Introduction to anaconda

Manage your programming environments

Managing environments

Environments are directories that contain a specific collection of packages that you have installed. For example you could have one with NumPy 1.7 and a different one with NumPy 1.6

They can be loaded and unloaded for different applications

Anaconda, pip and module are all examples of package managers

- Can be used to produce requirements.txt of all your packages (useful for documentation!)
- Or if someone sends you a *requirements.txt, you can load the packages needed to run their code*

Installing anaconda

https://www.anaconda.com/products/individual

- Windows users: must use graphical installer
- Linux users: must use the terminal
- Mac: option to choose method you prefer
- Installing about 3GB so might take some time

Activity

1). Try **conda info** - what response to you get?

2). Find out if you already have any conda environemnts with **conda env list**

3). Create a new python 3 environment with **conda create python=3.7,** call it *learning_conda* with **--name**

4). Load your new environment using **conda activate** (use **source activate** for mac)

5). Find out if this environment have the numpy package with **conda list**

Hint: you can add words after “conda list” and it will filter the output

Activity

5). Install the *jupyter, numpy, matplotlib, pandas* packages with **conda install**

Hint: you can specify package numbers, e.g. numpy=1.7

6). Find out what version of the numpy package with **conda list**

7). Create a file containing all the packages and versions named *requirements.txt* with **conda env create --file**

8). Open a **jupyter notebook**

9). Close the jupyter notebook with **ctrl+c**