Instructions for Installing and Running Jupyter Notebooks

The Jupyter Notebook is an open source application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. The notebooks run in your browser, as well as in your local machine. Anaconda conveniently installs in your local machine the Jupyter Notebook, Python, and other commonly used packages for data science.

Installation:

1. Download Anaconda. We recommend downloading the latest Python 3.7 version.

2. Install the version of Anaconda which you downloaded, following the instructions on the download page. Congratulations, you have installed Jupyter Notebook.

3. There are two Python libraries, Plotly and Graphviz that are not contained in Anaconda but you will need to use in later assignments. In fact, Graphviz is open source graph visualization software and it provides a python interface with the same name.
   Please follow the instructions to install them.
   Mac users:
   First, to open terminal, click “Applications >> Utilities >> Terminal”.
   Second, to install Plotly, a python package which we will use in the visualization assignment, please open your terminal and type in “pip install plotly” in the command line. “pip” is a Python package manager preinstalled with Anaconda.
   Third, to install Graphviz, you need to have homebrew installed first. Homebrew is a package manager for macOS and you can install it by pasting a command at your terminal. Click the hyperlink to see the command. Once Homebrew is installed, type in “brew install graphviz” in your terminal. You may be prompted to input password during installation. It is the login password for your Mac.

At last, to install the graphviz interface, type in “pip install graphviz” in the command line of your terminal.

Windows users:
First, search for Anaconda Prompt on your PC and open it.

Second, to install Plotly, a python package which we will use in the visualization assignment, please type in “pip install plotly” in the command line of the Anaconda Prompt window. “pip” is a Python package manager preinstalled with Anaconda.

Third, to install Graphviz, please input the following two commands separately. Conda is a packaging tool and installer that aims to handle library dependencies outside of the Python packages as well as the Python packages themselves. The installation might take some time, please be patient.

```
conda install -c anaconda graphviz
conda install -c anaconda python-graphviz
```

If you encounter errors when installing the libraries mentioned above, please try to run the following commands in the Anaconda Prompt window. These commands will update anaconda and pip, which might be helpful to your problem. If you still encounter errors after trying them, please ask your TA for help.

```
conda update anaconda
python -m pip install --upgrade pip
```

Running Jupyter Notebook:

1. To run Jupyter Notebook, open Anaconda Navigator (it might take a few seconds) and click “Launch Jupyter Notebook”.

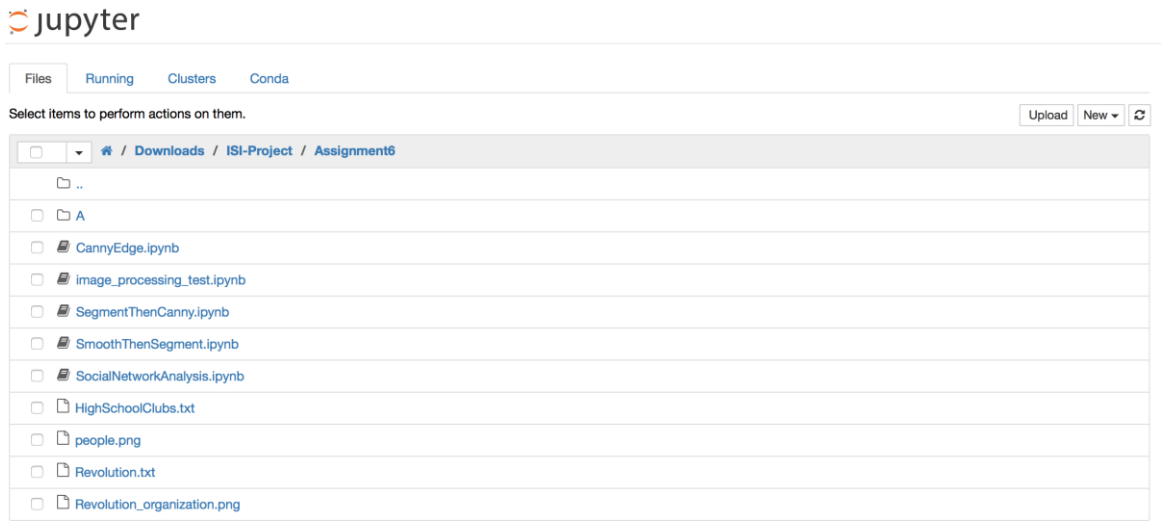
Another way to run Jupyter Notebook is to input “jupyter notebook” in the command line of your terminal. To open terminal, Mac users can click “Applications >> Utilities >> Terminal” and Windows users can click “Start Menu >>Programs >>Anaconda” and look for an icon called Anaconda Prompt.

2. When you connect to Jupyter Notebook for **the first time**, you will be prompted to copy and paste a URL into your browser. Below is a screenshot of terminal. The URL is in the red rectangle. (Jupyter has changed its configuration steps and probably you do not need to execute this step.)

```
MacBook-Pro:~        $ jupyter notebook
[I 13:47:03.635 NotebookApp] JupyterLab alpha preview extension loaded from /Use
rs/       /anaconda2/lib/python2.7/site-packages/jupyterlab
JupyterLab v0.27.0
Known labextensions:
[I 13:47:03.637 NotebookApp] Running the core application with no additional ext
ensions or settings
[I 13:47:03.646 NotebookApp] Serving notebooks from local directory: /Users/
eng
[I 13:47:03.646 NotebookApp] 0 active kernels
[I 13:47:03.646 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/?token=81a3b563993f2ca7ada9682792a1dbda8dce7417dd9c2bf0
[I 13:47:03.646 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 13:47:03.647 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=81a3b563993f2ca7ada9682792a1dbda8dce7417dd9
c2bf0
0.97. execution error: "http://localhost:8888/tree?token=b09eba5c28e3bb8bb58539d
bfd3ec00344cab41cbdc49184" doesn't understand the "open location" message. (-170
8)
```

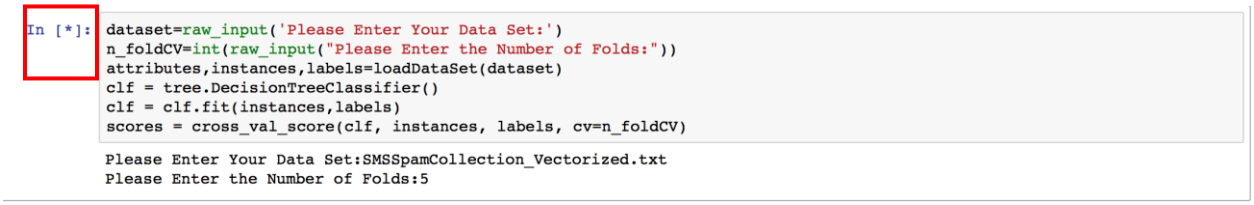
3. When you run Jupyter Notebook, it will open a webpage mirroring your local directory and thus we call it the Jupyter directory. Click the notebook you want to run and you will get a new window or a tab for that notebook. An example of Jupyter directory:



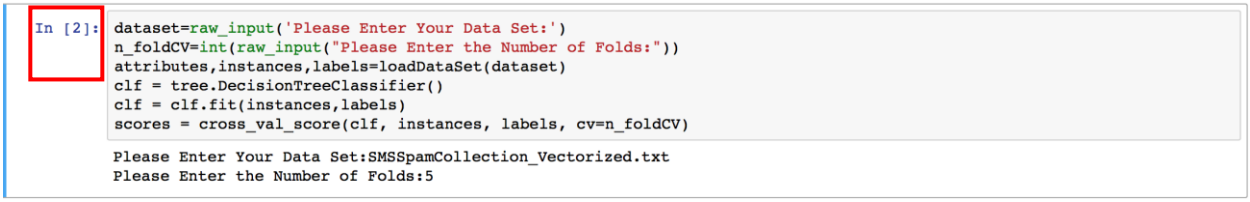
How to run cells in Jupyter Notebook?

1. You can run the notebook document step-by-step (one cell a time) by pressing **shift + enter** or by clicking either the "Play" (the black triangle) button in the toolbar, or Cell | Run in the menu bar.

2. When a cell is running for a long time, which happens a lot, there would be an asterisk showing at the left side of the cell. If a cell runs quickly, you might not be able to notice the asterisk. **Please run the next cell only if the previous cell has been run and the asterisk changes to an index (shown below).** An index of a cell is not a cell number but the number of runs in a notebook.



For example:



3. When you are running a notebook, it will automatically save your progress in your local disk and update your “ipynb” document in your local disk. You can know when the notebook is autosaved by reading the title of the notebook. For example,

4. When you want to run different datasets with the same notebook, we would recommend you to make a copy of the notebook and name the two notebooks with different suffixes from the datasets. Click “File>>Make a Copy…” in your Jupyter Notebook’s interface and then you will get a copy of the notebook. Then run the different datasets with the new notebook you get. Or you can copy and paste the cells you want to rerun and use the new datasets in these new cells. To create a new cell, click the "+" sign in the shortcut area and copy and paste the code that you want to rerun into it.

Saving and Closing Jupyter Notebooks:

1. When you are running a notebook and want to save any changes you have done, click the “File>>Save and Checkpoint” in your Jupyter Notebook’s interface on the browser.

2. When you are completely done with a notebook, you can close the tab. In addition, you should shut it down. This is done in the Running panel as

   shown below.

3. **After you are done with all the notebooks you were running, please check if all of them were shut down in the Jupyter Running panel.** When a notebook is running, it’s green in the directory. When it’s not running, it should be grey. After shutting down all the notebooks, close the terminal window for Jupyter Notebook. **Although it seems that Jupyter Notebook runs in a browser, it actually runs at your machine as well. If you don’t shut it down, it will keep running and using the computing power of your machine.**

FAQs:

1. What is a kernel? Why sometimes Jupyter Notebook tells me the kernel is interrupted?

A 'kernel' is a program that runs and introspects the user's code. You can treat it like a black box. Generally, when a kernel is interrupted, Jupyter Notebook will restart the kernel by itself. If it doesn't, click on the menu Kernel >>Restart. This can be useful to start over a computation from scratch (e.g. variables are deleted, open files are closed, etc...).

2. My cell is running for a long time and it never ends. What should I do?

You can click "Kernel>>Interrupt" or the black square button in the toolbar to manually interrupt the execution of the code cell and then rerun all the relevant cells one by one. If you keep encountering the same problem, check your dataset and turn to your TA for help.

3. I get an error message from the notebook. What am I supposed to do?

The error message may be long but actually you just need to read the last line of it. It will tell you what kind of problems you have. Try to solve the error by yourself. You can search for solutions on Google. If it doesn't work, turn to your TA for help.

For instance, (zoom in to see details):

```
In [8]: img=loadIMG()
        CannyEdge(img)

Pleanse input the image filename:yg-cedar.png

---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-8-36c876e70bda> in <module>()
----> 1 img=loadIMG()
      2 CannyEdge(img)

<ipython-input-6-f3269d9415c3> in loadIMG()
      7     filename=raw_input("Pleanse input the image filename:")
      8     #filename = os.path.abspath('rocks.jpg')
----> 9     img = io.imread(filename)
     10     if len(img.shape)!=2:
     11         img = color.rgb2gray(img)

/Users/gil/anaconda2/lib/python2.7/site-packages/skimage/io/_io.pyc in imread(fnam
e, as_grey, plugin, flatten, **plugin_args)
     59
     60     with file_or_url_context(fname) as fname:
---> 61         img = call_plugin('imread', fname, plugin=plugin, **plugin_args)
     62
     63     if not hasattr(img, 'ndim'):

/Users/gil/anaconda2/lib/python2.7/site-packages/skimage/io/manage_plugins.pyc in
call_plugin(kind, *args, **kwargs)
    209                                (plugin, kind))
    210
--> 211     return func(*args, **kwargs)
    212
    213

/Users/gil/anaconda2/lib/python2.7/site-packages/skimage/io/_plugins/pil_plugin.py
c in imread(fname, dtype, img_num, **kwargs)
     33     """
     34     if isinstance(fname, string_types):
---> 35         with open(fname, 'rb') as f:
     36             im = Image.open(f)
     37             return pil_to_ndarray(im, dtype=dtype, img_num=img_num)

IOError: [Errno 2] No such file or directory: 'yg-cedar.png'
```

4. I clicked some text cells and the cells don't come back to how them looked before. What can I do?

Rerun the cell by pressing **shift+enter**.

5. I want to write down some notes in a notebook. How can I do that?

There are four types of cells in Jupyter Notebook and we used code cell and markdown cell most frequently. All the text in a notebook were written by Markdown Language and you can write down your notes in the same way. For those of you who are curious about Markdown, you will have a notebook to practice how to use it. Also, here is a cheat sheet of Markdown Language.

6. I encountered warning messages while running cells. Should I care about it?

If you see messages like the one below, just ignore them as long as they don't prevent you from getting correct outputs. Such warning messages are shown in pink.

```
/Users/[redacted]/anaconda/lib/python2.7/site-packages/matplotlib/font_manager.py:273: UserWarning: Matplotlib is buildin
g the font cache using fc-list. This may take a moment.
  warnings.warn('Matplotlib is building the font cache using fc-list. This may take a moment.')
```

7. In the visualization notebooks, I see a cell called "Run Interact". Is this a different type of cell?

The "Run Interact" is actually a Jupyter Widget that provides interactive inputs for the visualizations. It is not a cell by the definition given by Jupyter Notebook and therefore the results you get from it will not be automatically saved after the notebook is close. To save each visualization you get, click "Download plot as a png" button on the upper right side of the visualization.

Click here!