Unix

Python

LaTeX

C/C++

bash

make

git

Statistics

ssh

Introduction to anaconda

Sabrina Einecke

HEAG Introduction Course

Anaconda

Package Manager

- Easy installation of a bunch of libraries, especially for data science

  (jupyter, NumPy, SciPy, pandas, Bokeh, matplotlib, scikit-lean, TensorFlow, theano etc.)

- Many python packages pre-installed

Anaconda

Environment Manager

- Often you will work on multiple projects in parallel. Each project requires different packages, versions and dependencies. How do you organise this?
- It might happen that you have an error in your code, google it, and find out the problem is a faulty package, and you need to upgrade it. But the upgrade requires the upgrade of other packages, too. That might influence other parts of your code or other projects. How can you test it without risking to damage anything? (Downgrading is mostly problematic.)

→ Different environments

Anaconda

- Create a new environment

  ```
  conda create --name=NAME
  ```

- Create a new environment with a specific python version

  ```
  conda create --name=NAME python=3
  ```

- Install packages into current environment

  ```
  conda install PACKAGE
  ```

- Install packages into environment *NAME*

  ```
  conda install --name NAME PACKAGE
  ```

- Install specific version of a package

  ```
  conda install PACKAGE=1.0.1
  ```

- Install non-conda packages

  ```
  pip install PACKAGE
  ```

- See installed packages

  ```
  conda list
  ```

Anaconda

- Activate a specific environment

```
source activate NAME
```

- List all environments

```
conda env list
```

→ See cheat sheet for more commands: https://conda.io/docs/_downloads/conda-cheatsheet.pdf