week0_materials

July 7, 2020

1 Introduction

1.0.1 Course structure

This is a data science course in *Python3* (hereafter referred to as just *Python*) designed for participants with basic Python experience. The course will be run over **6 weeks** with the following structure.

- **Monday lectures** will start with information about Python syntax, the Jupyter notebook interface, and move through concepts such as how to write functions and handle data, using the *pandas* and *numpy* packages, how to calculate summary information from a data frame, and approaches to do plotting. Each lecture will conclude with an **assignment**.

- Participants will then be invited to review the materials presented on the lecture and complete the assignment during the rest of the week (Tuesday-Thursday).

- The trainers will provide a walk-through the assigments on **Friday recap** sessions. Questions from the participants are welcomed by e-mail or chat during the week or directly during the lecture or recap sessions.

1.0.2 Aims

The course will cover concepts and strategies for working with data more effectively in Python with the aim of:

- Writing **reusable** code, using functions and libraries
- Acquiring a working knowledge of **key concepts** which are prerequisites for advanced programming like writing classes to build objects
- Using Python **libraries** to explore data in files
- Creating **functions** and **modules** to write reusable code
- Manipulating data using **pandas** and **numpy**
- Visualising data

1.0.3 Course materials

The course materials (lectures, assignments and solutions) are accessible on GitHub:

https://github.com/semacu/data-science-python

We'd like you to follow along with the example code as we go through the materials together, and attempt the assignment to practice what you've learned.

1.0.4 Audience

This course is open to everyone with some basic knowledge of Python. We are so excited that you want to learn Python :) ! We will start with a brief recap on Python basic concepts and we will build up from there. You will set the pace and the amount of material that we will cover.

1.0.5 Feedback

Questions are welcome at any point by e-mail or *via* the chat in Microsoft Teams!

1.0.6 Python and Jupyter

Python is a general purpose programming language that is useful for writing scripts to work effectively and reproducibly with data. It was initially created by Guido van Rossum in 1991. Python is widely used in data science, bioinformatics and scientific computing, as well as in academia and industry.

It is available in all popular operating systems (Mac, Windows and Linux). The default Python installation comes with "batteries included" and the standard library (some of which we will see in this course) provides built-in support for lots of common tasks e.g. numerical & mathematical functions, interacting with files and the operating system … There is also a wide range of external libraries for areas not covered in the standard library, such as Pandas (the Python ANalysis DAta Library), Matplotlib (the Python plotting library) and the BioPython Library which provides tools for bioinformatics.

Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Jupyter supports execution environments and has developed and supported the interactive computing products e.g. Jupyter Notebook, which we will be using during this course.

The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, interactive data visualizations and explanatory text.

How to run Python? Python is an interpreted language, this means that your computer does not run Python code natively, but instead we run our code using the Python interpreter. There are three ways in which you can run Python code:

- Directly typing commands into the interpreter: Good for experimenting with the language, and for some interactive work
- Using a Jupyter Notebook: Great for experimenting with the language, and for sharing and learning
- Typing code into a file and then telling the interpreter to run the code from this file: Good for larger programs, and when you want to run the same code repeatedly

2 Installations

Before starting this course, you need to have Python3 and Jupyter installed on your computer. If you do not have these installed already, we recommend installing Anaconda (a complete programming environment including Python3 and Jupyter) by following the instructions below:

2.0.1 Windows & macOS

1. Follow this link to the AZ Software Store: http://azsoftwarestore.astrazeneca.net/ESD

2. You should see this home screen:

3. In the "Search Catalog" bar at the top, search for "anaconda". This should return two options, "Anaconda 5.3" and "Anaconda3 2019.10":

4. Click the "Add to Cart" button under the "Anaconda3 2019.10" program, and this should be added to your cart. **NB: make sure you have added the correct version of Anaconda to your cart ("Anaconda3 2019.10"), as the other version ("Anaconda 5.3") is not suitable for the content of this course**

5. Click the Cart icon on the top right of the screen, and a preview of the contents of your cart should appear:

6. Click the "View cart and checkout" button, and you should be taken to a summary of your cart:

7. Check that the "Receive ASAP" option is selected

8. Click the "Next" button

9. Click the "Submit" button

10. You should go through to a "Request Complete" screen. Anaconda will be installed on your computer within a few hours.

2.0.2 Linux

Follow these instructions for installing Anaconda on your linux distribution.

2.0.3 Anaconda installation check

Once you have installed Anaconda, it is a good idea to check that your installation is working properly:

1. Open the "Anaconda Navigator (Anaconda3)" program, and you should get this home screen:

2. Click the "Launch" button underneath the Jupyter Notebook icon

3. A tab should open on your browser, showing the Jupyter logo at the top and your file system below