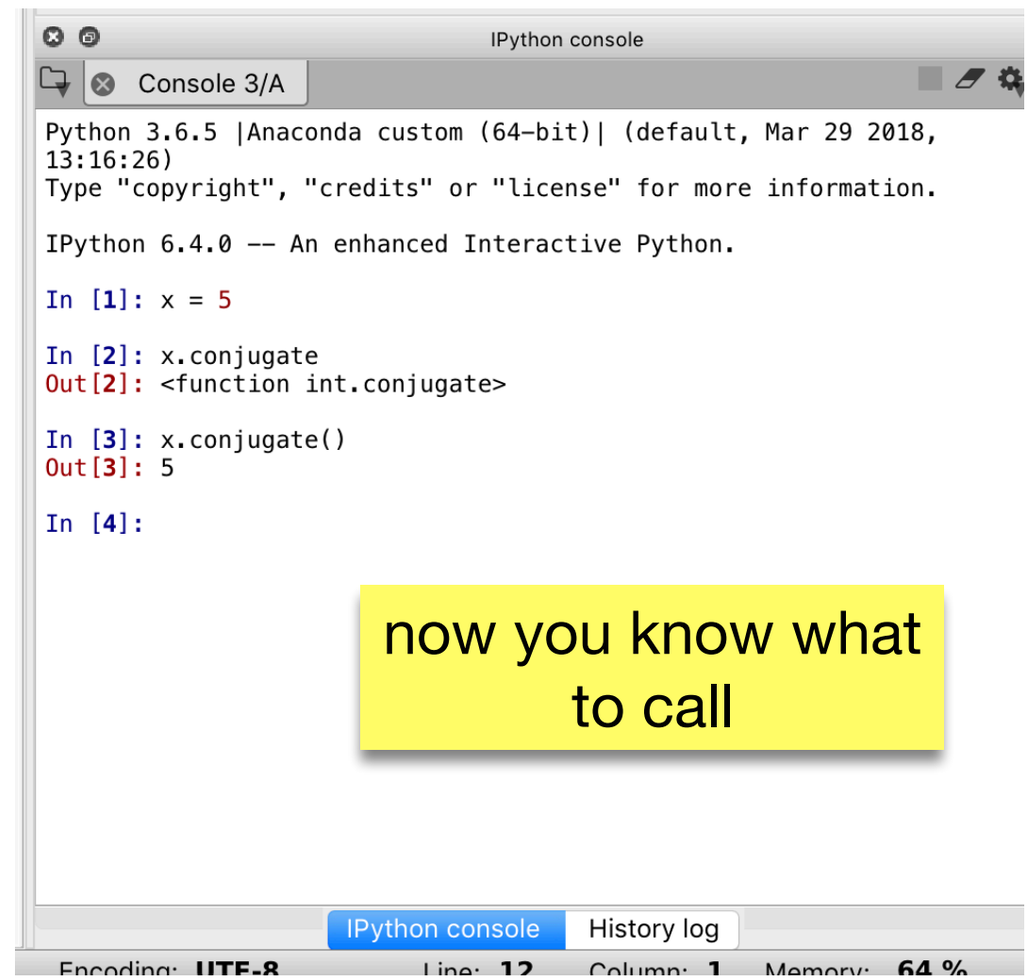
and you get a nice menu of everything available

Getting help

IPython also helps you explore new code with **TAB COMPLETION**



and you get a nice menu of everything available

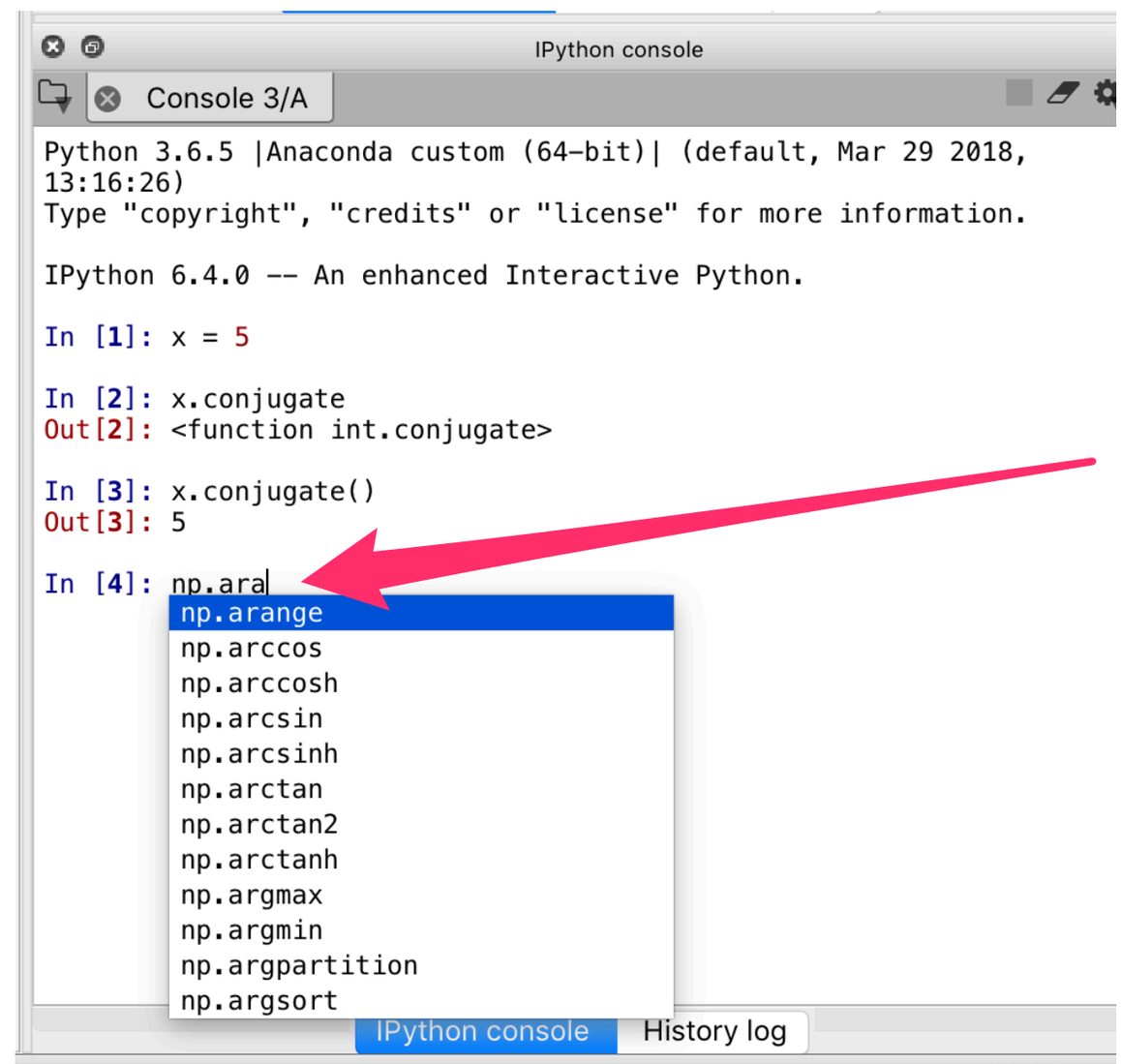


now you know what to call

Getting help

IPython also helps you explore new code with **TAB COMPLETION**

Sometimes there are very big completion menus—narrow your search by typing the first few letters of what you are looking for, either before or after you press "tab", and you will jump right there:



And: press "tab" a second time to put in the selected completion automatically (saves typing time for long names)

Appendix: set up your source code editor

If you really want to "roll your own" work environment for the course, **and have heeded the warning about being on your own to support it**, then you need to configure your text editor when working with Python code you will submit for grading.

Whitespace is a significant component of Python source code, with indentation defining blocks. One can indent a line of code by inserting spaces (using space bar) or tabs (using tab key). Python 3 will not allow mixtures of tabs and spaces.

—Required—

For all code in this course, configure your text editor so that pressing the tab key once is exactly the same as pressing the spacebar four times.

- Using spaces for tabs is known as "hard tabs"
- How to set up 4-space hard tabs depends on what editor you are using, which *you will need to figure out*
- Submit only code with 4-space indents, do not submit code containing tab characters

Confused? Don't worry, just use Spyder—it's all set up!