PYTHON FOR OCEAN SCIENCES

INSTALLATION, SET UP & PACKAGE MANAGEMENT USING ANACONDA

Nia Jones
niajones@bangor.ac.uk

Why?

- Appetite for learning Python at SOS
- Create a Python community to facilitate learning, support and using the language for research.
- Building on previous meetings in SOS

WHY PYTHON?

- Proprietary, **closed source** software ($$$)
- Interpreted language
- Popular in **academic, scientific and engineering** circles
- Strong at **signal and image processing**

- Free and **open-source**
- Large **community support**
- Interpreted language
- Uses libraries
- General purpose
- Strong for **Big Data**
- Simple syntax

- Free and **open-source**
- Uses libraries/packages
- Slower Runtime
- Strong in **statistical analysis**
- Strong in **visualisation**
- Complex syntax

Paul Romer

Jupyter, Mathematica, and the Future of the Research Paper

Fri, Apr 13, 2018

t Newsletters *The Atlantic*

SCIENCE

THE SCIENTIFIC PAPER IS OBSOLETE

Here's what's next.

By James Somers

APRIL 5, 2018 SHARE

"It's incalculable, literally ... how much is lost, and how much time is wasted, and how many results are misinterpreted or are misrepresented."

Theodore Gray, co-founder of Wolfram Research, on the cons of a traditional research paper compared to a '*computational essay*'.

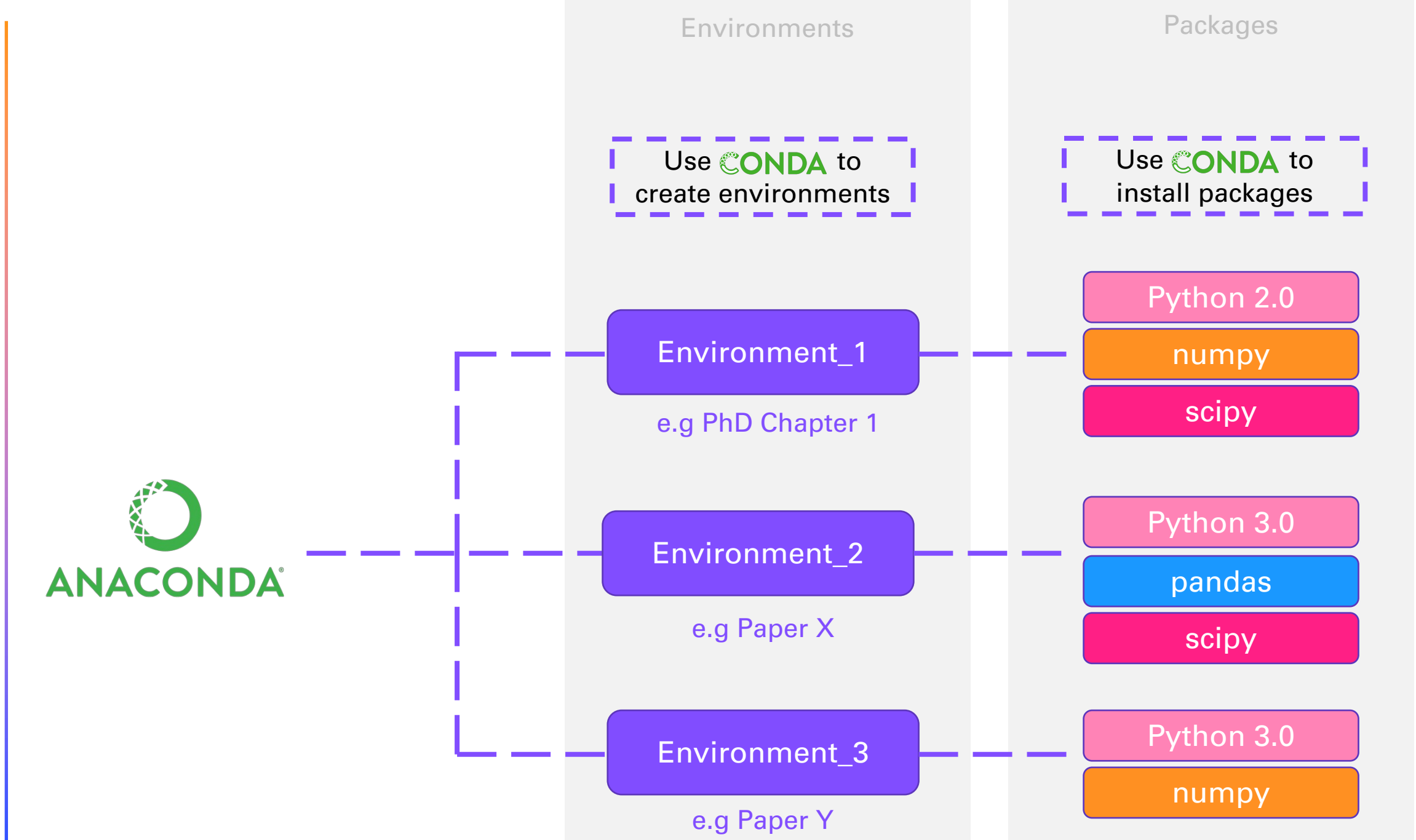
(AN) ARCHITECTURE FOR DATA SCIENCE USING PYTHON

- **Python** is the programming language 

- **Anaconda** is a ‘distribution’ for data science specific tools.

This is all you need to download onto your computer.

- **Conda** is the package and environment manager within Anaconda. 


Environments
Packages
Use CONDA to create environments
Use CONDA to install packages
ANACONDA
Environment_1
e.g PhD Chapter 1
Python 2.0
numpy
scipy
Environment_2
e.g Paper X
Python 3.0
pandas
scipy
Environment_3
e.g Paper Y
Python 3.0
numpy

DOWNLOADING ANACONDA

Download the graphical installer here: https://www.anaconda.com/products/distribution

Windows Documentation: https://docs.anaconda.com/anaconda/install/windows/

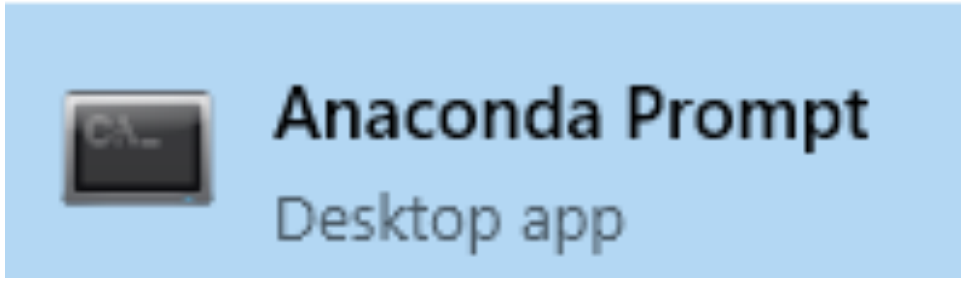
MacOS Documentation: https://docs.anaconda.com/anaconda/install/mac-os/

* You need **minimum 5 GB disk space** to download and install Anaconda. A smaller distribution, '**Miniconda**', is available but you'll have to do a lot more of the set up yourself.

CONDA – Access through the command line

Windows

On windows, ‘Anaconda Prompt’ should have been installed within your Anaconda Distribution. This is how you access ‘conda’.



Mac

On a mac, navigate to the ‘Terminal App’. This is how you access ’conda’.

CONDA – Access through the command line

To verify your conda is working type the following into the command line:

```
conda --version    # Conda displays the number of the version installed.
```

Now let's make our own 'environment' called 'python_tutorials'

```
conda create --name python_tutorials    # Create new environment
```

Get a list of all your python environments printed to the command window

```
conda info --envs     # List the environments
```

CONDA – Access through the command line

To start working within your environment you first have to ‘activate’ it.

```
conda activate python_tutorials   # Activate chosen environment
```

In Python, there are different packages which expand Python’s basic capabilities. Think of it like your data science toolbox. To list these packages type in the following:

```
conda list   # list all of the packages within the activated environment
```

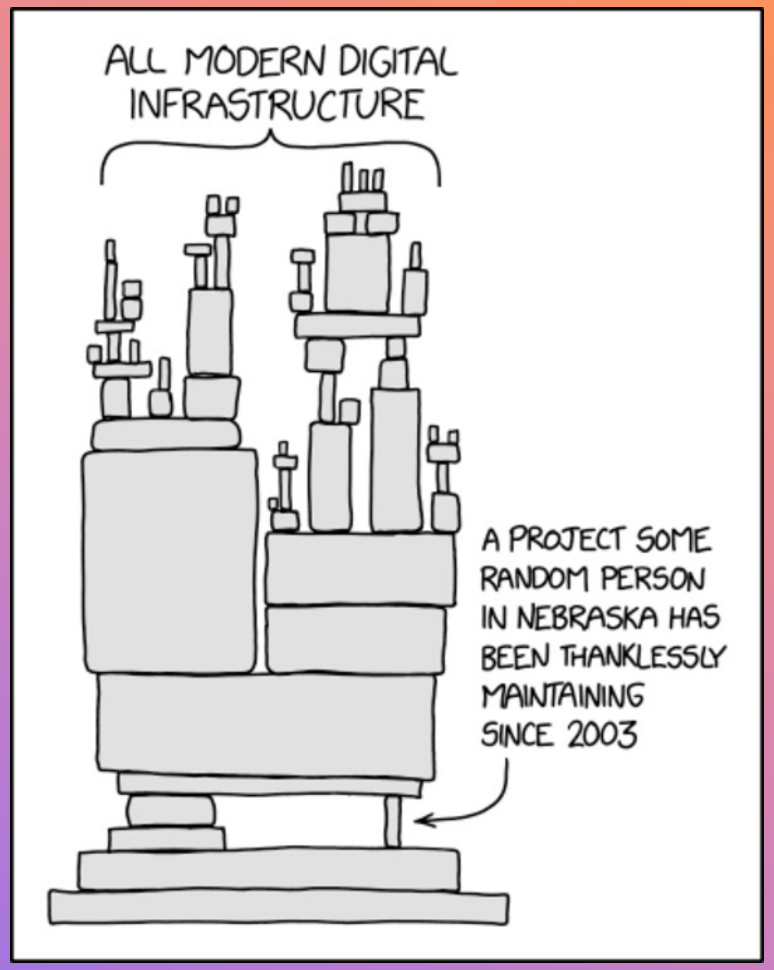
Oh no! There are no tools in the toolbox. We’ll have to download and install some packages. But first, a few words on python packages and why conda is so valuable…

PYTHON PACKAGES AND THE NINE CIRCLES OF DEPENDENCY HELL.

Dependency hell
a colloquial term for the frustration of some software users who have installed software packages which have dependencies on specific versions of other software packages.

Python is largely ‘package’ driven. Packages depend on other packages which depend on other packages *ad infinitum.* If one of those packages changes, or isn’t accounted for in an update, the whole system could break.

To overcome this, conda has a dependency resolver which identifies additional packages which need installing/updating. So **congratulations**, you may have (mostly) avoided dependency hell.



PYTHON PACKAGES AND THE NINE CIRCLES OF DEPENDENCY HELL.

Some popular python packages for data science...

NumPy	Array based data
Pandas	Labelled and heterogeneous data
SciPy	Scientific computing
Matplotlib	Visualisations
Scikit Learn	Machine Learning

CONDA – Access through the command line

Install some of these packages to your activated environment:

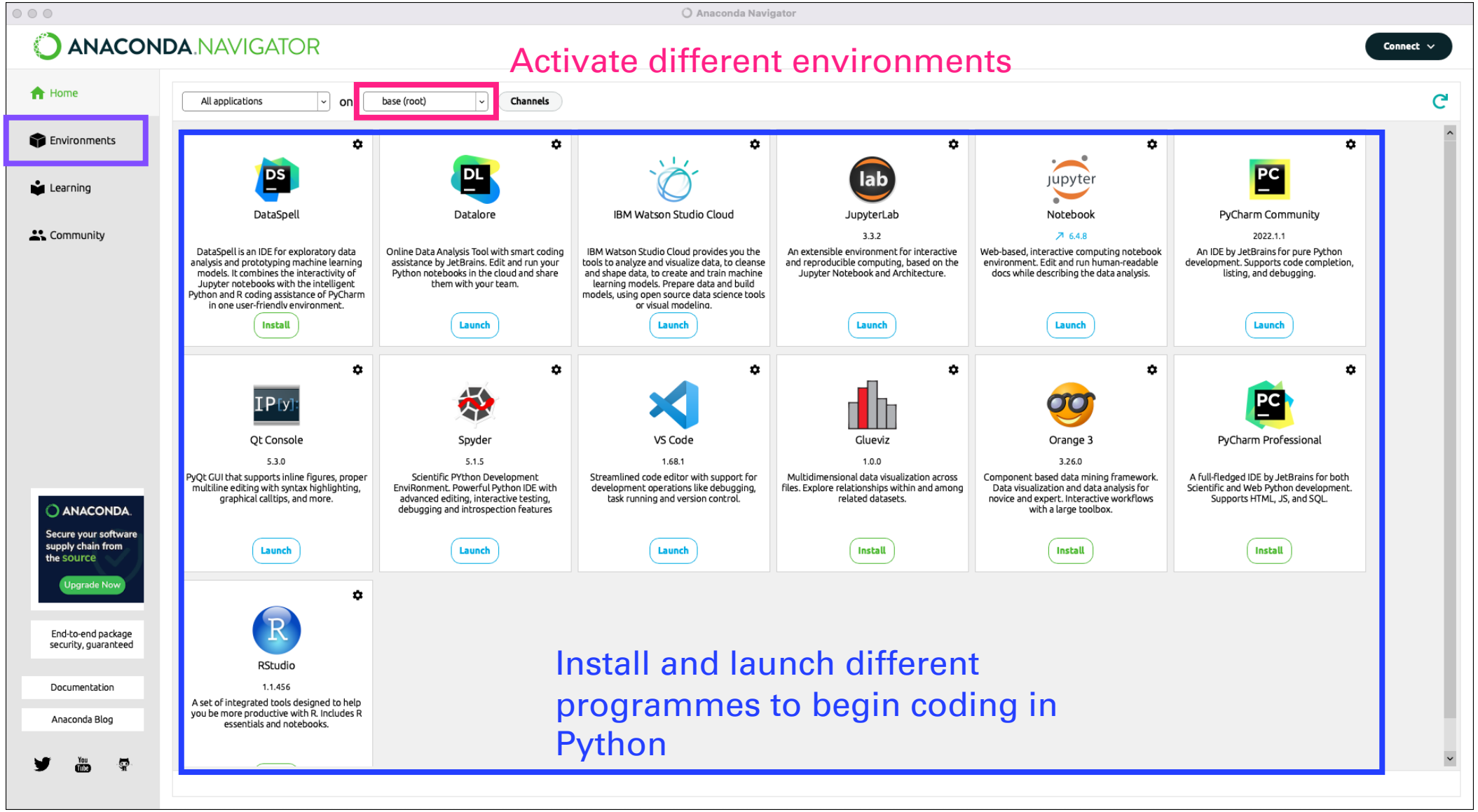
```
conda install numpy pandas scipy matplotlib  # install 4 different packages to
                                               the activated environment.
```

Now when you list your packages you should see a list of the ones you've just installed.

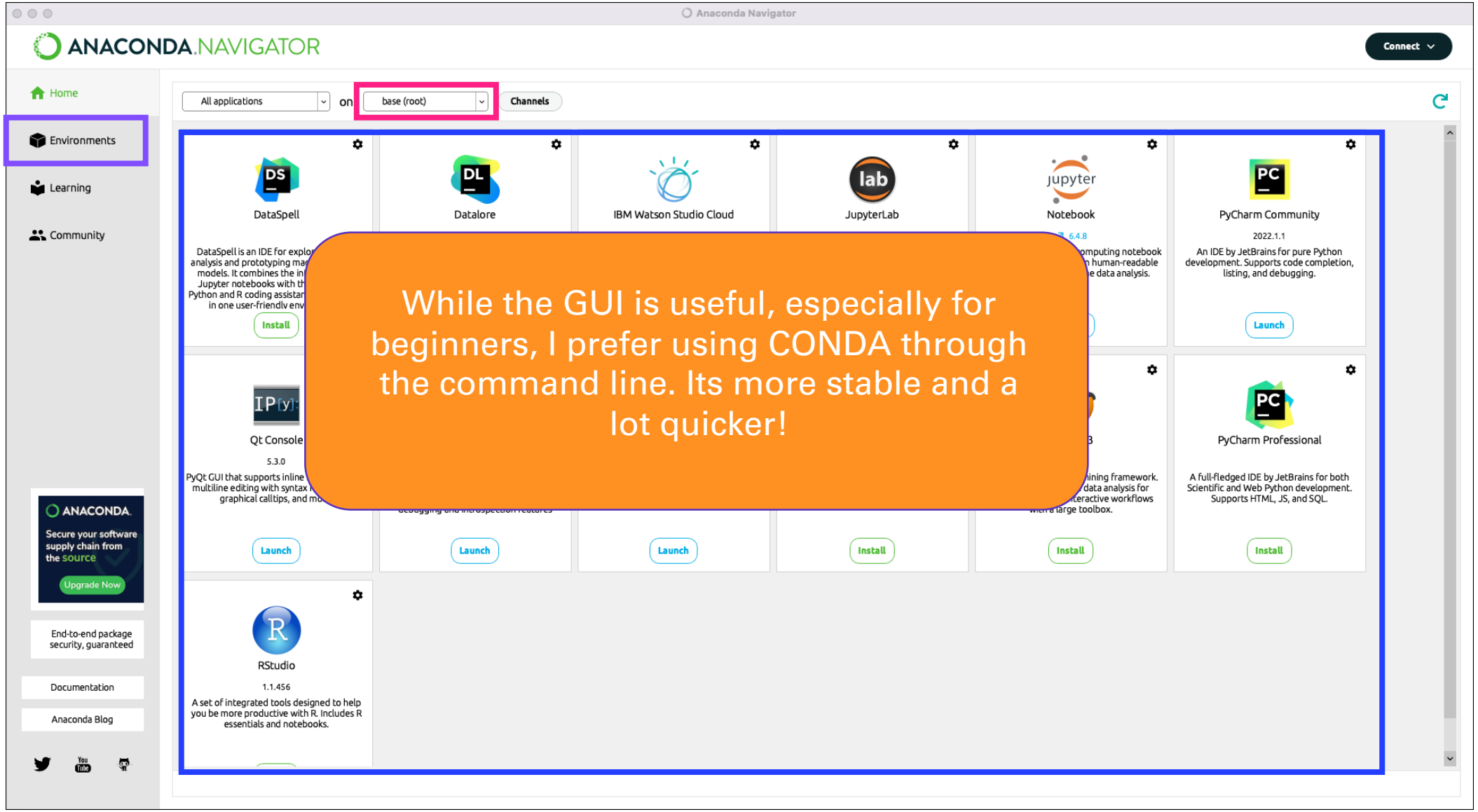
```
conda list   # list all of the packages within the activated environment
```

ANACONDA NAVIGATOR – Access through a GUI

Manage packages and environments



ANACONDA NAVIGATOR – Access through a GUI



Actually accessing and using Python.

Now you hopefully grasp the basics of what Anaconda is and have it installed on your computer, you now need something to be able to **write Python code in a meaningful way**.

There are many different software packages, or **Integrate Development Environments (IDEs),** that allow you to use Python. I'll be concentrating on two of them in these sessions: **Spyder** and **Jupyter Notebooks**.

To be able to use these, type the following two lines of code into your activated environment in your Anaconda Prompt/Terminal. You can also do this through Anaconda Navigator if you want to. (I recommend installing both for now)

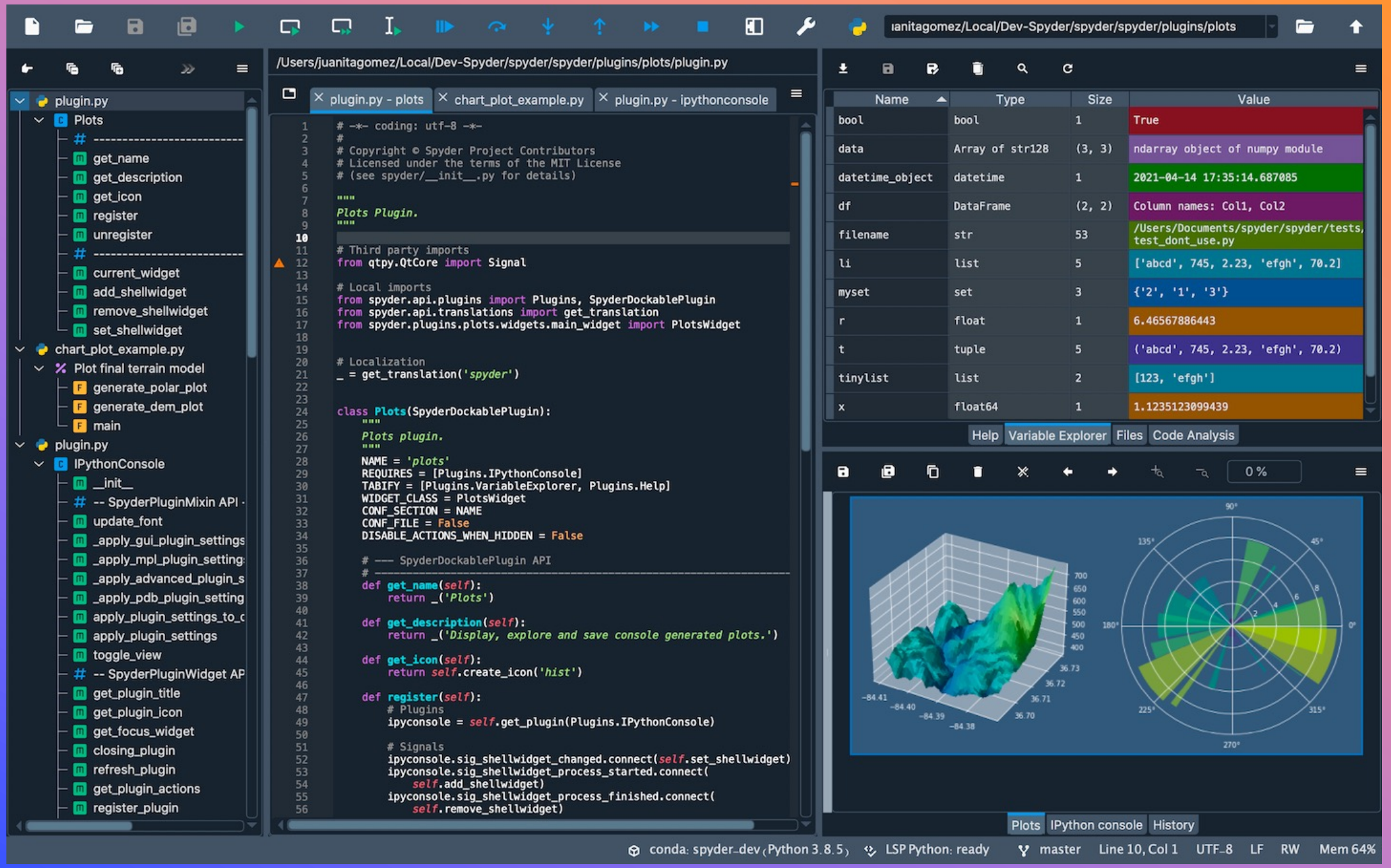
```
conda install jupyter notebook
conda install spyder
```

Actually accessing and using Python.

- More of a traditional coding environment
- Familiar layout
- .py files
- Strong for exploratory data analysis and debugging.

- Notebook environment
- Streamlined experience
- .ipynb files
- Sharing friendly
- Useful for visualisation

SPYDER


/Users/juanitagomez/Local/Dev-Spyder/spyder/spyder/plugins/plots/plugin.py
plugin.py - plots
chart_plot_example.py
plugin.py - ipythonconsole
Name
Type
Size
Value
Help
Variable Explorer
Files
Code Analysis
Plots
IPython console
History
conda: spyder-dev (Python 3.8.5)
LSP Python: ready
master
Line 10, Col 1
UTF-8
LF
RW
Mem 64%

JUPYTER NOTEBOOK

jupyter tutorial Last Checkpoint: 3 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

PyCon 2018: Using pandas for Better (and Worse) Data Science

GitHub: https://github.com/justmarkham/pycon-2018-tutorial

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
pd.__version__
```

Out[1]: '0.24.1'

Dataset: Stanford Open Policing Project (video)

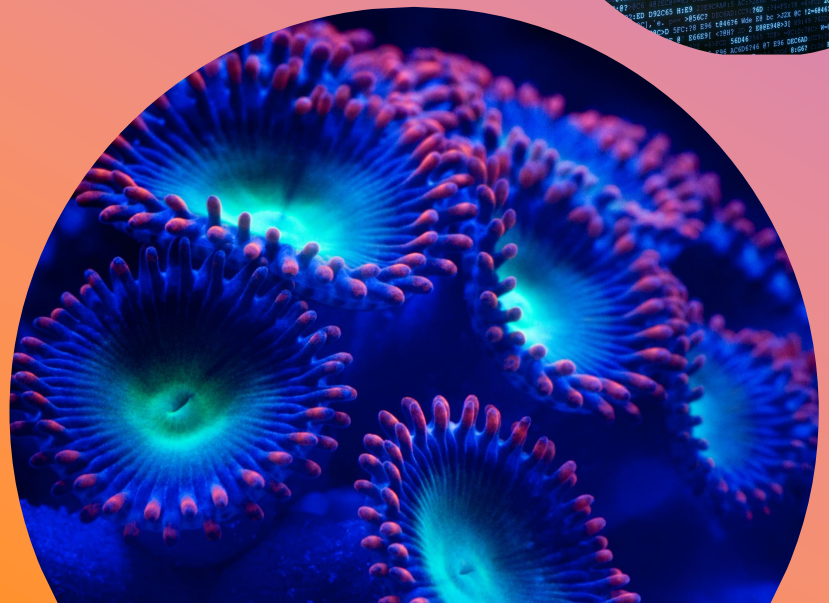
In [2]:
```
# ri stands for Rhode Island
ri = pd.read_csv('police.csv')
```

In [3]:
```
# what does each row represent?
ri.head()
```

Out[3]:

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	

PYTHON FOR OCEAN SCIENCES



NEXT SEMINAR:

JANUARY 23RD

THE BASICS AND IMPORTING DATA