Using Conda for Python Data Science projects

Outline of Presentation

- What is (Ana)conda?
- What is a ‘package manager’?
  - How can we install all of the packages that we need?
- What is a conda environment?
  - Why do we need to use environments?
- How does conda differ from other package managers?
- Live demonstration of installation procedure – with command line tutorial

Anaconda distribution

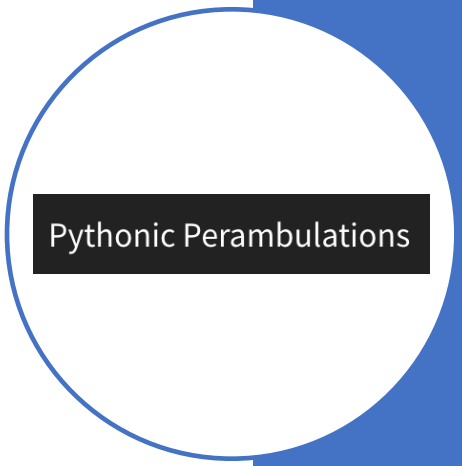
- https://www.anaconda.com/
- Full data science platform for both individuals and enterprise teams
- Not just focused on Python
  - Also uses R
  - But Python is the topic of this presentation
- How I think of it:
  - A framework to install all the software I need for projects
  - How to keep this isolated from other software dependencies

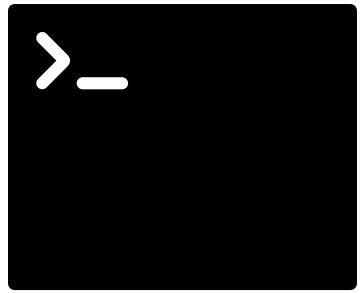
Full version vs miniconda

- Two main versions:
  - Anaconda
  - Miniconda
- Anaconda includes a full GUI called the ‘Navigator’
  - Use a point and click interface to create environments and install packages
  - Easy to learn – but a large install
- Miniconda
  - Everything done via the command line
  - *Much* smaller installation
  - Most online material will always use command line solutions

Blog article by Jake VanderPlas – Pythonic Perambulations

- **Conda: Myths and Misconceptions**
  - https://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/
- Ana/Mini-conda is a **Distribution**
  - **“**A *software distribution* is a pre-built and pre-configured collection of packages that can be installed and used on a system”
- Conda is a **Package-Manager**
  - **“**A *package manager* is a tool that automates the process of installing, updating, and removing packages”
- If JVP wrote something – read it!





My preference - Miniconda

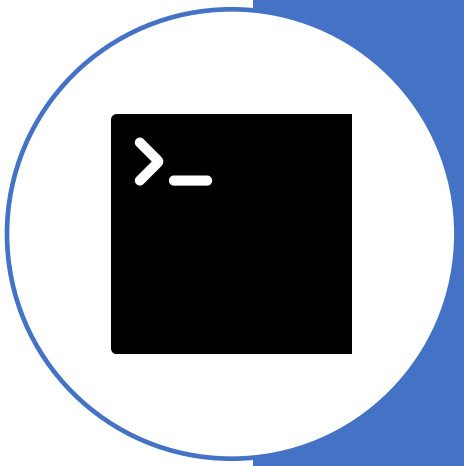
- Less bloat with the install
- Most (all?) reference articles and blog posts will make references to terminal commands
- As data scientists – you will work with code and scripts – the command line (once learned) is a fantastic tool
- But Anaconda comes with many additional packages
  - Check it out for comparison

Installing miniconda

- Go to https://docs.conda.io/en/latest/miniconda.html
- Installers for Windows, Mac OSX, and Linux
- Windows:
  - Use the graphical installer
- Mac OSX:
  - I use the command line (bash) installer
- Linux:
  - Only the command line option
  - You use Linux – you'll figure it out!

Basic terminal commands

- `cd` Change directory
  - `cd ../` Change to the ‘parent’ directory
- `~` References the ‘home’ directory
  - `ls ~` List the ‘home’ directory
  - `ls ~/Desktop` List the Desktop directory (from any location)
- `ls` List the current directory
- `mv` Move a file to a different directory
- `mkdir` Make a new directory
- `rmdir` Remove an **empty** directory
- `rm` Remove a file (it does NOT go to your recycle bin!!! 😟)
- `rm -f` ‘Force’ remove a protected file 😰
- `rm -r` ‘Recursively’ remove a directory with all sub directories be VERY careful 😱
- `rm -rf` Force remove into all sub directories – be BEYOND careful 🤯



Mac OSX installation

- First – install iTerm2
  - https://www.iterm2.com/
  - It ‘does amazing things’ 👀
- Change into directory with the miniconda installer
  - `Miniconda3-latest-MacOSX-x86_64.sh`
- Run the following, accept Terms and defaults
  - `./Miniconda3-latest-MacOSX-x86_64.sh`
- ‘ . /’ means ‘run this script *from here*’
  - Needed as the directory you are in is not in your PATH
- Of note - You can view your path by typing
  - `echo $PATH`
  - The $ sign means ‘Environment Variable’

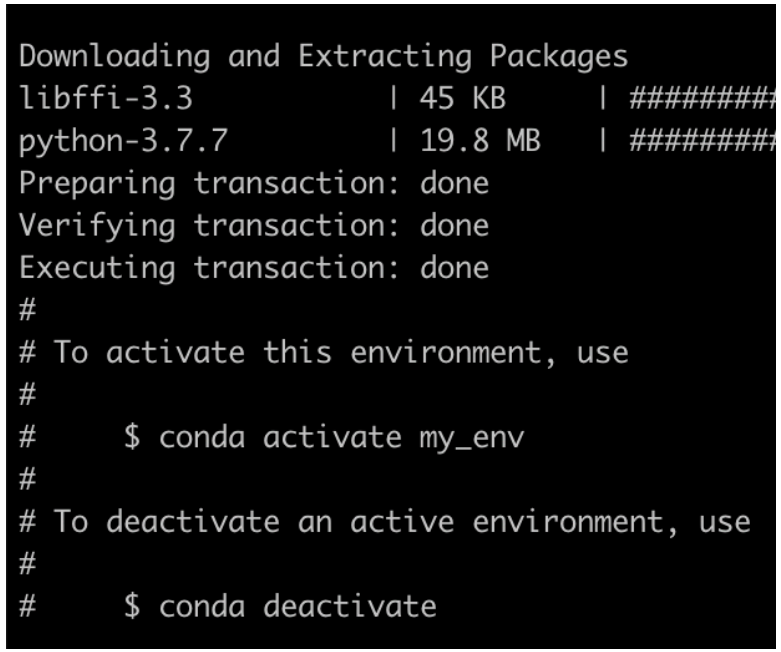
Best resource for all conda commands

- Refer to the documentation at:
  - https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

```
(base) conda_presentation % conda create --name my_env python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/tim/miniconda3/envs/my_env

  added / updated specs:
    - python=3.7


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    libffi-3.3                 |       h0a44026_1          45 KB
    python-3.7.7               |       hf48f09d_4        19.8 MB
    ------------------------------------------------------------
                                           Total:        19.9 MB

The following NEW packages will be INSTALLED:

  ca-certificates    pkgs/main/osx-64::ca-certificates-2020.1.1-0
  certifi            pkgs/main/osx-64::certifi-2020.4.5.1-py37_0
  libcxx             pkgs/main/osx-64::libcxx-4.0.1-hcfea43d_1
  libcxxabi          pkgs/main/osx-64::libcxxabi-4.0.1-hcfea43d_1
  libedit            pkgs/main/osx-64::libedit-3.1.20181209-hb402a30_0
  libffi             pkgs/main/osx-64::libffi-3.3-h0a44026_1
  ncurses            pkgs/main/osx-64::ncurses-6.2-h0a44026_1
  openssl            pkgs/main/osx-64::openssl-1.1.1g-h1de35cc_0
  pip                pkgs/main/osx-64::pip-20.0.2-py37_1
  python             pkgs/main/osx-64::python-3.7.7-hf48f09d_4
  readline           pkgs/main/osx-64::readline-8.0-h1de35cc_0
  setuptools         pkgs/main/osx-64::setuptools-46.1.3-py37_0
  sqlite             pkgs/main/osx-64::sqlite-3.31.1-h5c1f38d_1
  tk                 pkgs/main/osx-64::tk-8.6.8-ha441bb4_0
  wheel              pkgs/main/osx-64::wheel-0.34.2-py37_0
  xz                 pkgs/main/osx-64::xz-5.2.5-h1de35cc_0
  zlib               pkgs/main/osx-64::zlib-1.2.11-h1de35cc_3


Proceed ([y]/n)?
```

Creating an environment

- A new environment is made by using the `conda create` command
- In this, you can specify the version of Python – and the name of the environment you wish to create
- A summary of what will be installed is given to you
- To proceed – press y



Note the output screen

- This is **Very Important** for you to remember
  - You **must** move into this environment in order to install material into it
  - You **will** make this mistake periodically for the rest of your career!
- Note the following:
  - `activate`
  - `deactivate`

Location of Python Binary after activating environment

- With my terminal setup – it shows that the environment is activated
- Using the `which` command – we can now see where the Python binary is located

```
(base) conda_presentation % which python
/Users/tim/miniconda3/bin/python
(base) conda_presentation % conda activate my_env
(my_env) conda_presentation % which python
/Users/tim/miniconda3/envs/my_env/bin/python
(my_env) conda_presentation %
```

Just the minimum package

- Think of this as a fresh Python installation
- We don’t yet have what is needed for a data science project

```
(my_env) conda_presentation % python
Python 3.7.7 (default, May  6 2020, 04:59:01)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'numpy'
>>>
```

Installing numpy

This is done using the simple command:
`conda install numpy`

```
(my_env) conda_presentation % conda install numpy
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/tim/miniconda3/envs/my_env

  added / updated specs:
    - numpy


The following NEW packages will be INSTALLED:

  blas               pkgs/main/osx-64::blas-1.0-mkl
  intel-openmp       pkgs/main/osx-64::intel-openmp-2019.4-233
  libgfortran        pkgs/main/osx-64::libgfortran-3.0.1-h93005f0_2
  mkl                pkgs/main/osx-64::mkl-2019.4-233
  mkl-service        pkgs/main/osx-64::mkl-service-2.3.0-py37hfbe90
  mkl_fft            pkgs/main/osx-64::mkl_fft-1.0.15-py37h5e564d8_
  mkl_random         pkgs/main/osx-64::mkl_random-1.1.0-py37ha77172
  numpy              pkgs/main/osx-64::numpy-1.18.1-py37h7241aed_0
  numpy-base         pkgs/main/osx-64::numpy-base-1.18.1-py37h65755
  six                pkgs/main/osx-64::six-1.14.0-py37_0


Proceed ([y]/n)?
```

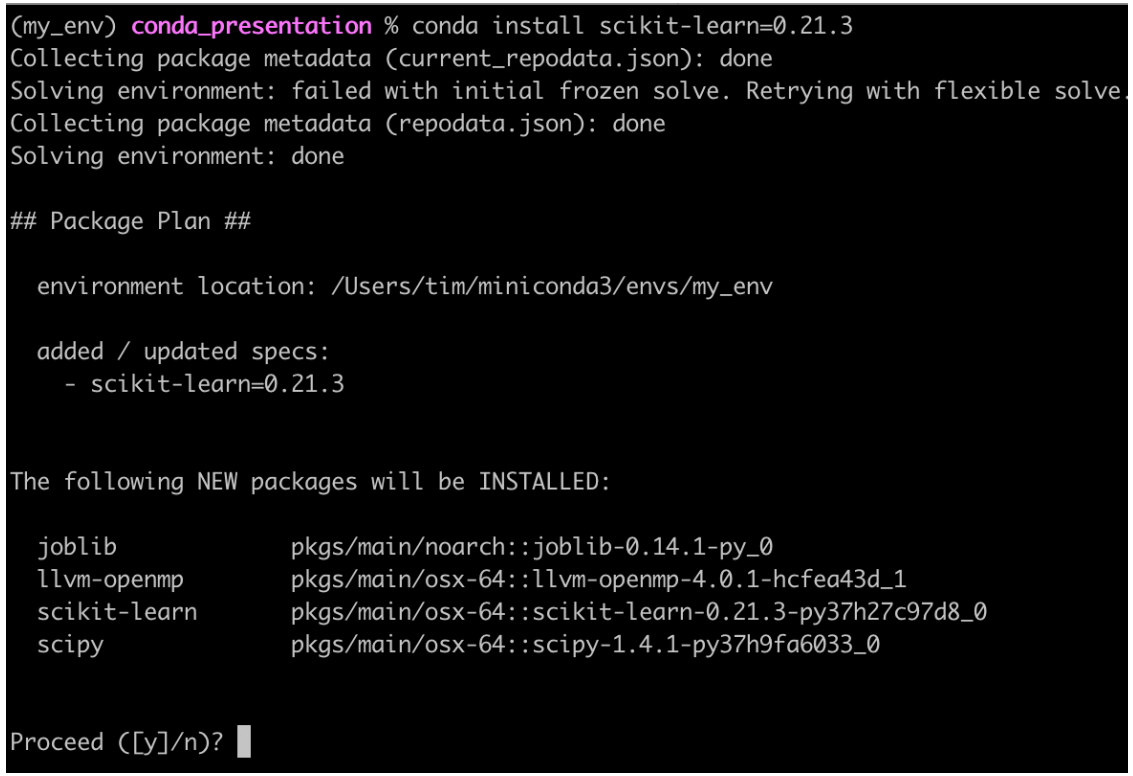
Be kind to yourself!

- Just make a habit of doing:
  - `conda install ipython`
- We can also see that numpy has been installed

```
(my_env) conda_presentation % ipython
Python 3.7.7 (default, May  6 2020, 04:59:01)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.13.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import numpy as np

In [2]: ar = np.array([1, 2, 3, 4])

In [3]: print(ar**2)
[ 1  4  9 16]
```

Installing a specific version

- Due to dependency problems, you might have to use earlier versions of packages
  - This is where using an isolated environment really comes into its own
- This is done with the following command:
  - `conda install <package>=<version>`
- Here, we install a version of scikit-learn that is not the latest stable version
  - This was due to a genuine problem I was having using the latest version of scikit-learn with the 'yellowbrick' library
  - This shows that this is not just a 'token' example



Thanks for listening

- If anyone has any issues – either contact us
- We can help to do this in person in a social meetup as soon as the lockdown is relaxed