Notes on setting up a scientific python environment

Course in Computational Cognitive Neuroscience 2016/2017

We provide all exercises in Jupyter Notebooks compatible to Python 2.7. Exercises make use of scientific modules that are written in the python programming language. To install such an environment and a Jupyter Notebook viewer with least problems we strongly recommend the Anaconda framework **for Python 2.7**: *https://www.continuum.io/downloads*

The developers behind this package avoid bloatware and focus on installing the most important scientific python modules in versions that are compatible to each other:
https://docs.continuum.io/anaconda/pkg-docs

Your first task is to open `0. Python Introduction.ipynb` where you may find out what that all means.

Opening Jupyter Notebooks

After any of the installation methods described below you need to download the course notebooks somewhere (download directory is called `/path/to/your/download/directory` below) and then type

```
$ cd /path/to/your/download/directory
$ jupyter notebook
```

in a terminal (on Windows that means `cmd.exe` (search for it), alternatively `PowerShell` (needs to be installed)).

This will open a notebook server in the background that you need to keep open since your code runs there, and a browser window with the directory structure in the foreground.

In this browser window you just need to click on a notebook to open it.

You can avoid installation by using campus PCs.

The general campus PCs in the public student computer lab Spinozagebouw -1.55 A/B use Windows 7 and will have the Anaconda environment plus the IDE PyCharm and PowerShell installed during the coming days. You can login with your RU account there.

Ubuntu on the PCs of the AI department (e.g. in the tk student computer room) has access to a directory where Anaconda python lives. *You need a science account to login to Ubuntu.* When logged in to one of these PCs you can activate the full environment by typing

```
$ source /vol/optdcc/anaconda2/bin/activate
```

in a terminal. After the environment is activated the python-related commands you need can all access the scientific modules that are installed. Also, extra software like `$ jupyter notebook` or the IDE Spyder (`$ spyder`) is available then.

The command above is only temporary. If you want to avoid typing it every time you open a terminal, create a file called `.bash_rc` in your home directory if it does not exist. Then add to this `~/.bash_rc` the line:

```
export PATH=/vol/optdcc/anaconda2/bin:$PATH
```

The Anaconda framework will then be available every time you log in.

Installation on your own computer

Download **Anaconda 4.1.1. Python 2.7** for your operating system from here:

https://www.continuum.io/downloads

Linux & OS X
Follow the Anaconda installation instructions. E.g. after downloading do:

```
$ bash Anaconda2-4.1.1-Linux-x86_64.sh
```
or
```
$ bash Anaconda3-4.1.1-MacOSX-x86_64.sh
```

depending on your system. The scientific Python framework is then being installed.

In addition to the standard scientific modules that have been installed now, we need `chainer` (for neural networks) and `pymc` and `pymc3` (for bayesian methods).

To install those three, just open a terminal and type:
```
$ pip install chainer
$ conda install pymc
$ conda install -yc trung pymc3=2016.4.27
```

In case you choose not to let Anaconda reset your PATH (i.e. in case you want to keep your own system-wide python environment because your own python projects rely on specific versions), you will need to use the `anaconda2/bin/activate` script every time you want to access the framework and Jupyter Notebooks (similar to activating it for the tk computers above):

```
$ source /home/username/anaconda2/bin/activate
```

Windows
For any programming (except enterprise software or non-mobile game programming) it is generally recommended to work inside a Unix-like system such as Linux (e.g. Ubuntu) or OS X. It is possible to install a scientific python framework under Windows, but you may run into more problems, e.g. because there is no decent terminal or python module developers are more likely to ignore Windows compatibility.

Follow the instructions from *https://www.continuum.io/downloads* to install Anaconda with the *.exe file. We recommend resetting the PATH (default option).

After the installation succeeded, you will have to install the additional modules `chainer` (for neural networks) and `pymc` and `pymc3` (for bayesian methods).

To install, open `cmd.exe` or `PowerShell` and type:
```
$ conda install pymc
```

For installing pymc3 you need to follow points 2 and 3 from here:
http://datahans.blogspot.nl/2016/04/installing-pymc3.html

For installing `chainer`, you will need to install the `Microsoft Visual C++ Compiler for Python 2.7` from here:
https://www.microsoft.com/en-us/download/details.aspx?id=44266

After installation you can type the following into a terminal to get chainer, too:
```
$ pip install chainer
```

All systems and methods

Test whether your environment is set up by opening an iPython console in a terminal window:

```
$ ipython
```

When opening it should mention Anaconda, e.g.:

```
Python 2.7.12 |Anaconda 4.1.1 (x86_64)
```

Exit by typing: `exit()` or pressing `[Ctrl] + [D]`.

Debugging, code editors and IDEs

When doing computation intensive programming you will find it necessary to inspect variables during certain steps of the code to find problems. For *efficient debugging* we recommend making use of a debugger inside an IDE.

The IDE **spyder** is installed together with Anaconda.

PyCharm needs to be installed separately, but is compatible with Anaconda:
https://www.jetbrains.com/pycharm/

It is recommended to first install Anaconda, then PyCharm. In the `Preferences` of PyCharm you will then need to search for `Python Interpreter` and choose the Anaconda binary in the settings window:
https://docs.continuum.io/anaconda/ide_integration (slightly outdated screenshots)

You can of course also write Python code in your code editor of choice (gedit, Notepad++, TextWrangler…) and make use of Python's standard debugging tools if your prefer to use them.

If you run into problems with the above, contact Katja (no official CCN tutor):
k.muller@donders.ru.nl