Introduction to Python for Social Science

Musashi Harukawa, DPIR

1st Week Hilary 2020

Course Overview

Schedule

8-week long course will take place on Wednesdays at the IT Lab in the Manor Road Building. Each week will consist of a three-hour session divided roughly as follows:

- 1615-1715: Lecture
- 1715-1725: Break
- 1725-1825: Workshop
- 1825-1925: Clinic

Lecture

- During the lectures I will discuss a mixture of theory and methods. The slides will be made available at: **to be confirmed**
- Please feel free to stop me and ask questions! I have alloted time for this.

Workshop

- In the workshop, you will work through a number of set programming problems and discussion questions.

Course Structure

The topics for the eight weeks are as follows:

1. Introduction to Python and the Development Environment
2. Data Structures and Pandas I
3. Data Structures and Pandas II
4. Data Visualisation
5. Machine Learning with `scikit-learn` I
6. Machine Learning with `scikit-learn` II
7. Mining the Web
8. Introduction to Natural Language Processing

Feedback

- This is a brand new course, and far from perfect.
- I would appreciate and welcome any feedback on any aspect.
- Feedback can either be:
  - sent to me at `musashi.harukawa{=^._.^= minus C}ox.ac.uk`
  - or communicated via a Google survey link that will be provided.
- Please also feel free to come talk to me in person!

Week 1: Introduction to Python and the Development Environment

This week will cover the following points:

1. What is Python?...
2. ... and what can I use it for?
3. What are the tools I have to write, test and run Python code?
4. Writing your first Python script.

What is Python?...

Python is an *open-source, general-purpose scripting language*.

Open-Source

- Built by a community
- Maintained by a community
- Free to use for all

General-Purpose

- If you're doing it on a computer and there's some repetitive element, then you can automate it in Python.
- Python isn't limited to Data Science, but it's very popular with data scientists!

Note: Large community means that a larger number of people create, contribute to, and maintain the data analysis tools that we all use.

Scripting

- No strict definition for what a "script" is.
- Series of commands to automate some task.
- Like a pipeline: takes some inputs, does some things to these

and what can I use Python for?

Some reasons you, as a social science researcher, may want to learn Python:

I want to...

- Clean up my messy data!
- Run analyses with (hundreds of) millions of data points
  - it won't fit into an excel spreadsheet!
- I want to automate downloading several decades of newspaper articles!
- I want to create beautiful (interactive) visuals to accompany my analyses!
- I want to uncover hidden structures linking parliamentary committees!
- ... and more!

Python vs R

Task	python	R
General Purpose Programming	Yes	No
Regression Analysis	Kinda	Yes
Machine Learning	Yes	Yes
Web Scraping	Yes	Kinda
Natural Language Processing[1]	Yes	Yes
Data Visualisation	Yes	Yes

Conclusion: . . . it depends, but ideally you want to learn both!

[1] Python and R both provide extensive and powerful natural language processing libraries, e.g. `nltk`, `gensim` in Python; `tm`, `quanteda` in R, and `spaCy` in both. Unfortunately, there are many techniques that are only implemented in one language but not the other.

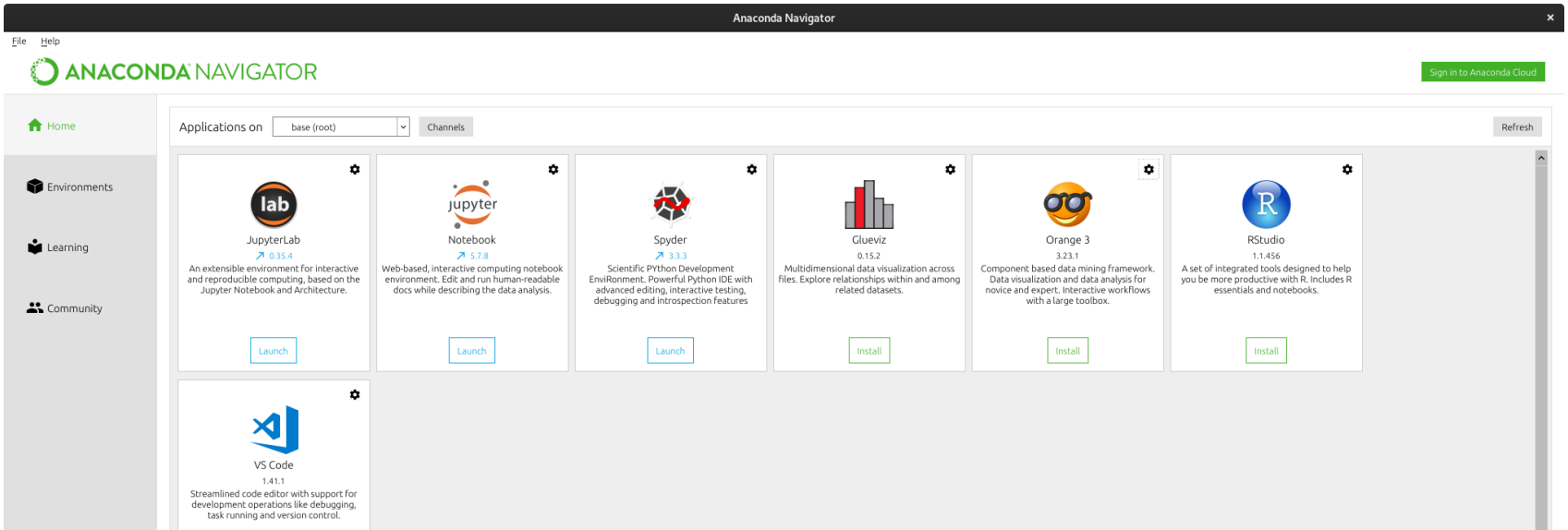
Basic Coding Tools

- `Python` is a language, and not an application.
- We have a variety of tools for developing our python code.

Anaconda

- Environment and software manager.
- Can be used from the command line (`cli`) or browser-like interface (anaconda-navigator).

Note: At this point, I switch windows to open up the anaconda navigator, and go through the various relevant tabs. These include the launcher for various apps and the environment manager.



Basic Workflow

1. Open up Anaconda Navigator
2. Open up Jupyter Lab (or Notebook)
3. Navigate to relevant directory
4. Open pre-existing notebook, or create new one.
5. Start coding!

Getting Started!

We are now going to write our first python command!

Step 1: Create Notebook

Following the steps from before, create a new notebook in your working directory, and call it `my_first_notebook.ipynb` (or whatever).

Step 2: Write Command

Type the following code into the first cell:
`print("Hello World!")`
and now hit `CTRL+ENTER`

Breaking down our first script

Let's break down what we did here.

print()

```
print("This is my first line of code!")
^___^^                                 ^
```

This is the *print function*. It prints whatever argument it is given to the console. We know it is a function because it has brackets.

input

```
print("This is my first line of code!")
      ^-----------------------------^
```

This is the input to the function. There is only one argument for this function, so there are no commas. This is what the print function prints to the console.

quotes

```
print("This is my first line of code!")
      ^                               ^
```

The double quotes are used to create a *string*. We can also use single quotes, but if we don't use any quotes let's see what happens