Introduction to Python for Social Science

Musashi Harukawa, DPIR

Optional, Hilary Term 2020

Course Description

Students have the option of taking an 8-week fast-paced introduction to conducting social science research with Python. This will take the form of a weekly 1-hour lecture, 1-hour workshop, and optional 1-hour clinic. These run consecutively from 16:15-19:15 on Wednesdays (dates provisional).

The aim of this course is two-fold. The first goal is to teach students essential *data analysis* and *scripting* skills so that they are able to put together short programs and run their own analyses. The second aim is to give a introduction to the numerous techniques and technologies that researchers can integrate into their own research, and to provide incentives to invest in computational methods and skills. Some of the techiniques that will be taught include:

- Using Jupyter as a Research and Development Tool
- Data Cleaning and Merging with `pandas`
- Static Data Visualisation with `matplotlib` and `seaborn`
- Introduction to Machine Learning with `scikit-learn`
- Introduction to Web Scraping with `beautifulsoup`
- Introduction to Natural Language Processing with `spaCy`

Note that this course is not a course in *programming.* Students will learn how to use Python for data analysis and research, but the primary focus is on teaching them about the available methods and the bare minimum level of programming to implement these methods. Also note that this course is optional, and there will be no marked assignments, but there will be weekly tasks designed to aid learning. Students are encouraged to complete these tasks, and to ask questions about them during the workshop and clinic.

This course is aimed at complete beginners, although experience with other programming languages (such as `R`) may provide some useful reference points. As spaces are limited, priority will be given to students without prior experience using Python, and those who have a use case for computational tools in their research.

Students are required to bring their own laptop to class. The recommended minimum specifications are:

- *Operating System*: Linux or MacOS optimal, Windows 10 acceptable.[1]
- *RAM*: 4GB minimum
- *Storage*: 10GB free space minimum

Students will also be required to install relevant software prior to coming to class (details below).

The course does not follow any particular textbook. However, students are encouraged to look at *Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, 2nd edition* by Wes McKinney. This text is slightly more advanced than the course, but provides clear explanations and a lot of coding examples.

More generally though, given the rapidly changing nature of the field, students will be encouraged to learn how to solve programming problems by consulting the broader Python community, whether through stackoverflow, Reddit, or package documentation.

For further questions regarding course specifics or accessibility requirements, please feel free to write to me at musashi.harukawa@merton.ox.ac.uk.

[1] If you have something other than Windows, Mac or Linux (e.g. Windows RT, iPadOS, Android), come talk to me.

Course Outline

Please note that this course outline is still provisional, and the exact structure of the course may change. Please also refer to the final published version of this course, which will be available from -1st Week of Hilary Term 2020.

Week 0: Installing Anaconda

Aim: Install Python and Development Tools

Prior to beginning the course, students need to install Python 3 and the relevant development tools. Although there are many ways to go about this, I recommend that students use Anaconda, which offers an easy-to-use graphical interface for package installation.

Anaconda can be downloaded from https://www.anaconda.com/distribution/. Follow the instructions on the website, and *install the Python **3.x** (i.e. 3.7, unless 3.8 comes out in the next few months) edition* for your operating system.

Once you have downloaded the installer, simply follow the instructions provided by the installer. Note that the installation size is somewhat large; 2.9 to 3.7GB, depending on your operating system. In general, the default settings should work, but please feel free to opt out of Anaconda Cloud, mail subscription, and error reporting.

Week 1: Introduction to Python and the Development Environment

Learning Aims:

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

There are three learning goals in the first week. The first relates to what Python is, and how it can be useful for social science researchers. Students will learn about the various use cases for Python, and come up with ways that it may help them achieve their research aims.

The second learning goal is to gain familiarity with the tools used to code in Python and present their research. These include Jupyter notebooks, IDEs and the terminal. Students will primarily use Jupyter notebooks in this course, but are welcome to use alternative development tools.

The final goal is to write their first program in Python. Commands and operators such as `print`, `+`, `&` etc. will be introduced.

Week 2: Data Structures and Pandas I

Learning Aims:

1. Data Structures in Base Python
2. Data I/O with `pandas`
3. Selecting, filtering and indexing data in `pandas`
4. Summary statistics in `pandas`
5. `NumPy` Data Types

The second week introduces students to some basic data structures in base Python and a key library for data analysis: `pandas`. In base Python, students will learn about lists and dictionaries.

In `pandas`, students will learn how to read in various data formats, clean and index data, and produce summary statistics. Students will also be introduced to data types.

The goal of this week is to be able to use `pandas` to open `csv`, `html`, `xls`, or `dta` files, to slice and filter them, and then to produce summary statistics.

Week 3: Data Structures and Pandas II

Learning Aims:

1. Writing Python functions
2. Vectorize with `apply`
3. Split-apply-combine with `groupby`
4. Working with datetime data

The third week builds on students' knowledge of `pandas`, introducing two key tools in data analysis: `apply` and `groupby`. Students will also learn how to write functions and be introduced to the idea of namespaces.

By the end of this week, students should have a sufficient grounding in handling tabular data with base Python and `pandas` to deal with most data cleaning and reshaping tasks they use in their own research.

Week 4: Data Visualisation

Learning Aims:

1. The "philosophy" of `matplotlib`
2. Figures, subplots, axes, legends
3. Plotting uni- and bivariate datasets in `matplotlib`
4. The convenience of `seaborn`
5. Customising your plots

The fourth week introduces two key libraries for data visualisation: `matplotlib` and `seaborn`. Students will learn the philosophy behind data visualisation, and how to create a number of useful 2D graphs.

Week 5: Machine Learning I

1. Introduction to Machine Learning
2. Introduction to `scikit-learn`
3. kmeans clustering with `scikit-learn`
4. Presenting your kmeans clustering results

The fifth week introduces to students to machine learning with the `scikit-learn` library. After discussing the aims and developments within the field, students learn about unsupervised clustering with the k-means algorithm.

Week 6: Machine Learning II

1. Random Forest regression and classification with `scikit-learn`
2. k-fold cross validation with `scikit-learn`
3. Hyperparameter Tuning with `Grid` and `RandomizedSearchCV`
4. When not to use linear models

The sixth week introduces supervised machine learning with the random forest algorithm. Students then learn about cross-validation techniques and their implementation in `scikit-learn`. Finally students learn about hyperparameters, and how to choose the optimal initialising parameters for the model. The lesson ends with a discussion about the difference between prediction and explanation.

Week 7: Mining the Web

1. The Structure of Websites: `html`
2. Requesting webpages with `requests` or `urllib`
3. Parsing `html` with `beautifulsoup`
4. Introduction to regular expressions (`regex`)

Students will learn the fundamentals of writing a script to automate web-based data collection. This will include a discussion of the legality and ethics of the method, when and how it should be employed, and the potential consequences of inappropriately applying it.

Students will learn the basics of the structure of every webpage; `html`, and how the library `beautifulsoup` can help them parse and navigate this in order to extract data from webpages. Students will also learn a basic introduction to regular expressions with the `re` library.

Week 8: Introduction to Natural Language Processing

1. What is NLP?
2. Syntactic Parsing with `spaCy`
3. Entity recognition and document similarity with `spacy`

Students will be introduced to an advanced and increasingly popular natural language processing library: `spaCy`, and discuss how it can be useful in their own research. Students will learn basic concepts in natural language processing, and the challenges of operationalising text. References for more advanced models will also be included in the readings.