

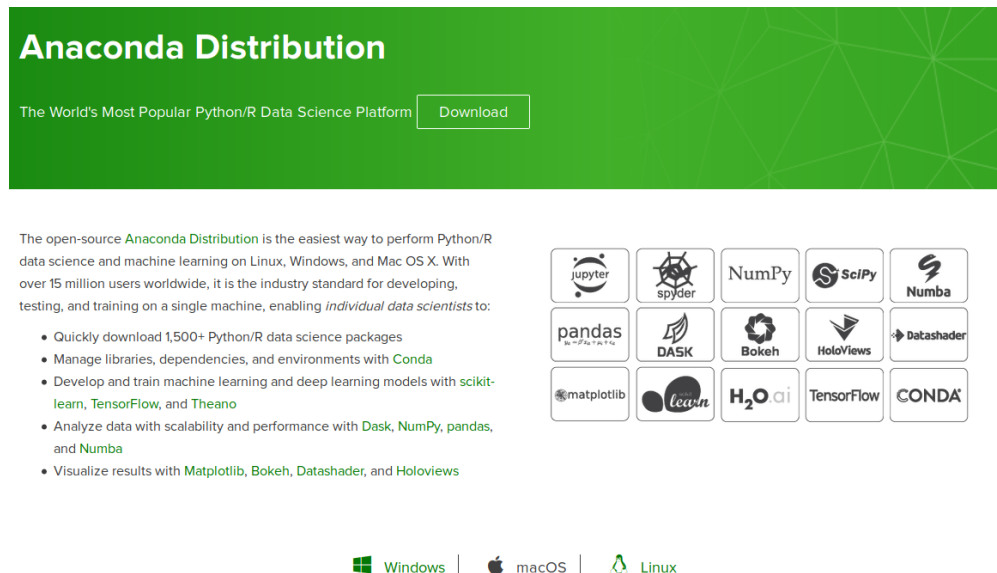
01_installation

July 16, 2019

0.1 # 01_Installion: installing anaconda

- python distribution
- python package management
- Resolve dependencies

Specifications: - Linux, mac, windows (hopefully :p) - Anaconda 2019.03



figure

0.2 Step 1: Install anaconda

<https://docs.anaconda.com/anaconda/install/>

0.3 Step 2: Verify installation

- open python prompt: `ipython`
\$ `ipython`

0.4 # Setup conda environments

<https://conda.io/projects/conda/en/latest/>

Conda



Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

In its default configuration, conda can install and manage the thousand packages at repo.continuum.io that are built, reviewed and maintained by Anaconda®.

Conda can be combined with continuous integration systems such as Travis CI and AppVeyor to provide frequent, automated testing of your code.

The conda package and environment manager is included in all versions of Anaconda and Miniconda. [Anaconda Repository](#). Conda is also included in [Anaconda Enterprise](#), which provides on-site enterprise package and environment management for Python, R, Node.js, Java and other application stacks. Conda is also available on PyPI, although that approach may not be as up to date.

figure

Making a new environment

Example environment test installing package numpy 1.14

```
$ conda create -n test numpy=1.14 # specific package
```

List conda environments

```
$ conda info --envs
```

Initialize (must restart terminal)

```
$ conda init
```

Activate environment test

```
$ conda activate test
```

Deactivate environment test

```
$ conda deactivate
```

0.5 # Setup conda environments

<https://conda.io/projects/conda/en/latest/>

0.6 Step1 : Install jupyterlab

using yaml file for conda

```
$ conda env create -f binder/environment.yml
```

0.7 Step 2: Start jupyterlab

```
$ jupyter lab
```