COSC 2671 Social Media and Network Analytics

Lab 1

Python Revision & Jupiter Notebook Setup

Learning outcomes:

- Revise usage of Python and Jupyter Notebooks
- Familarise with Anaconda for installing packages if using your own laptops

Requirements:

- A PC with Internet connection and Python 3 installed

Resources:

- This lab worksheet

Introduction

In this lab, we will revise how to open Jupyter Notebooks and perform some basic Python commands. We will be using Python (and Jupyter Notebooks) extensively in this course. If you are using your own devices, we also practice/revise how to install packages, as we use a number of additional ones in this course.

Anaconda & Jupyter Notebook (for those using their own machines)

If you haven't done so, please download Anaconda from https://www.anaconda.com/distribution/ . For this course we will be using Python 3, but you are welcome to use Python 2, but unfortunately we won't have resources to support this if you run into trouble (the difference between versions are not great, but there are some significant ones).

Install it, which will install Jupyter notebooks as well as Conda for package management.

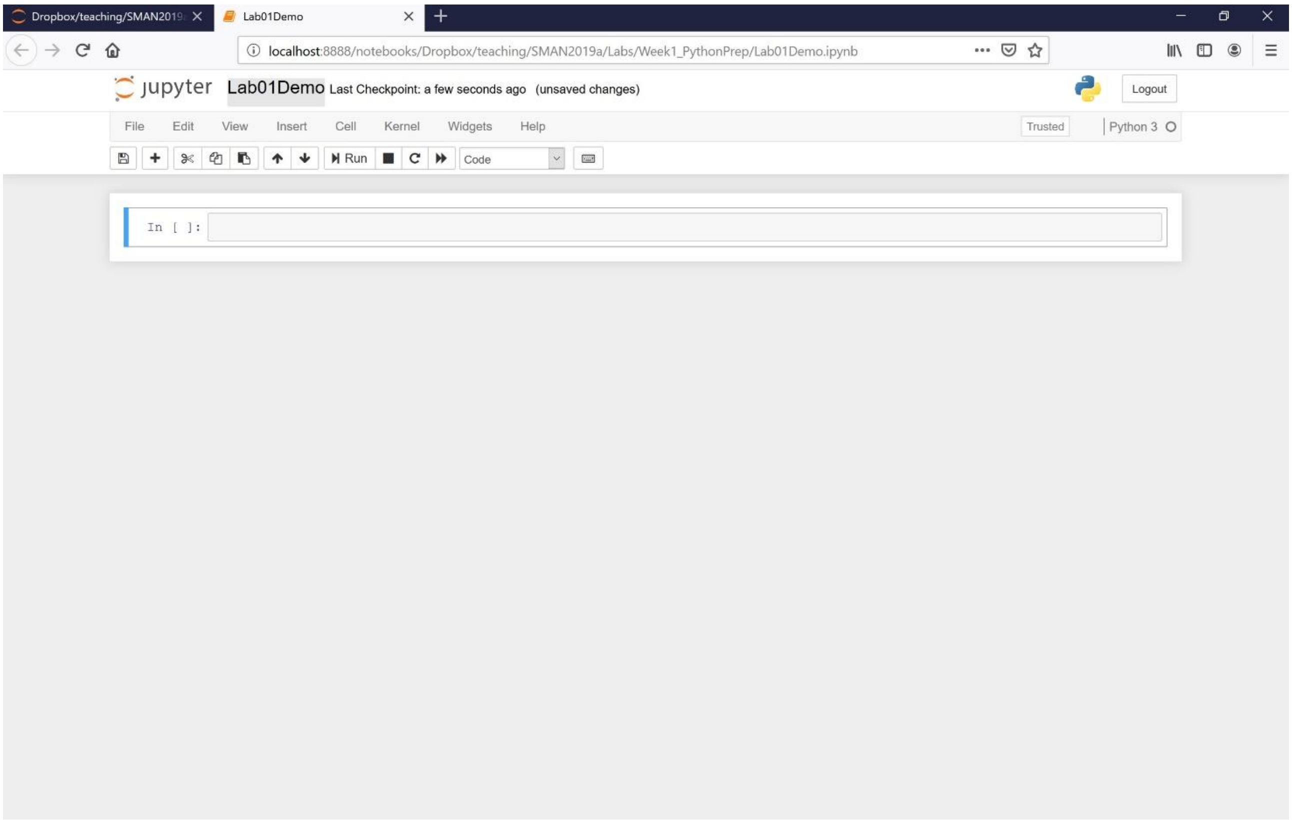
Note, we assume you have installed Python 3 already.

More materials for Jupyter Notebook:

- A tutorial:
    - Jupyter Notebook Tutorial: The Definitive Guide (https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook)

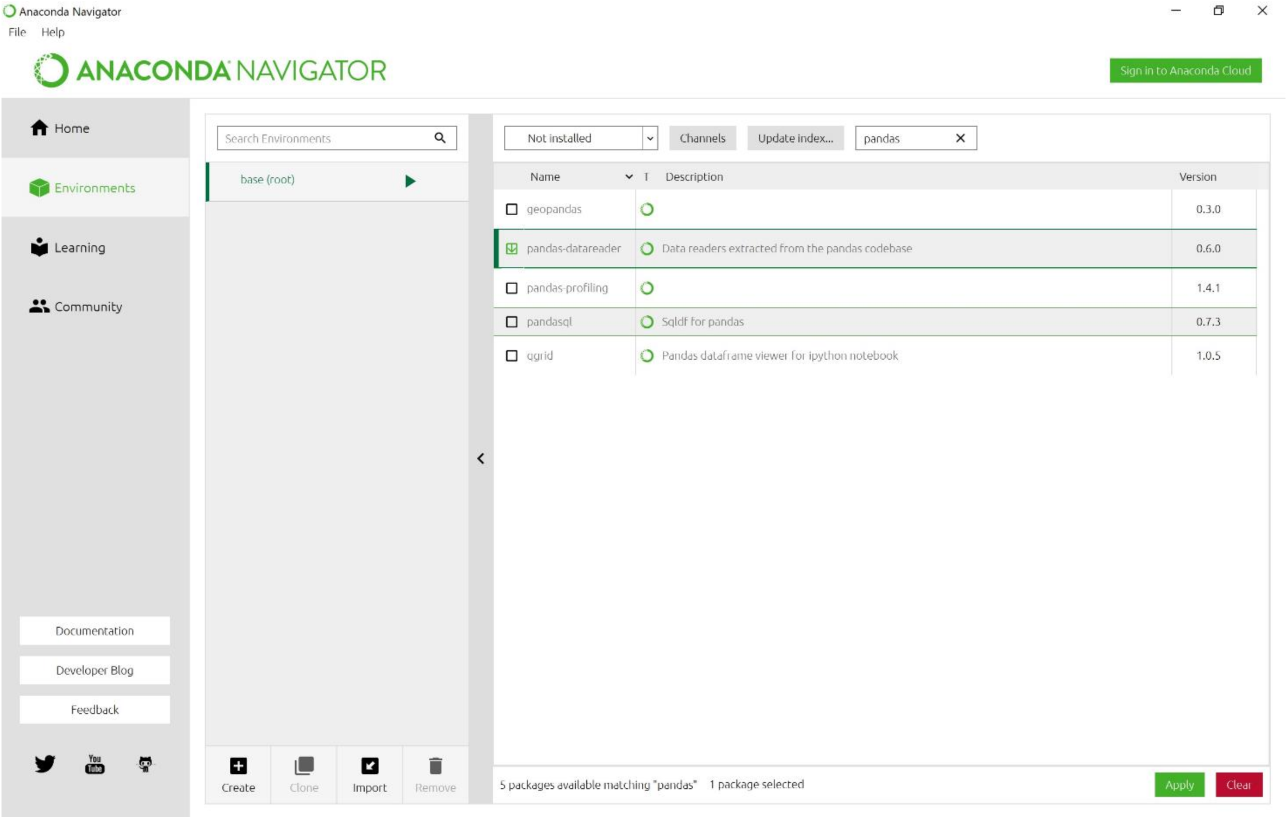
Opening Anaconda & Jupiter Notebook

Start up Anaconda and create a new Jupiter notebook terminal. If on lab machines, you don't need to startup Anaconda, directly find Jupiter Notebooks from the start menu and click on it. After creating a terminal should see something like this:


Dropbox/teaching/SMAN2019
Lab01Demo
localhost:8888/notebooks/Dropbox/teaching/SMAN2019a/Labs/Week1_PythonPrep/Lab01Demo.ipynb
jupyter
Lab01Demo Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

Jupyter Notebooks runs an iPython shell (interactive Python). It is a very useful approach to doing prototyping and analysing data, common tasks for data science. But as a side note, been able to write scripts is also important to develop tools that can be reused multiple times.

For those using their own machines, an additional useful note is how to install new packages. In Anaconda navigator, on the left hand side bar, click on "Environments". This shows a list of installed Python packages. To install additional ones, that would likely be the case for this course, select "not installed" and type in the package name you wanted install – in the following screenshot, we typed in "pandas":



	Name	T	Description	Version
□	geopandas	○		0.3.0
☑	pandas-datareader	○	Data readers extracted from the pandas codebase	0.6.0
□	pandas-profiling	○		1.4.1
□	pandasql	○	Sqldf for pandas	0.7.3
□	qgrid	○	Pandas dataframe viewer for ipython notebook	1.0.5

5 packages available matching "pandas"   1 package selected

The click on apply and it will install the package for use.

Python Revision

For revision, we are going to go through a few commands and their output:

Now enter the following commands to print out “Hello Machine Learning, here I come!”:

```
In [1]: print(“Hello Social Media, here I come!”)
```

This should output the string to standard output.

Python is an interpreted language, and you can use it as such like a calculator:

```
In [2]: 1 + 2
```

What does it display?

Python can store values into variables, which are assigned a type according to what is assigned to it. Type in the following:

```
In [3]: a = “Hello World”
```

This stores the string “Hello World” in the variable a. We can print that out as follows:

```
In [4]: print(a)
```

In addition to strings, Python has other types, such as integers:

```
In [5]: b = 3 * 4
```

This will store the integer ‘12’ into b.

Python has two major builtin data structures (there are other useful ones we will use for machine learning, but for now it is important to understand the built-in ones), one is a list and another is a dictionary. We will focus on list for now (please read up about dictionaries). A list In Python is essentially a sequence of elements, that we can reference using indices (thing of this as similar to a Java array, but can dynamically increase in size). First we create a list:

```
In [6]: c = [1,2,3]
```

Creates a list with 3 elements of 1, 2, 3. To access an element, we use an index. Python list index starts from 0:

```
In [7]: c[0]
```

What does this display?

Finally, to exit your iPython session, simply type

```
In [8]: exit
```

For the rest of the lab and to learn Python, please go through the following or online resources described in the Machine Learning Canvas shell.

You can learn more about iPython from the official documentation. There are also many tutorials available online.

Introductory guide that those new to programming or need a refresher:
https://www.programiz.com/python-programming

If you familiar with programming and scripting languages in general, consider looking at this site, which goes through the major parts of Python:
https://www.stavros.io/tutorials/python/