Introduction to Python

Installation

Matthew Hielsberg

Department of Mathematics (https://www.math.tamu.edu) & Institute of Data Science (https://tamids.tamu.edu/)

This tutorial series uses Jupyter Notebook as the primary means of providing both instruction and examples. To utilize the provided notebooks you can use an online-only setup or a local installation.

- Online-only Setup
  - Use Google Colab, which requires no local installation and allows you to use all of the provided notebooks.
- Local Installation (only need one of the following)
  - (Option 1) Install Python and Jupyter Notebook
  - (Option 2) Install Anaconda, which includes Jupyter Notebook, Jupyter Hub and other tools.

For Windows users wanting a local installation we recommend using Anaconda as it is a single install that includes all of the tools necessary to get started.

The sections below give comments/instructions with links to get you started with each of these options.

Online-only Setup

The online-only setup is by far the fastest/easiest means of getting started with the notebooks.

Google Colab

To use Python in Google Colab no installation is required, simply open Google Colab (https://colab.research.google.com/notebook#create=true) in your browser. More details on using Google Colab will be given in the chapter on IDE's.

Local Installation (Option 1)

Install Python

As of October 2021 the current version of Python is 3.10.0, however much of this tutorial series was written prior to the release of the current version and should work with version 3.7 or later.

The Python website (python.org) has download links, documentation, and tutorials, including a Beginners Guide (https://wiki.python.org/moin/BeginnersGuide/) with instructions on how to check for and if necessary install Python on your system, see Downloading Python (https://wiki.python.org/moin/BeginnersGuide/Download).

Before installing Python you should verify whether or not you already have Python installed. Several systems come with some version of Python pre-installed. Typically from the command line you can run `python3 --version` or `python --version`. If either of these commands come back with version 3.7 or better then you should be good to go. If you need to install Python we have included the installation instructions for the major systems below.

- **Linux**:
    1. Using your system's pacakge manager (e.g. dnf, yum, apt) install `python3` and `python3-devel` (Red Hat, CentOS, and Fedora) or `python3.x` and `python3.x-dev` (Debian and Ubuntu), where `x` is replaced with the version you want, e.g. `apt install python3.9 python3.9-dev`
    2. Alternatively, you can go to the source code download page (https://www.python.org/downloads/source/) and download the latest tarball. Compilation and installation instructions should be found within the downloaded pacakge.
- **Mac**: Macports or Homebrew may be used to install the latest Python, but will not be covered here. Instead we recommend getting the Python installer directly from Python.org (https://www.python.org/downloads/).
    1. The download page (Python.org (https://www.python.org/downloads/)) should automatically detect your OS and present you with a button to download the latest version of Python. Click this button to begin the download.
    2. If the page incorrectly identifies you OS you can either click the link directly below this button for your specific OS or click `Downloads` in the top menu on the webpage to find the appropriate download link. Older versions and direct download links can be found on the Downloads for MacOS (https://www.python.org/downloads/macos/) page.
    3. Run the downloaded installer package (typically double click the pkg) to start the installation, and follow the prompts.
- **Windows**:
    1. The download page (Python.org (https://www.python.org/downloads/)) should automatically detect your OS and present you with a button to download the latest version of Python. Click this button to begin the download.
    2. If the page incorrectly identifies you OS you can either click the link directly below this button for your specific OS or click `Downloads` in the top menu on the webpage to find the appropriate download link. Older versions and direct download links can be found on the Downloads for Windows (https://www.python.org/downloads/windows/) page.

3. Run the downloaded installer package (typically double click the exe) to start the installation, and follow the prompts. If given the option (typically a checkbox on the first page) to 'Add Python 3.X to PATH', select the option as this can make using Python easier.

Install Jupyter Notebook

Project Jupyter (https://jupyter.org) is an open source project desinged to enable interactive computing for multiple programming languages. The installation page (https://jupyter.org/install) and detailed installation guide (https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html) provides several options for installation. For those using `pip`, installation should be as easy as running `pip install notebook`, and then running with `jupyter notebook`. For other setups/systems we encourage you to view the links above for more details on installing for your specific system. Below are more detailed instructions for specific systems.

- **Linux (Ubuntu 20.04)**:

    1. By default Ubuntu 20.04 comes with Python 3, however we need to ensure that we use the correct version and that `pip` is installed. We can see which python we are using with `which python` which should return the path to python and `python --version` should give a version 3.8 or better. It may be that the alias *python* has not been set on your system in which case `python3` should be used instead. By default `pip` is not installed (you can verify on your system using `which pip` or `which pip3`. To install pip use the following:

        ```
        $ sudo apt update
        $ sudo apt install python3-pip
        $ pip --version
        ```

        The last line should output the version 20.0.2 or better from python 3.8 or better.
    2. Install Jupyter Notebook. We recommend installing this for the current user (or in a virtual environment) rather than using sudo, but this will of course be up to you and/or your system administrator. To install Jupyter Notebook for the current user run:

        ```
        $ pip install notebook
        ```

    3. If running `which jupyter` in the terminal now displays the path to the jupyter executable, then you can skip to step #5.
    4. If you do not use sudo when installing with pip, it will attempt to install only for the current user. In which case you may need to update your `$PATH` to include the local bin directory. During the installation of Jupyter notebook if there is a warning stating that "The script xxxx is installed in '/home/username/.local/bin' which is not on PATH", then this path will need to be added to your environment. In Ubuntu, by default, the file ~/.profile should already contain code to add the ~/.local/bin directory to your PATH upon login. In this case, to update the PATH you can either logout/login or run `source ~/.profile` in the terminal. However, in the event you need to manually update your PATH do the following:

A. Copy the path (all the warnings should give the same path) from the warnings during the install of notebook.
B. In an editor of your choice open/create the file `.bash_profile` in your home directory. If you are already using `.profile` or another file/setup for your PATH modifications, then feel free to continue using those instead. If you are uncertain on which file to use, it is never a bad idea to check out `man bash`. The following instructions will assume that you are using .bash_profile and the editor *vi*: in the terminal execute `vi ~/.bash_profile`
C. Once in vi, use the arrow keys to move to the bottom of the file (or press capital 'G', `<shift> G`). Once at the bottom of the file press lowercase `o` to create a new line and enter insert mode, so that we may insert text.
D. Add the following text to the file, replacing the path after the ':' with the one from the warnings output during the installation of Jupyter Notebook: `export PATH=$PATH:/home/username/.local/bin`.
E. Press `ESC` to exit insert mode.
F. Type `:wq` and hit the enter/return key to write (save) the file and quit vi.
G. To update the PATH execute `source ~/.bash_profile` in the terminal.
H. To ensure that your PATH has been updated type `echo $PATH` in the terminal and verify that the new directory path is included in the ouput. You should also now see the path to the jupyter executable output when you run `which jupyter`.

5. Jupyter Notebook may now be started with

```
$ jupyter notebook
```

6. It is important to not close the terminal window until you are finished using Jupyter notebook and have saved all changes made to your open notebooks.

- **Mac (Big Sur)**:

1. Before installing with `pip` you will need to install the command line developer tools using

```
$ xcode-select --install
```

Follow the prompts to complete the installation.

2. Verify that you are using the correct `pip`, in the event that you have mulitple Python installations (e.g. HomeBrew, MacPorts, etc). Using `which python` and/or `which pip` will tell you the path of those executables. Also, keep in mind that pip may not have an alias installed by default, and you may need to use `pip3` instead (again you can verify this using `which`).
3. Upgrade pip to the latest version. At the time of writing argon2-cffi did not install with pip version 19.2.3, and upgrading to pip version 21.1.3 corrected the issue. To upgrade pip use the following:

```
$ sudo pip3 install --upgrade pip
```

In the example, we assume that pip3 is the executable name (i.e. no alias 'pip' is available), and yes, *--upgrade pip*, without the '3' in pip is correct.

4. Install Jupyter Notebook. We recommend installing this for the current user (or in a virtual environment) rather than using sudo, but this will of course be up to you and/or your system administrator. To install Jupyter Notebook for the current user run:

```
$ pip3 install notebook
```

5. If running `which jupyter` in the terminal now displays the path to the jupyter executable, then you can skip to step #7.
6. If you do not use sudo when installing with pip, it will attempt to install only for the current user. In which case you may need to update your `$PATH` to include the local bin directory. If you are using HomeBrew or MacPorts for your Python installation than it is likely that you may skip this step as they typically modify the path upon install. However, if during the installation of Jupyter notebook there is a warning stating that "The script xxxx is installed in '/Users/username/Library/Python/3.8/bin' which is not on PATH" (the displayed path may be differ in your setup), then this path will need to be added to your environment. To update your PATH do the following:
   A. Copy the path (all the warnings should give the same path) from the warnings during the install of notebook.
   B. In an editor of your choice open/create the `.zprofile` file (or `.bash_profile` if you are using bash). If you are already using `.zshrc` or `.zshenv` for your PATH modifications, then feel free to continue using them. If you are uncertain on which to use then it is never a bad idea to check out `man zshall` (or `man bash`). The following instructions will assume that you are using .zprofile and the editor *vi*: in the terminal execute `vi ~/.zprofile`
   C. Once in vi, use the arrow keys to move to the bottom of the file (or press capital 'G', `<shift> G`). Once at the bottom of the file press lowercase `o` to create a new line and enter insert mode, so that we may insert text.
   D. Add the following text to the file, replacing the path after the ':' with the path from the warnings output during the installation of Jupyter Notebook: `export PATH=$PATH:/Users/username/Library/Python/3.8/bin`
   E. Press `ESC` to exit insert mode.
   F. Type `:wq` to write (save) the file and quit vi.
   G. To update the PATH execute `source ~/.zprofile` in the terminal.
   H. To ensure that your PATH has been updated type `echo $PATH` in the terminal and verify that the new directory path is included in the ouput. You should also now see the path to the jupyter executable output when you run `which jupyter`.
7. Jupyter Notebook may now be started with

   ```
   jupyter notebook
   ```

8. It is important to not close the terminal window until you are finished using Jupyter notebook and have saved all changes made to your open notebooks.

- **Windows 10**:

1. In order to install Jupyter Notebook, we must have Python installed, if you already have Python then you may skip this step.
   A. In a web browser visit www.python.org
   B. Moving the cursor to hover over the Downloads menu item opens a submenu that should auto-detect your OS and display a "Download for Windows" section with a button to download the latest version of Python. Click the button to download Python.
   C. Once the download completes, open the python installer file to begin the installation.
   D. In the first window presented by the installer, be sure to enable the option at the bottom to "Add Python to PATH" (if available), and then select "Install Now".
   E. Windows User Account Control will prompt you and ask if you want to allow the installer to make changes to your device, click yes.

F. Once the installer has finished, click "Close".

2. Open a Command Prompt. To do this, type "cmd" in the search area of the task bar, and with Command Prompt highlighted on the left select "Open" on the right side for the Command Prompt.
3. To verify that Python is installed and that we have a recent version type `py --version` at the prompt. This should output something like "Python 3.9.6", although your version may differ.
4. Do the same for pip and type `pip --version` to veriy that it too is available.
5. Install Jupyter Notebook:

```
$ pip install notebook
```

6. Jupyter Notebook may now be started with

```
$ jupyter notebook
```

7. It is important to not close the command prompt window until you are finished using Jupyter notebook and have saved all changes made to your open notebooks.

Local Installation (Option 2)

Install Anaconda

Anaconda is an alternative product that can help new users to manage and maintain separate environments for R, Python, Jupyter Notebook and more. The Anaconda package also includes tools like Spyder and JupyterLab, and can be downloaded from the Anaconda (https://www.anaconda.com/products/individual) webpage. The Anconda Installation Documentation (https://docs.anaconda.com/anaconda/install/) webpage contains detialed instructions on how to install Anaconda for each system so we will not duplicate those instructions here.