

Getting started with python on Pegasus

Ray Bell

Lunch Bytes. Fall 2018



```
$ git clone https://github.com/milancurcic/lunch-bytes.git  
$ cd lunch-bytes/Fall_2018/LB14
```

Installing python on pegasus

- Setup folders:

```
$ mkdir -p src local/bin
```

```
$ cd src
```

- Install miniconda into your directory <https://conda.io/miniconda.html>

```
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86\_64.sh
```

```
$ chmod +x Miniconda3-latest-Linux-x86_64.sh
```

```
$ ./Miniconda3-latest-Linux-x86_64.sh [-u]
```

Install in /nethome/USERNAME/local/bin/miniconda3

Creating an environment file

On laptop

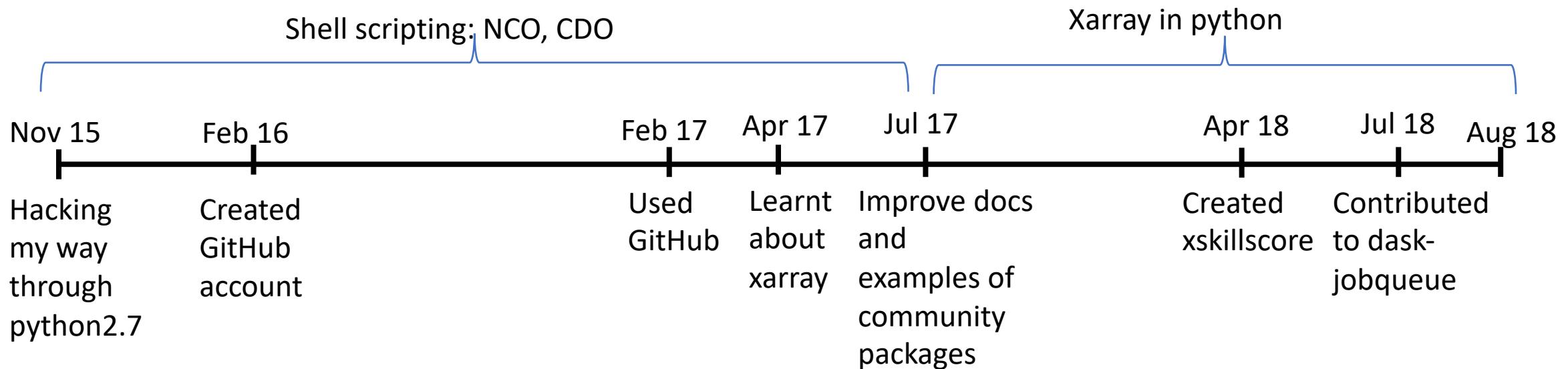
```
$ conda env create -f py_intro_lb_env_file.yml  
$ conda activate py_intro_lb_env
```

On Pegasus (can take up to 20 mins...)

```
$ bsub < submit_create_env.sh  
$ bjobs  
$ bpeek  
$ conda info --envs # after job finished. Should see py_intro_lb_env  
$ conda activate py_intro_lb_env  
[Note: when I tested this morn dask-jobqueue didn't install  
$ pip install git+https://github.com/dask/dask-jobqueue]
```

About me

- Started as a post-doc in November 2015. Previously used IDL and Matlab.
- Had a couple of options: NCL or Python.
- Leo Siqueira used python and could help me!



Pegasus

- See seminar by Natalie Perlin on September 14th

What is python?

- Created by Guido Van Rossum in 1991
- Open source (free!)
- Object orientated (procedures can access and modify them ‘self’). E.g.



Rolling seasonal average of monthly data:

```
pd.DataFrame({'A': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]}).rolling(3).mean()
```

Python package
(pandas).
Import pandas as pd

How to create
data/array in
pandas

Rolling operator
over 3 values

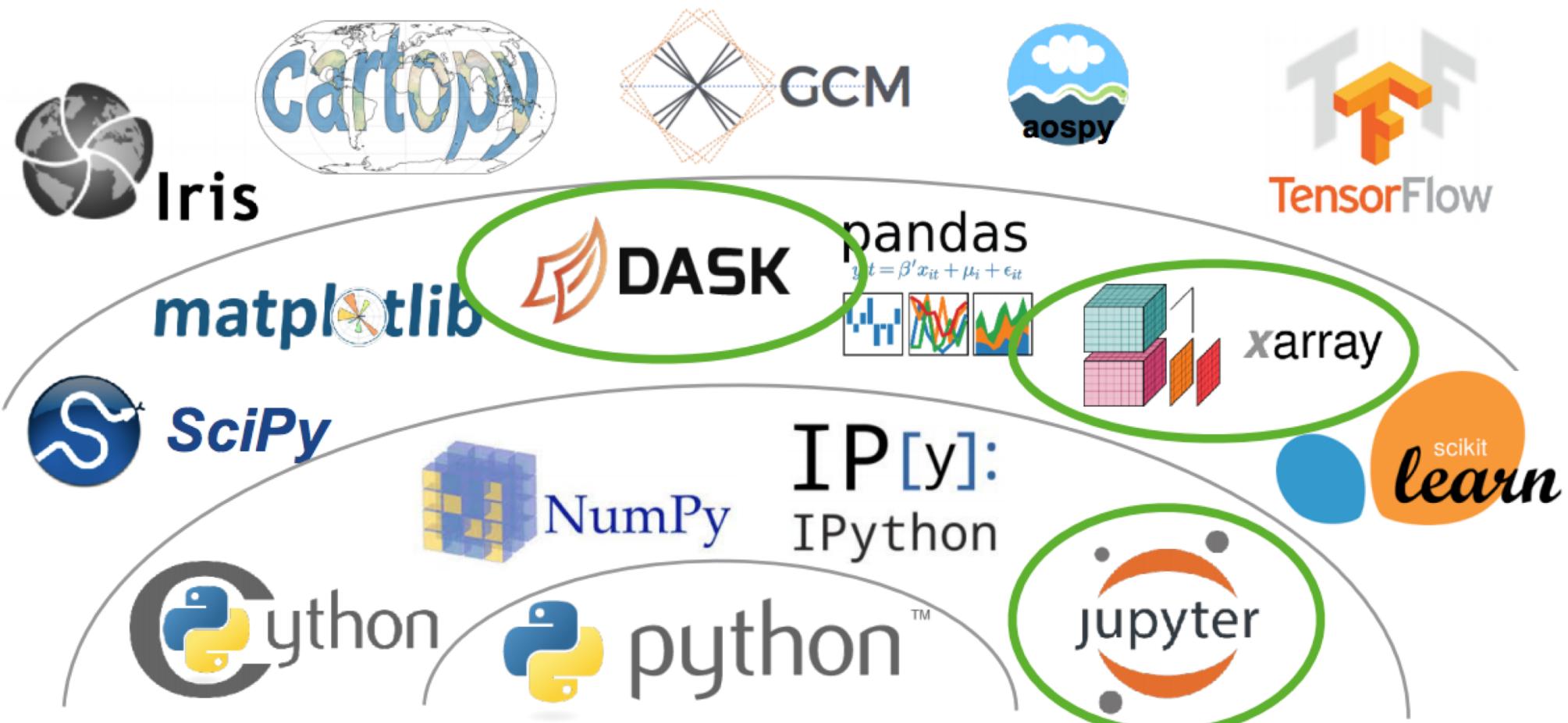
Take the mean

What is python?

- Funding from DARPA (Defense Advanced Research Projects Agency) in ~2000. Proposal entitled ‘Computer Programming for Everybody’
<https://www.python.org/doc/essays/cp4e/>
- Python Software Foundation ‘board members’ and community effort work on improving python. <https://devguide.python.org/>
- Agree on PEPs (Python Enhancement Proposals)
<https://www.python.org/dev/peps/>
- NumFOCUS provide support for some packages e.g. numpy. Matplotlib
- PyCon, SciPy, pyladies

What is python?

SCIENTIFIC PYTHON FOR DATA SCIENCE

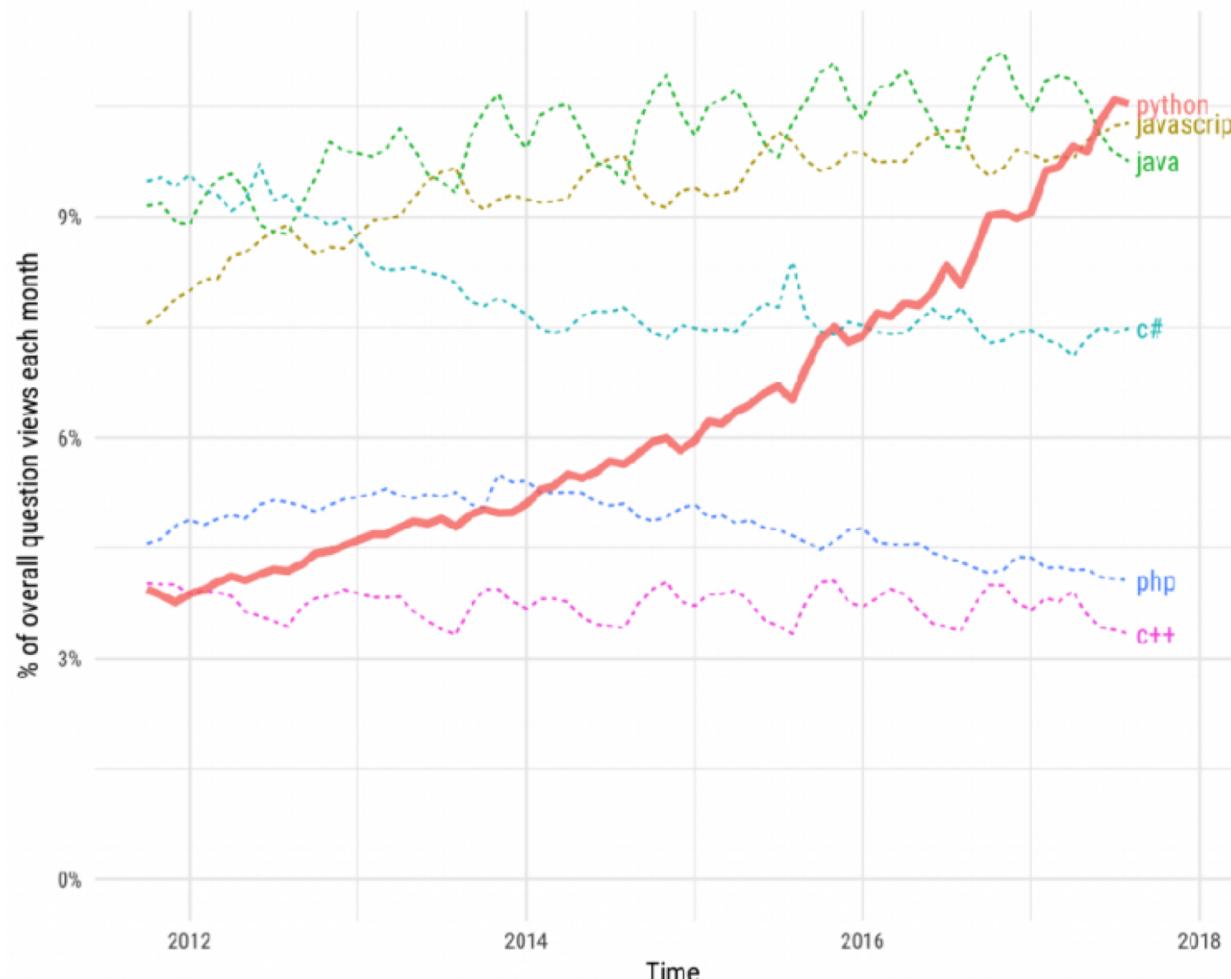


Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

Why python?

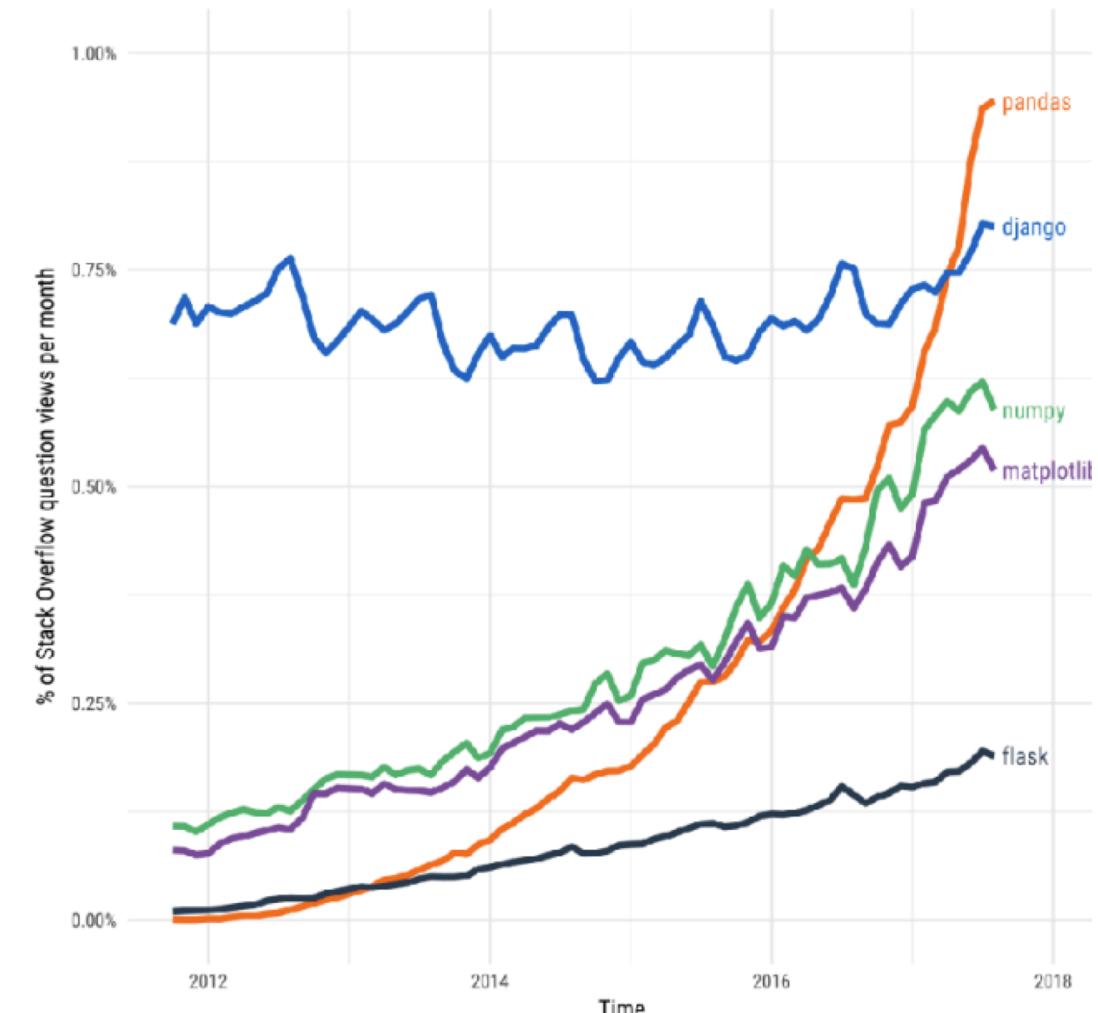
Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Stack Overflow Traffic to Questions About Selected Python Packages

Based on visits to Stack Overflow questions from World Bank high-income countries



source: stackoverflow.com

Why python

- Large uptake by the geoscientific community e.g. unidata workshops (<https://www.unidata.ucar.edu/events/>) and ECMWF conference (<https://www.ecmwf.int/en/learning/workshops/2018-workshop-developing-python-frameworks-earth-system-sciences>)
- Easy to stay on top of developing packages.
- ~150,000 packages (<https://pypi.org/>)
- Industry applications.

Python resources

- RSMAS python list serv. Send an e-mail to python-subscribe@lists.rsmas.miami.edu
- Stackoverflow <https://stackoverflow.com/questions/tagged/python> and how to ask questions see <https://stackoverflow.com/help/mcve>
- GitHub
- Python people at RSMAS: Milan, Leo, Tiago, Suvy, Brian
- Text editors: <https://atom.io/>
<https://pythonhosted.org/spyder/editor.html>

Overview

- Install miniconda
- Python
- ipython
- SSH tunneling for jupyter notebooks
- Dask-jobqueue for parallel jobs

Python strings

<http://swcarpentry.github.io/python-novice-inflammation/>

Other resources?

```
$ python  
>>> print('Hello RSMAS!')  
>>> word = 'lead'
```

Indexing starts at 0

```
>>> word[0]
```

What happens when you type >>> word[4]?

Python for loops

Use 4 spaces for indentation:

```
>>> for char in word:  
...     print(char)
```

Create a sequence of numbers

```
>>> for i in range([start],4):  
...     print(i)
```

Python basic math

```
>>> 1 + 2
```

```
>>> 1 / 2
```

```
>>> 2 * 3
```

```
>>> 2 ** 3
```

Python scripts

Add to mypyscript.py:

```
mylist = [10, 7, 5, 18]
for i, val in enumerate(mylist):
    print(i, val)
```

```
$ python mypyscript.py
```

Python functions

Add to func_script.py:

```
def fib(n):
    """ Return Fibonacci series up to n """
    result = []
    a, b = 0, 1
    while a < n:
        result.append(a)
        a, b = b, a + b
    return result
print(fib(10))
```

```
$ python func_script.py
```

Or

```
$ python
>>> import func_script as fs
>>> fs.fib(10)
```

ipython

- Interactive python. Works best on a notebook on laptop.
- Has useful ‘magic words’
<https://ipython.readthedocs.io/en/stable/interactive/magics.html>

```
$ ipython
```

```
In [1]: %time sum(range(1000000))
```

```
In [1]: alpha = 123
```

```
In [2]: beta = 'test'
```

```
In [3]: %whos
```

```
In [1]: %run myscript.py
```

ipython

Add to buggy_script.py:

```
mylist = [10, 7, 5, 18]
for i, val in enumerate(mylist):
    print(i, v)
```

```
%run buggy_script.py
```

<https://docs.python.org/3.6/tutorial/errors.html>

```
%debug ['u' for up Traceback and 'd' for down Traceback]
```

environments

- How not to fudge up your work space.
- <https://conda.io/docs/user-guide/tasks/manage-environments.html>

```
$ conda create --name test_env python=3.6
```

```
[$ conda remove --name test_env --all]
```

```
[$ conda info --envs]
```

```
$ conda activate test_env
```

Installing packages

- Quick way: \$ pip install PACKAGE
- Get the bleeding edge version: \$ pip install git+https://github.com/pydata/xarray.git
- Recommended to use conda / conda forge <https://conda-forge.org/>
\$ conda install -c conda-forge notebook

Plotting

Add to plotting_script.py

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.savefig('test.png')
```

```
$ python plotting_script.py
```

Jupyter notebook for plotting

- See seminar by Suyv next week

Configuring the notebook

- http://pangeo.io/setup_guides/hpc.html#configure-jupyter

```
$ jupyter notebook --generate-config
```

This creates `~/.jupyter/jupyter_notebook_config.py`

```
$ ipython
```

```
In [1]: from notebook.auth import passwd; passwd()
```

Copy the password hash to the `c.NotebookApp.password` line in
`~/.jupyter/jupyter_notebook_config.py`

```
$ chmod 400 ~/.jupyter/jupyter_notebook_config.py
```

Tunneling the notebook

```
$ jupyter notebook --no-browser --ip=10.141.226.234 --port=8888  
[$ jupyter notebook --no-browser]
```

On my laptop open a terminal and do

```
$ ssh -N -L 8888:10.141.226.234:8888 USERNAME@pegasus.ccs.miami.edu  
[$ ssh -N -L 8888:localhost:PORTNUMBER  
USERNAME@pegasus.ccs.miami.edu ]
```

Open <http://localhost:8888> on laptop

New -> Notebook: Python 3

Jupyter notebook for plotting

- See https://github.com/milancurcic/lunch-bytes/blob/master/py_intro/py_intro_lb_nb_plotting.ipynb

The examples use:

<https://matplotlib.org/>

<https://pandas.pydata.org/pandas-docs/stable/visualization.html>

<https://seaborn.pydata.org/>

<https://scitools.org.uk/cartopy/docs/latest/>

<http://xarray.pydata.org/en/stable/plotting.html>

MyBinder

- See <https://github.com/raybellwaves/SSElFaro>

<https://mybinder.org/>

Packages for interactive plotting:

<https://bokeh.pydata.org/en/latest/>

<http://geo.holoviews.org/>

<https://altair-viz.github.io/>

<https://ipyleaflet.readthedocs.io/en/latest/>

<http://python-visualization.github.io/folium/>

Submitting parallel jobs using dask-jobqueue

- See https://github.com/milancurcic/lunch-bytes/blob/master/py_intro/py_intro_lb_nb_parallel.ipynb

<https://dask-jobqueue.readthedocs.io/en/latest/>

<https://distributed.readthedocs.io/en/latest/>

To find the IP address to tunnel

\$ python

```
>>> from dask_jobqueue import LSFCluster
>>> from dask.distributed import Client
>>> cluster = LSFCluster(cores=2, memory='4 GB')
>>> client = Client(cluster)
>>> print(client)
```