Quick Start for Anaconda and Jupyter

We will be using Anaconda and Jupyter Notebook for this semester. This tutorial will walk you through the installation process.

Introduction

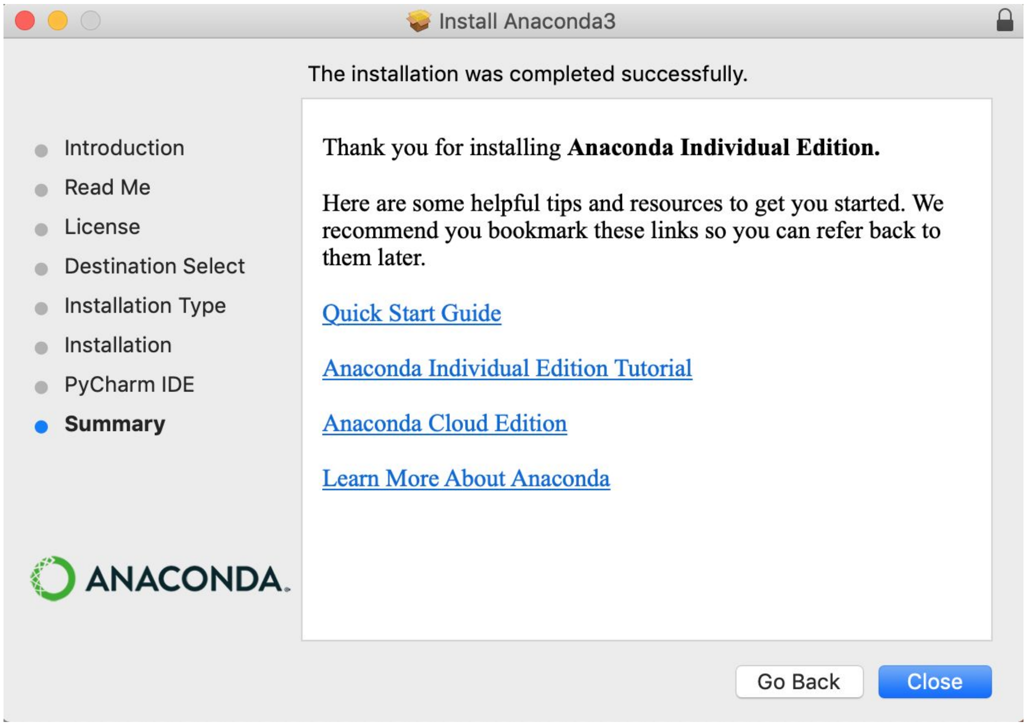
Anaconda is an open source package management system and environment management system that runs on Windows, macOS and Linux. It helps us create and switch between various Python environments and install packages within a specific Python environment on your computer. With the help of Anaconda, we can easily handle compatibility and dependency issues of various Python packages.

Generally speaking, one can install packages in Anaconda in two ways - Anaconda Navigator (desktop graphical user interface) and Conda (command line interface tool).

Installation

- Installing on macOS

    1. Download the graphical macOS installer for your version of Python (Python 3 recommended) at https://www.anaconda.com/products/individual#macos
    2. Double-click the installer and click continue to start the installation.
    3. A successful installation message:



The installation was completed successfully.

Thank you for installing **Anaconda Individual Edition.**

Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

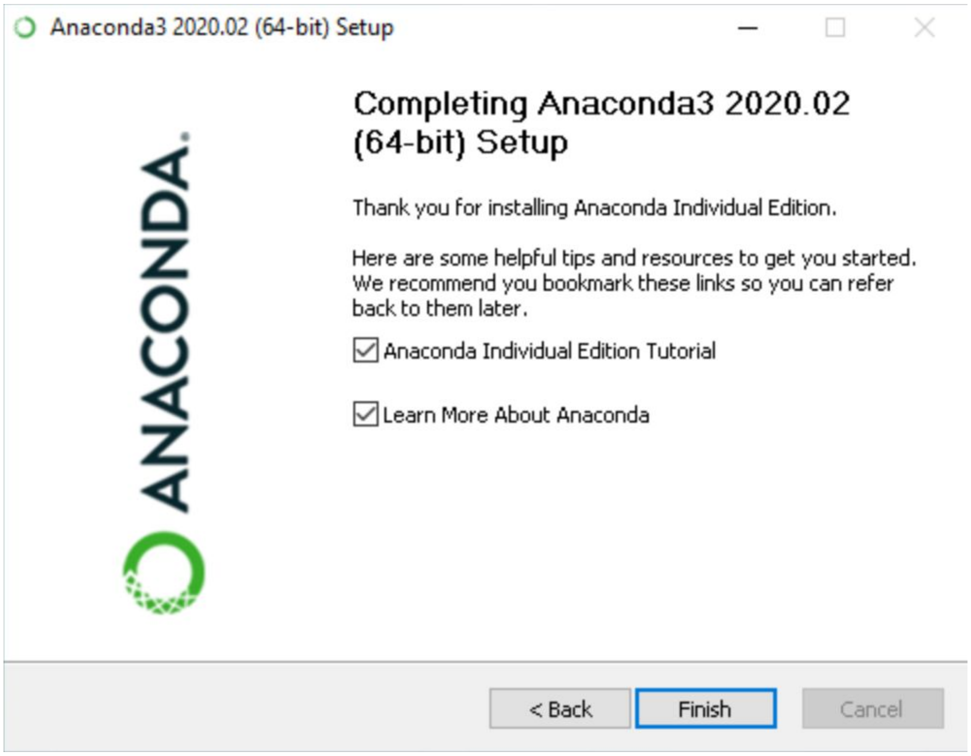
Quick Start Guide

Anaconda Individual Edition Tutorial

Anaconda Cloud Edition

Learn More About Anaconda

- Installing on Windows
    1. Download the graphical macOS installer for your version of Python (Python 3 recommended) at https://www.anaconda.com/products/individual#windows
    2. Double-click the installer and click continue to start the installation.
    3. A successful installation message:


Anaconda3 2020.02 (64-bit) Setup

Completing Anaconda3 2020.02 (64-bit) Setup

Thank you for installing Anaconda Individual Edition.

Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

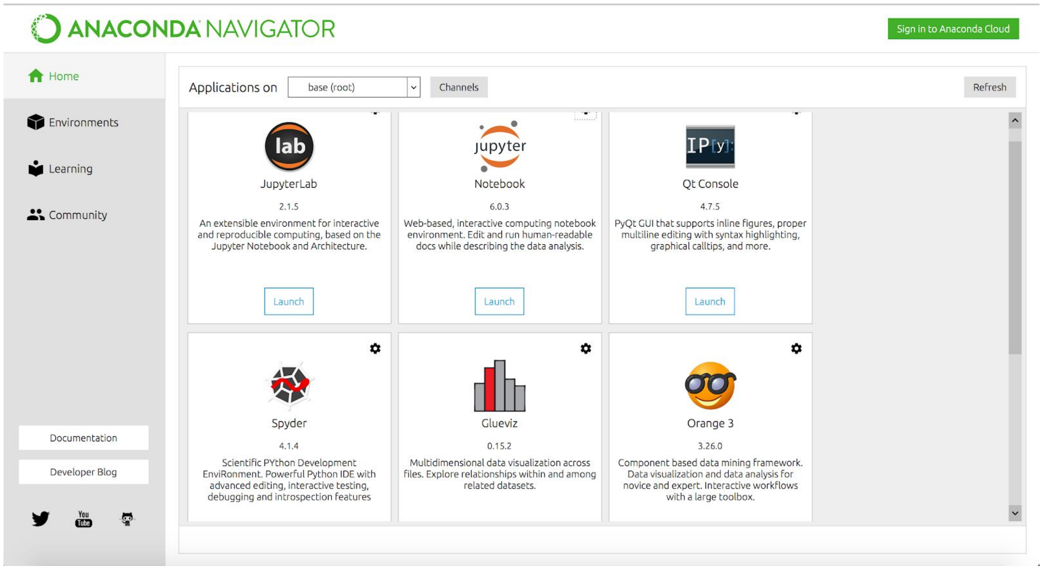
☑ Anaconda Individual Edition Tutorial

☑ Learn More About Anaconda

User Guide for Anaconda

With Navigator, we can create separate Python environments in which Python packages and their dependencies will not interact with other environments. After installing all the packages needed, we will launch the Jupyter Notebook to write and run Python scripts.

1. **Open the application**: open the Anaconda Navigator application either through Mac Launchpad or Windows Start menu.



JupyterLab
2.1.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

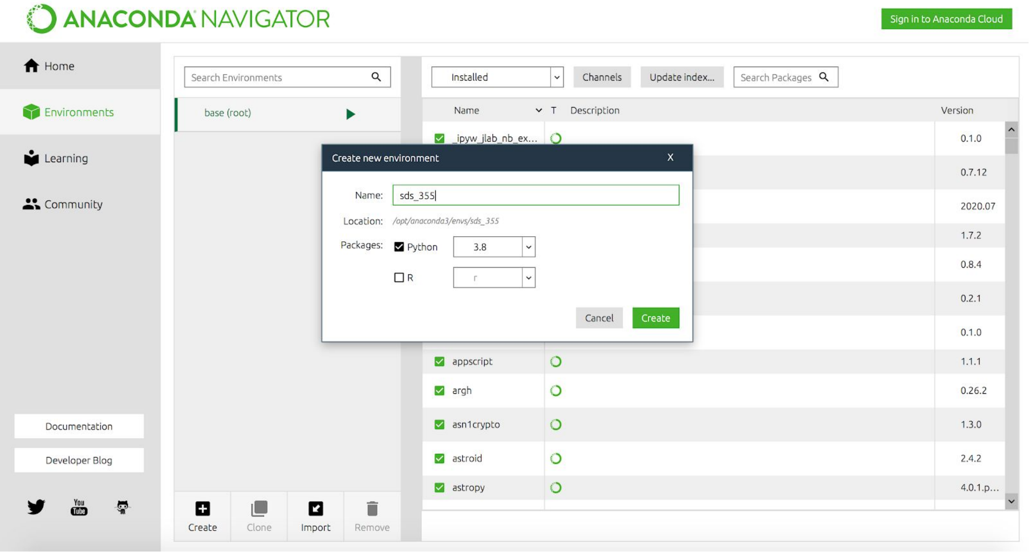
Qt Console
4.7.5
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

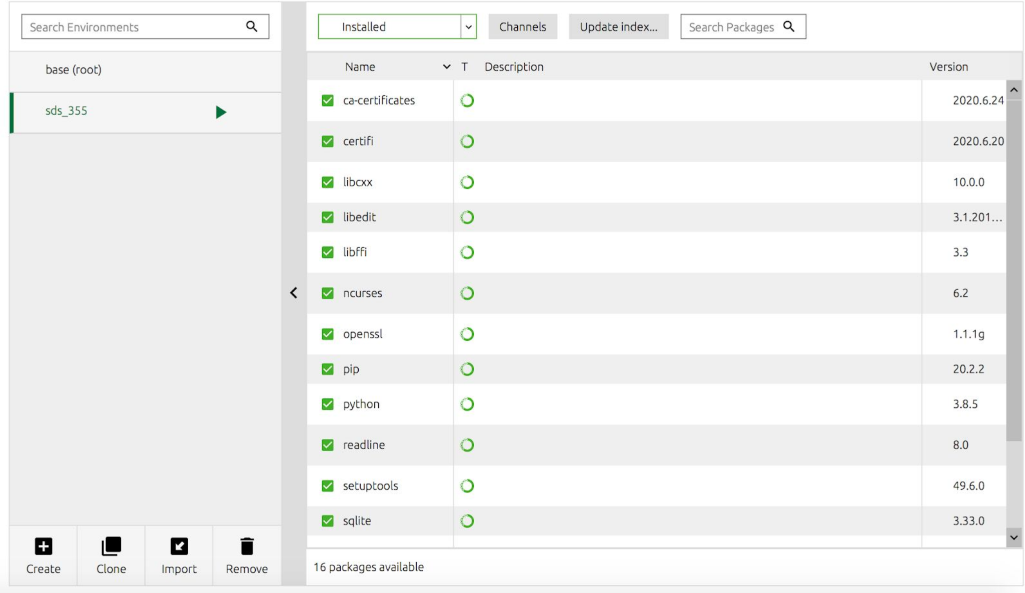
2. **Create new environments**: Create a new environment named sds_355 by clicking the Environments tab and clicking the Create button.



Name: sds_355
Location: /opt/anaconda3/envs/sds_355
Packages: ☑ Python 3.8
□ R r

Name	T	Description	Version
☑ _ipyw_jlab_nb_ex...	○		0.1.0
			0.7.12
			2020.07
			1.7.2
			0.8.4
			0.2.1
			0.1.0
☑ appscript	○		1.1.1
☑ argh	○		0.26.2
☑ asn1crypto	○		1.3.0
☑ astroid	○		2.4.2
☑ astropy	○		4.0.1.p...

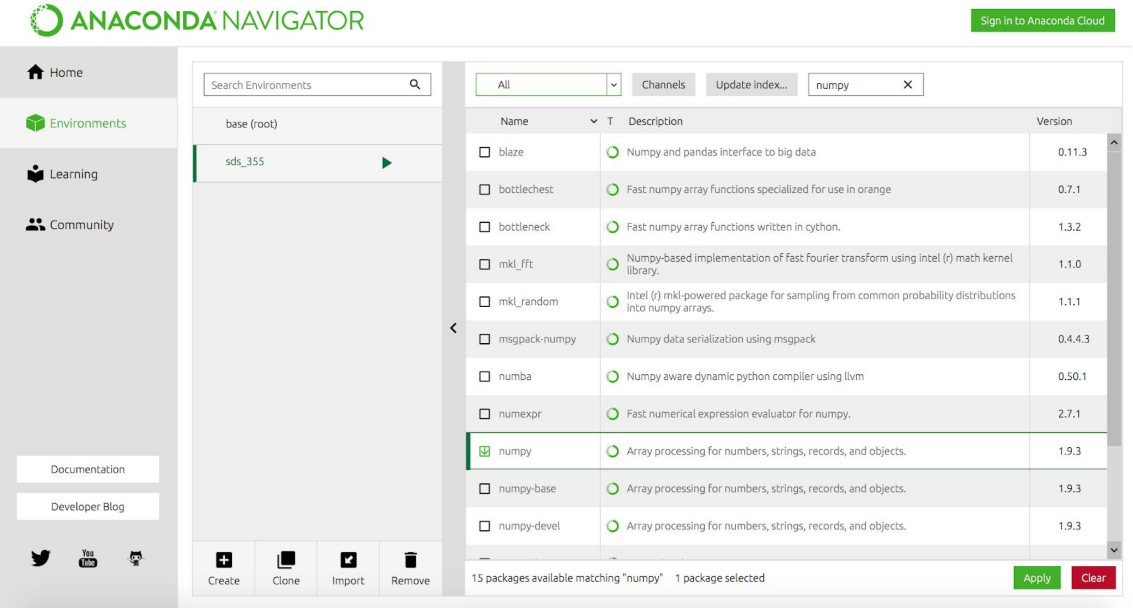
3. **Activate the environment**: Now you should be able to see two Python environments - base(root) and sds_355. Activate the sds_355 environment by clicking the sds_355 tab. The green triangle represents the activated environment.



	Name	T	Description	Version
☑	ca-certificates	○		2020.6.24
☑	certifi	○		2020.6.20
☑	libcxx	○		10.0.0
☑	libedit	○		3.1.201...
☑	libffi	○		3.3
☑	ncurses	○		6.2
☑	openssl	○		1.1.1g
☑	pip	○		20.2.2
☑	python	○		3.8.5
☑	readline	○		8.0
☑	setuptools	○		49.6.0
☑	sqlite	○		3.33.0

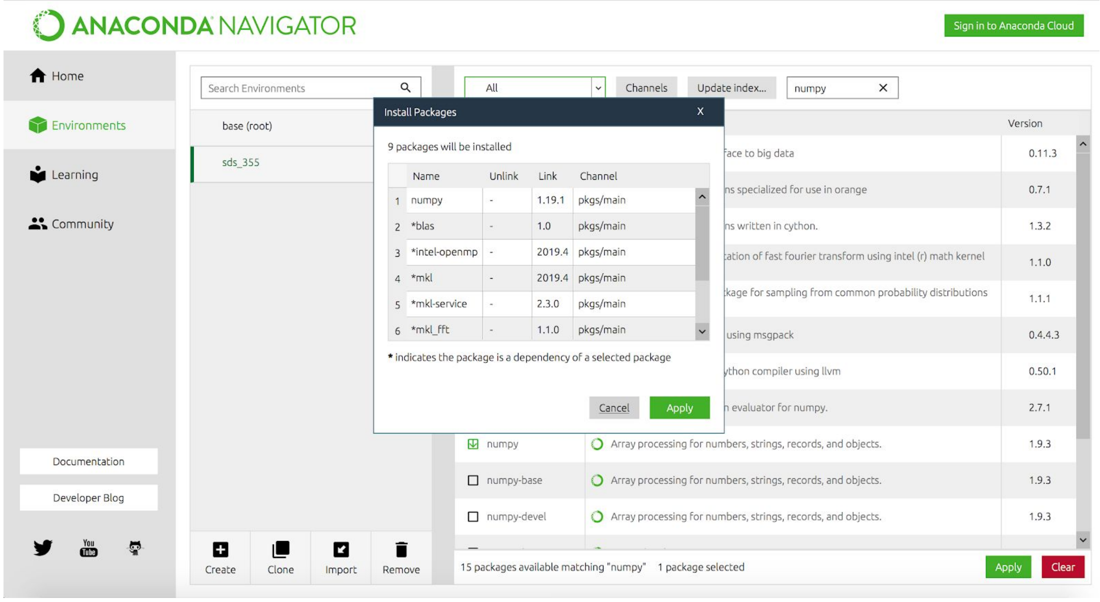
16 packages available

4. **Install packages**: In order to add packages to the current environment, we could click the Update Index first and put the name of the package in the search box. Among all the packages matching the search criteria, we could choose one and click apply. The packages and the dependencies could be automatically installed.



	Name	T	Description	Version
□	blaze	○	Numpy and pandas interface to big data	0.11.3
□	bottlechest	○	Fast numpy array functions specialized for use in orange	0.7.1
□	bottleneck	○	Fast numpy array functions written in cython.	1.3.2
□	mkl_fft	○	Numpy-based implementation of fast fourier transform using intel (r) math kernel library.	1.1.0
□	mkl_random	○	Intel (r) mkl-powered package for sampling from common probability distributions into numpy arrays.	1.1.1
□	msgpack-numpy	○	Numpy data serialization using msgpack	0.4.4.3
□	numba	○	Numpy aware dynamic python compiler using llvm	0.50.1
□	numexpr	○	Fast numerical expression evaluator for numpy.	2.7.1
☑	numpy	○	Array processing for numbers, strings, records, and objects.	1.9.3
□	numpy-base	○	Array processing for numbers, strings, records, and objects.	1.9.3
□	numpy-devel	○	Array processing for numbers, strings, records, and objects.	1.9.3

15 packages available matching "numpy"   1 package selected



Install Packages

9 packages will be installed

	Name	Unlink	Link	Channel
1	numpy	-	1.19.1	pkgs/main
2	*blas	-	1.0	pkgs/main
3	*intel-openmp	-	2019.4	pkgs/main
4	*mkl	-	2019.4	pkgs/main
5	*mkl-service	-	2.3.0	pkgs/main
6	*mkl_fft	-	1.1.0	pkgs/main

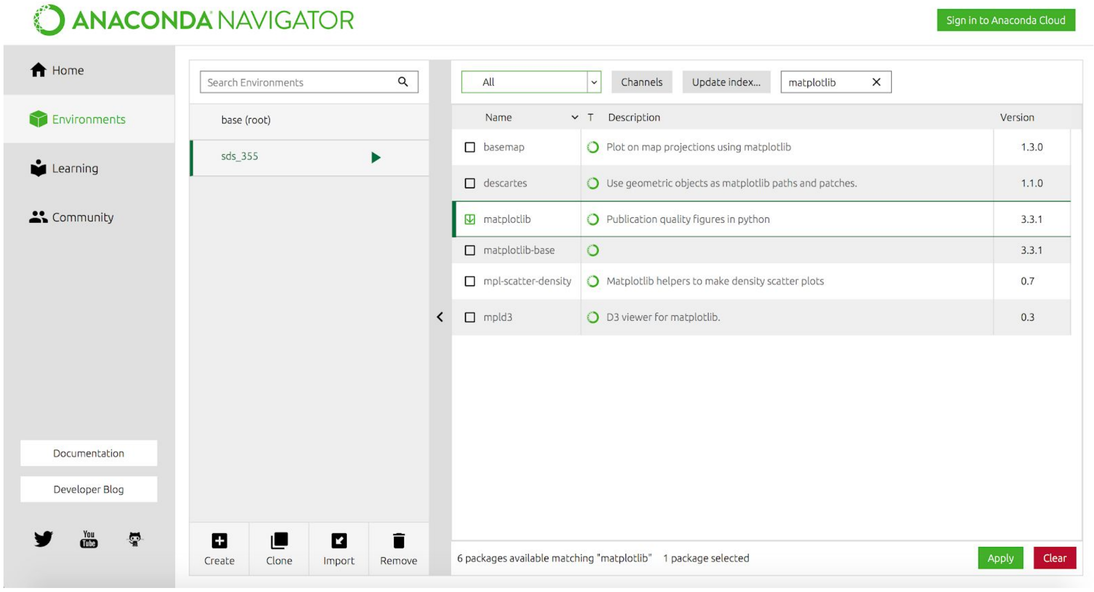
* indicates the package is a dependency of a selected package

Cancel Apply

		Version
	face to big data	0.11.3
	ns specialized for use in orange	0.7.1
	ns written in cython.	1.3.2
	tation of fast fourier transform using intel (r) math kernel	1.1.0
	kage for sampling from common probability distributions	1.1.1
	using msgpack	0.4.4.3
	ython compiler using llvm	0.50.1
	n evaluator for numpy.	2.7.1
☑ numpy	○ Array processing for numbers, strings, records, and objects.	1.9.3
□ numpy-base	○ Array processing for numbers, strings, records, and objects.	1.9.3
□ numpy-devel	○ Array processing for numbers, strings, records, and objects.	1.9.3

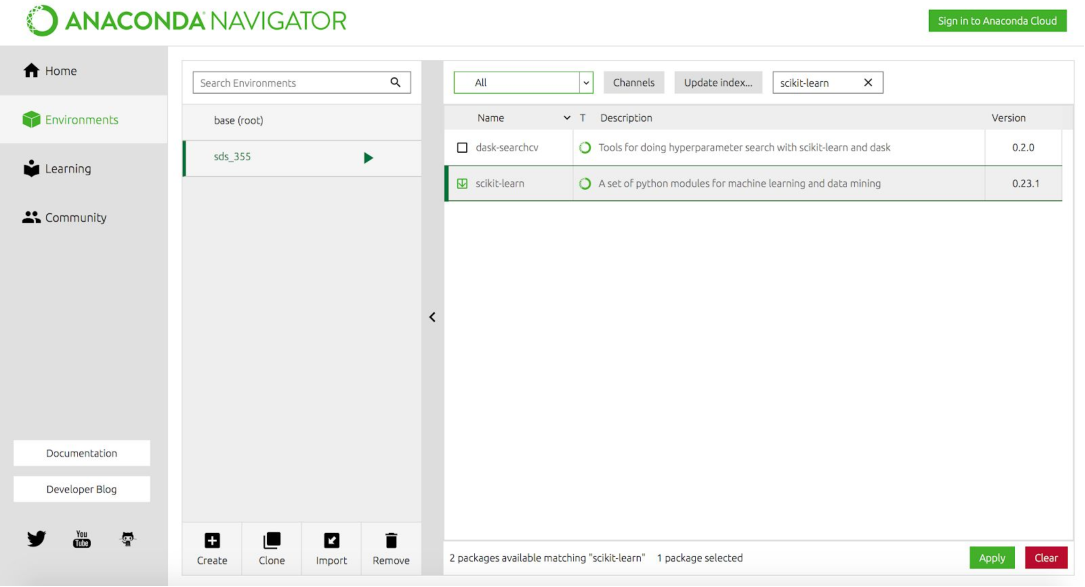
15 packages available matching "numpy" 1 package selected

5. Following the same step, we could also install other packages needed in this class including matplotlib, scikit-learn, etc.



Name	T	Description	Version
□ basemap	○	Plot on map projections using matplotlib	1.3.0
□ descartes	○	Use geometric objects as matplotlib paths and patches.	1.1.0
☑ matplotlib	○	Publication quality figures in python	3.3.1
□ matplotlib-base	○		3.3.1
□ mpl-scatter-density	○	Matplotlib helpers to make density scatter plots	0.7
□ mpld3	○	D3 viewer for matplotlib.	0.3

6 packages available matching "matplotlib" 1 package selected



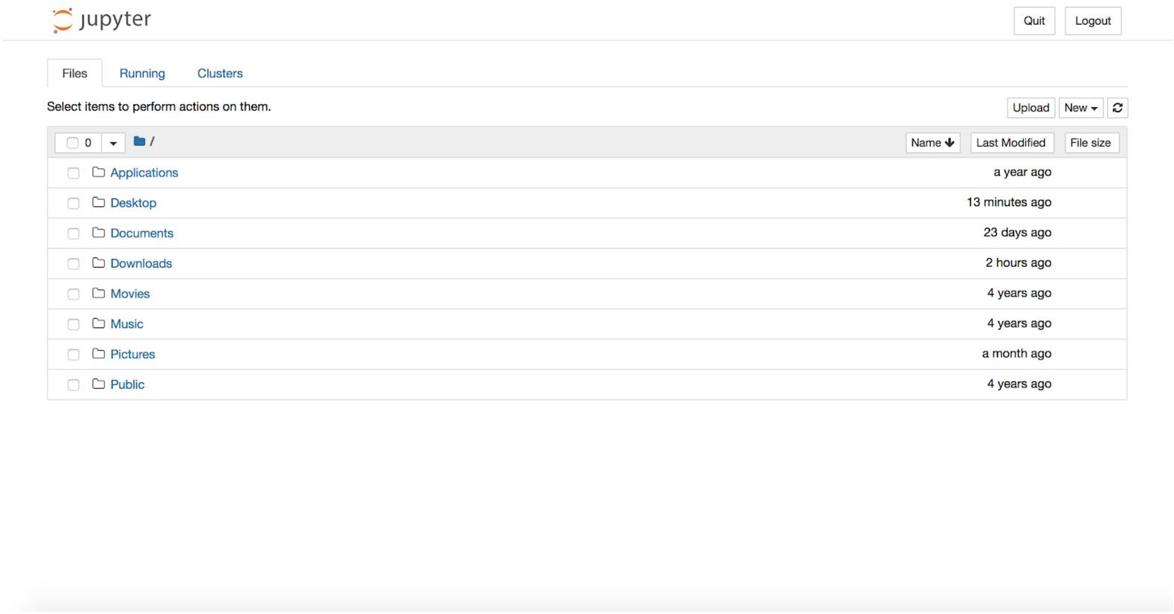
	Name	T	Description	Version
□	dask-searchcv	○	Tools for doing hyperparameter search with scikit-learn and dask	0.2.0
☑	scikit-learn	○	A set of python modules for machine learning and data mining	0.23.1

2 packages available matching "scikit-learn"   1 package selected

Start Coding with Jupyter

After building the Python environment, we are about to launch the Jupyter Notebook to write some Python code. The Jupyter Notebook is a web application where one can write code, equations and narrative text and share documents with others.

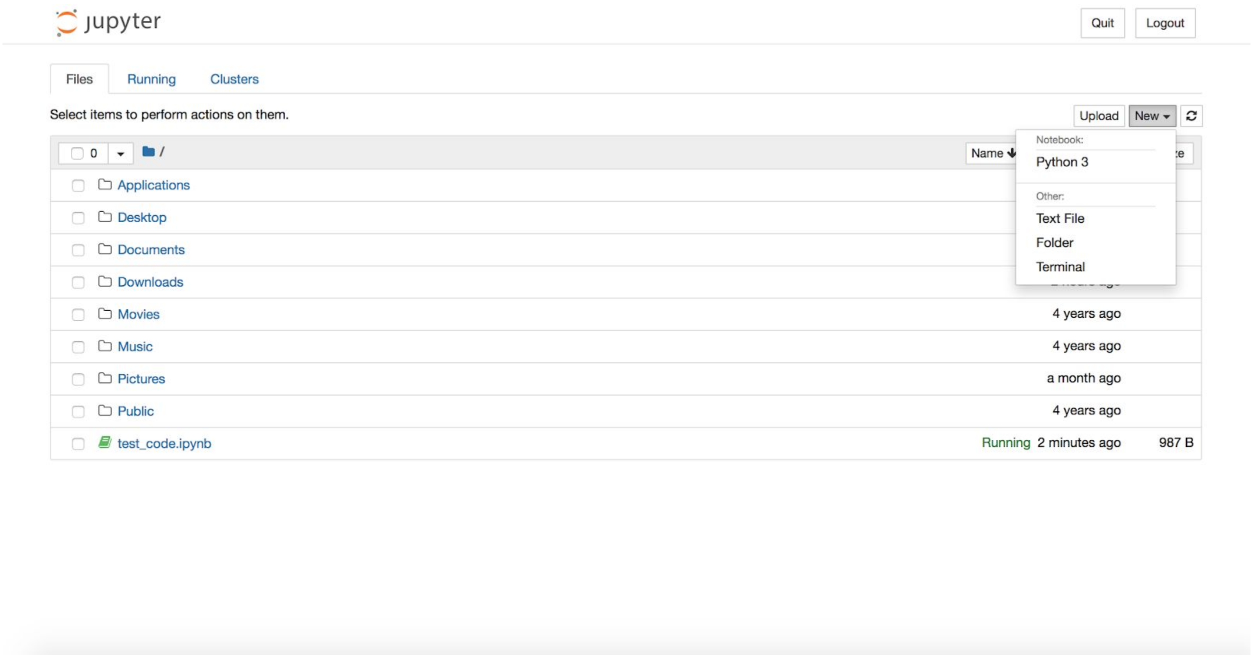
The Jupyter Notebook has been pre-installed for the base environment. For new environments, you can easily install it and launch it in the home view. You should be able to see the following webpage:



Select items to perform actions on them.

Name	Last Modified
□ Applications	a year ago
□ Desktop	13 minutes ago
□ Documents	23 days ago
□ Downloads	2 hours ago
□ Movies	4 years ago
□ Music	4 years ago
□ Pictures	a month ago
□ Public	4 years ago

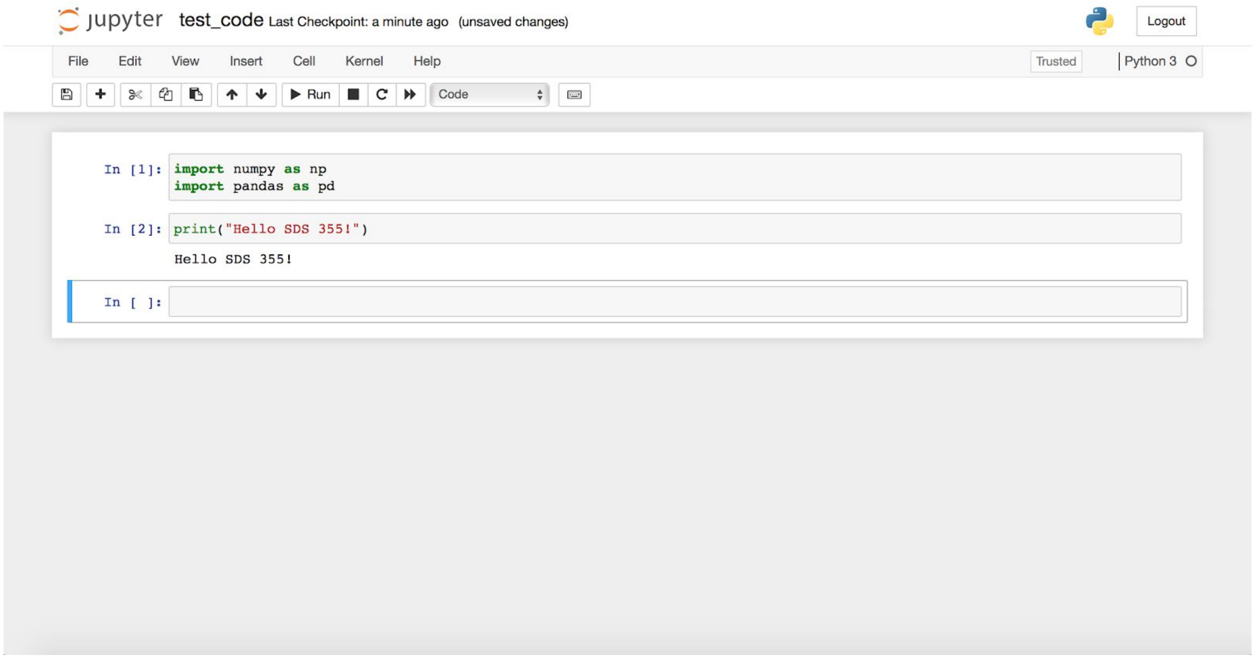
We can click New tab and create a new Python 3 notebook:



Select items to perform actions on them.

```
Applications
Desktop
Documents
Downloads
Movies                                   4 years ago
Music                                    4 years ago
Pictures                                 a month ago
Public                                   4 years ago
test_code.ipynb        Running  2 minutes ago        987 B
```

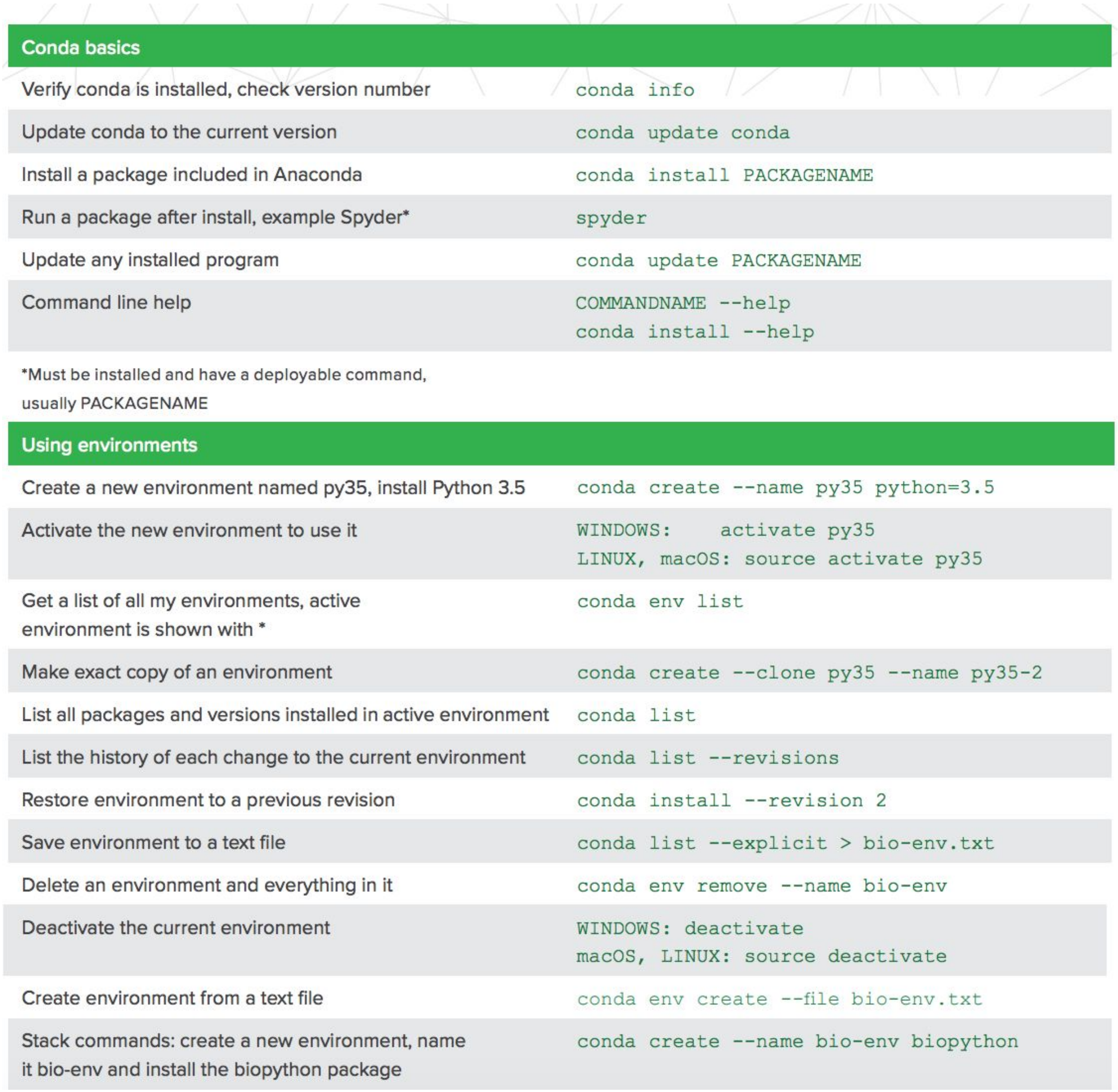
Start coding and have fun:



```
In [1]: import numpy as np
        import pandas as pd

In [2]: print("Hello SDS 355!")

        Hello SDS 355!

In [ ]:
```

For people who are interested in **Conda CLI** tool:

This cheat sheet will be extremely helpful:



Conda basics	
Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`

Command line help	`COMMANDNAME --help`
	`conda install --help`

*Must be installed and have a deployable command, usually PACKAGENAME

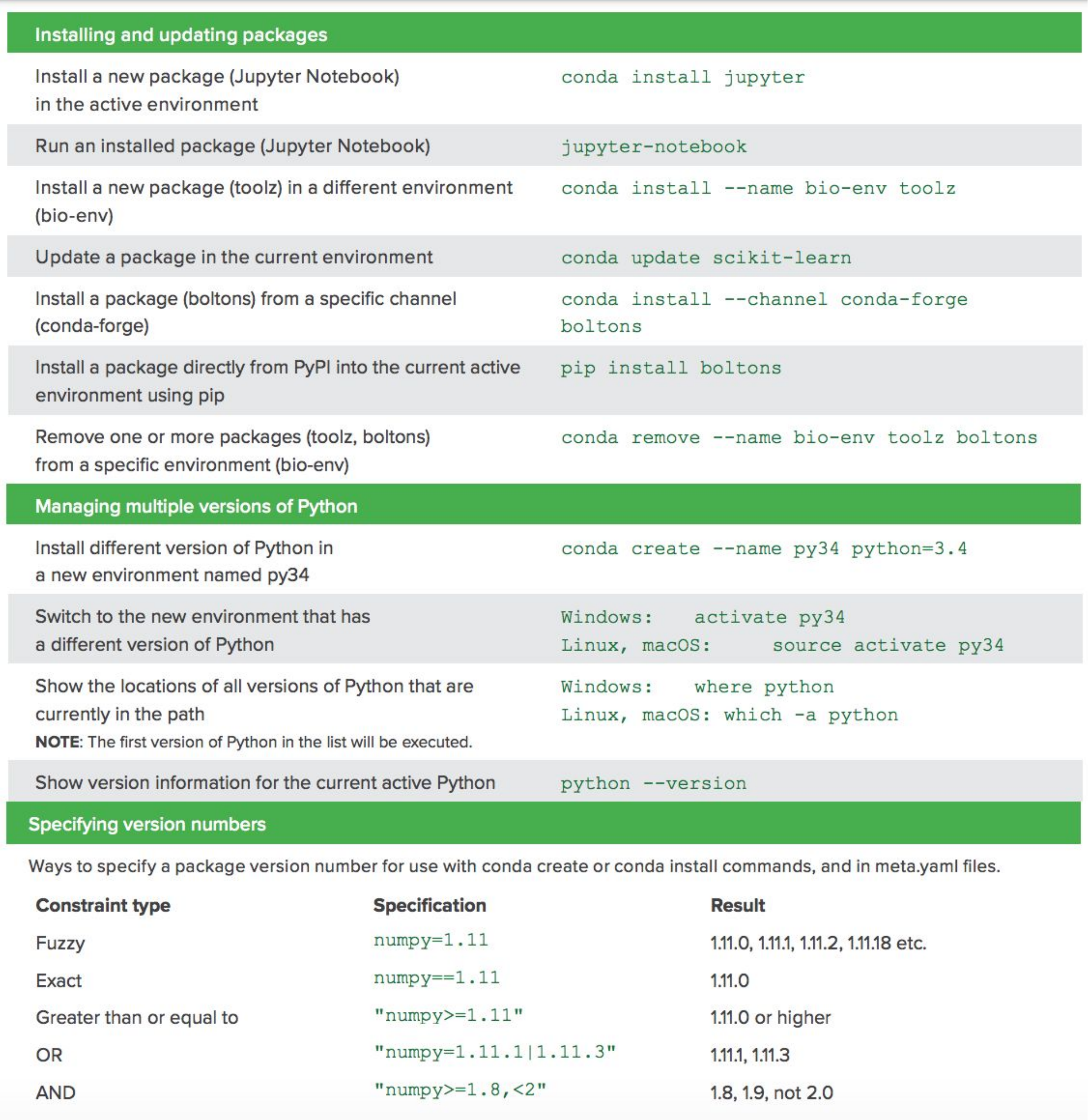
Using environments	
Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`

Activate the new environment to use it	`WINDOWS:    activate py35`
	`LINUX, macOS: source activate py35`

Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`

Deactivate the current environment	`WINDOWS: deactivate`
	`macOS, LINUX: source deactivate`

Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`



Installing and updating packages	
Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python	
Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`
Switch to the new environment that has a different version of Python	Windows: `activate py34` Linux, macOS: `source activate py34`
Show the locations of all versions of Python that are currently in the path **NOTE**: The first version of Python in the list will be executed.	Windows: `where python` Linux, macOS: `which -a python`
Show version information for the current active Python	`python --version`

Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	`numpy=1.11`	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	`numpy==1.11`	1.11.0
Greater than or equal to	`"numpy>=1.11"`	1.11.0 or higher
OR	`"numpy=1.11.1|1.11.3"`	1.11.1, 1.11.3
AND	`"numpy>=1.8,<2"`	1.8, 1.9, not 2.0

References

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