Bio5075 Fundamentals of Biostatics: Pre-course Computational Primer

*****Before class starts on Sep. 14, you *must* have Jupyter Notebook installed and working on the laptop that you will bring to class.*****

If you don't have a laptop that you can bring to class, we can provide you with a loaner. Please email the instructors at bio5075-admin@lists.genetics.wustl.edu.

Use this primer to install the software and make sure it's running. If you have trouble, just email us and we'll be happy to help. We will use this software on the first day of class, so it is critical that you install the software and complete this primer beforehand.

Why Jupyter Notebook?

In this class, we learn some basic programming skills and use them to computationally solve statistical problems. For all class activities and homework, we will use the Python programming language within an application called **Jupyter Notebook.** Jupyter Notebook runs inside your web browser and makes it easy to write, run and document your code. You can think of Jupyter Notebook as the computational equivalent of a lab notebook – an organized record of what you did, why you did it, and what the results were. For the instructors, Jupyter Notebook makes is easy to distribute, collect, and grade the homework.

This primer will show you how to:

1) Install the Anaconda distribution of Python and Jupyter Notebook.
2) Launch Jupyter Notebook and move around your filesystem.
3) Do basic tasks in Jupyter Notebook.

Part 1: Install the Anaconda Python Distribution

Anaconda is a free, pre-packaged collection of software that is very useful for scientific computing. It includes Python itself, the Jupyter Notebook application, and a pile of specialized software packages for specific tasks like performing statistical analyses and making plots. These software tools are conveniently collected into a free "distribution" called Anaconda, making it easy to install everything you need.

To download and install the Anaconda distribution, go to this web page:
https://www.anaconda.com/distribution/

Click **"Download"**, which should take you to the download page for your operating system (macOS, Windows, or Linux). Download the installer for **Python 3.8** (make sure you select the correct installer for your operating system).

Once the installer downloads, follow these instructions to install the software: https://docs.anaconda.com/anaconda/install/

If you have trouble with the installation, contact the instructors.

Part 2: Launching Jupyter Notebook

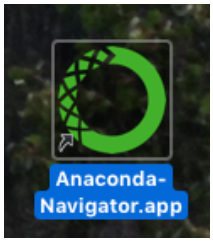
Jupyter Notebook is an interface that you will use to interact with data files and Python software libraries, much like you use a web browser to access files and software on web servers. In fact, Jupyter Notebook runs inside of your web browser (though you will use it to access programs and data files on your local hard drive, not the web).

In **Part 2**, you'll learn how to launch Jupyter Notebook and navigate to different directories or folders on your hard drive. You'll create a class folder, where you will store your files for this class.

Step 1: Launch Jupyter Notebook

There are two ways to launch Jupyter Notebook – the easy way (using your mouse), and the slightly less easy way (using your keyboard).

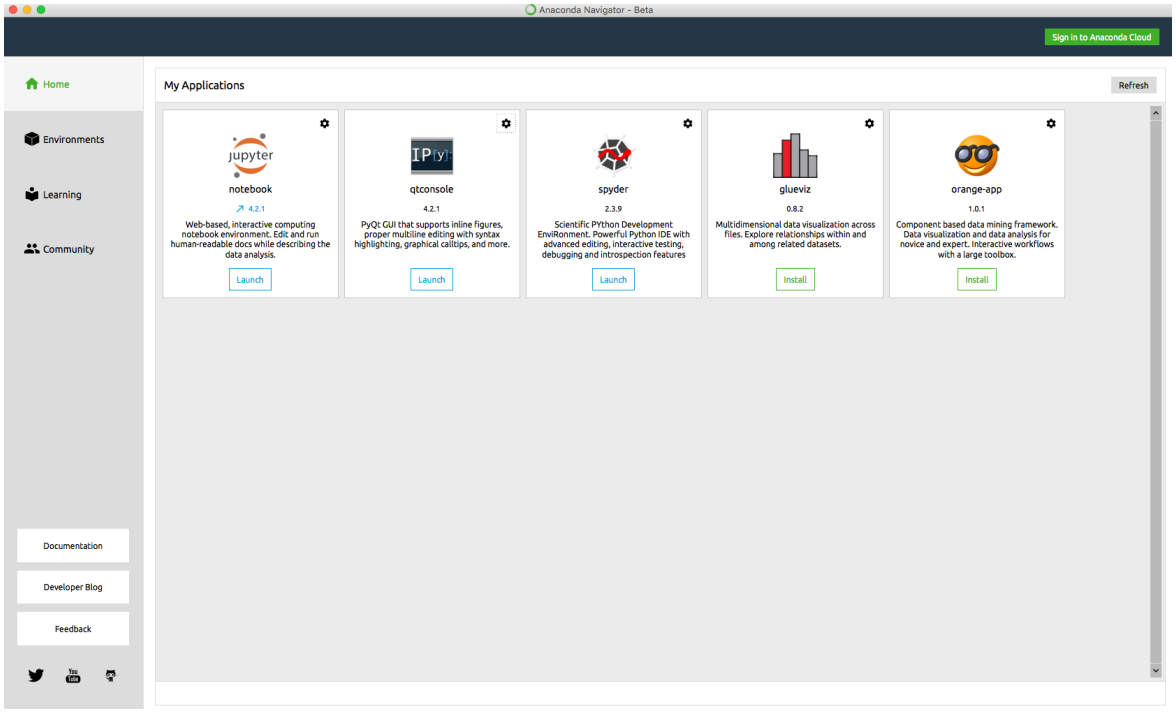
Easy way: Double-click the **Anaconda Navigator** icon that was created by the Anaconda software installer, which may be on your Desktop:



If you don't see this icon on your Desktop, it might be in your home folder or the root folder of your hard drive, inside a folder called **anaconda.** (You can find this folder using the search

function in Windows Explorer or Spotlight on a Mac). Once you find it, drag the icon onto your Desktop and keep it there for easy access.

The Navigator application will open a window that looks something like this (your version wil look slightly different):



To launch a Jupyter Notebook session, click the blue **Launch** button in the **Jupyter Notebook** panel. Jupyter Notebook will open in your default web browser. If you have an older laptop, this may take a few moments; be patient. Once the browser window opens, you're ready to go.

Slightly less easy way: If you're comfortable working from the command line, you may prefer to launch Jupyter Notebook from there. First, open a terminal window.

*(To open a terminal window on a Mac, launch the **Terminal** app in your **Application/Utilities** folder. You can also type **Terminal** in the Spotlight search field to quickly find and launch Terminal. On a Windows computer, select **Start -> Run** and type **cmd** in the field.)*

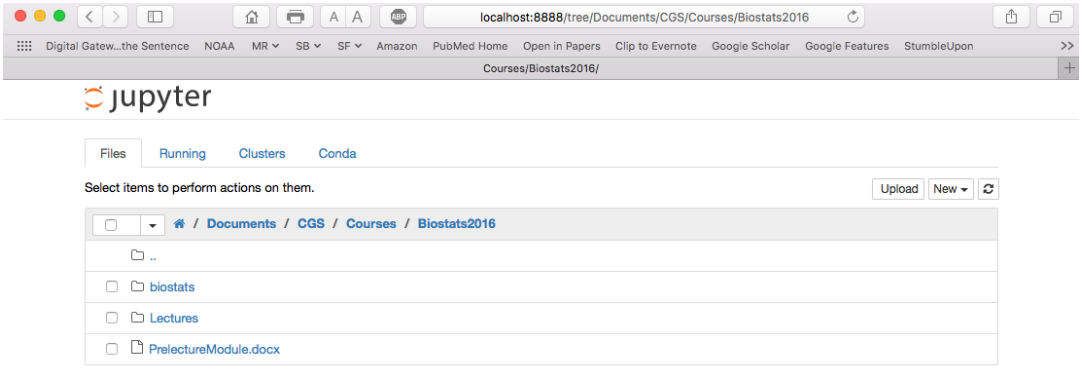
From the command line, type **jupyter notebook** and hit return. A Notebook session will launch in your default web browser.

NOTE: Before you launch jupyter notebook from the command line, be that you are in your home directory. Once you launch jupyter, you will only be able to navigate to folders within the folder from which you launched jupyter.

For more on using the command line, see this tutorial for Mac and Linux: https://swcarpentry.github.io/shell-novice/ and this for Windows: http://www.ntu.edu.sg/home/ehchua/programming/howto/CMD_Survival.html.

Step 2: Create your class folder

Jupyter Notebook will open in your browser window and present you with a page that looks something like this:



This may look like a web page. But rather than viewing files on the web, you're using your browser to view folders on your hard drive. Jupyter Notebook behaves like a web page. Each folder name is a link, including all of the blue folder names in the gray header bar. The folder at the top of the list with two periods next to it is a link to move up one level from your current folder.

Click the blue links to move around your hard drive's filesystem. Go ahead and try it out – in this class, never be afraid to explore!

Next, create a class folder to hold all your files for this class. This folder will store data files (usually plain text .txt files) and Notebook (.ipynb) files that we'll work with in class.

To create your class folder, click the blue links to navigate to wherever you plan to keep your class folder. Then click on the **New** drop-down menu in upper right corner, and select folder. Name your folder 'biostats' or something similar. You're now ready to write some code.

To learn more about Jupyter Notebook, here is the official beginner's guide:

http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

Part 3: Get to know Jupyter Notebook

Using Python

Python is a programming language that you use in two ways. First, you can use it to write **scripts** – complete programs that are saved as standalone files that you run again and again, each time you need them. Second, Python can be used **interactively**, much like a graphing calculator: you type a command and Python immediately gives you an answer.

Running scripts is a good way to automate tasks that you do routinely, such as processing data files or performing the same analysis over and over. If you take the Genomics course next semester, this is how you will use Python.

In this course, we will use Python interactively, which is more useful for exploratory data analysis. You can use Python interactively from the command line, but we'll use Jupyter Notebook as our Python interface. As you'll see, a major advantage of Jupyter Notebook is that

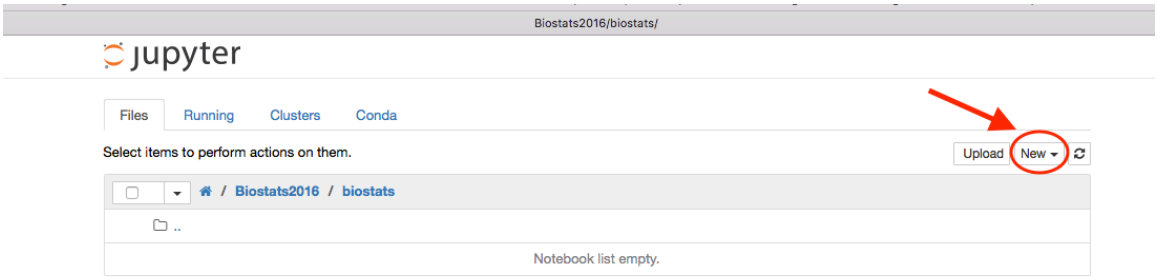
it serves essentially as a lab notebook for computational work. The notebook format makes it easy to organize, annotate, and save your code and data analyses – which is important for making your computational work reproducible by yourself and others.

In **Part 3,** you'll try some basic Jupyter Notebook functions. If you are interested in getting more practice before class, check out some of the tutorials on the web:

http://nbviewer.jupyter.org/github/jupyter/notebook/blob/master/docs/source/examples/Notebook/Notebook Basics.ipynb

https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook - gs.ISKAsU8

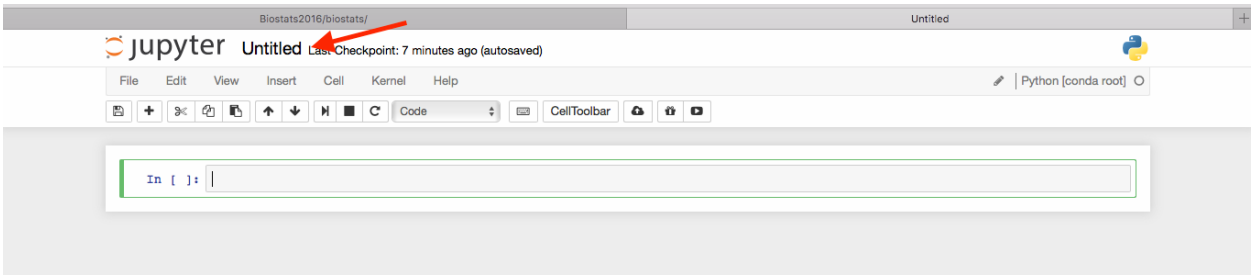
1. Create a new notebook:
When you launch Jupyter Notebook (**Part 2**), a new browser window opens the **Jupyter dashboard**. The dashboard shows you the files in the current folder (under the **Files** tab, top left, which opens by default).



You can't write or run code from the *dashboard*. To work with code, you need a *notebook*. To create a new notebook, click on the **New** button (top right), and select "Python" under Notebooks. (You installation may say "Python 3" – chose that one.) A notebook will open in a new browser window.

2. Name your notebook:

In the notebook window, you'll see that the default name is "Untitled." Click on "Untitled" to open a dialog box. Then rename your notebook "Primer." This will create a file called "Primer.ipynb" in your class folder.



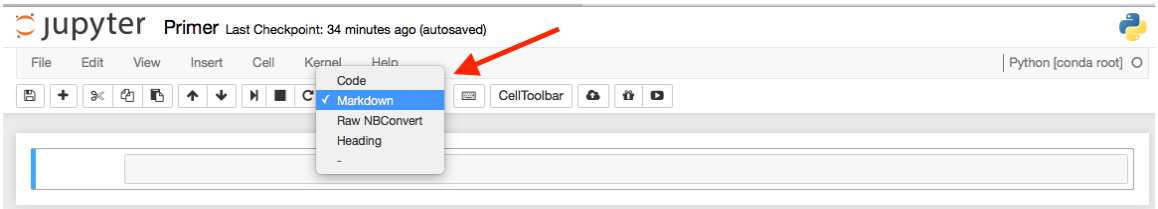
3. Using the notebook:

A notebook consists of a sequence of **cells**. The first cell of your new notebook is the blank box below the menu (with the `In [ ]:` on the left side – this will display the *input* line number).

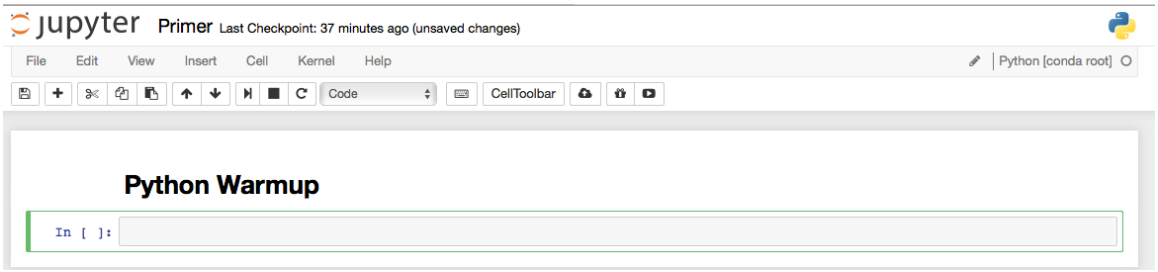
You will type either Python commands or regular text into these cells, depending on what type of cell it is. We will use two types of cells:

> **Code cells:** Here you type Python code, which is then run when you hit the keys ***shift-enter***.
>
> **Markdown cells:** These cells are for text to organize and comment your code. You can create headers, descriptions, etc., just like you would in your lab notebook. You type text, then press ***shift-enter*** to exit the cell.

Let's start out by writing a header for the first part of our notebook. Click on the drop-down menu that says "Code", in the toolbar at the top of the notebook window, and select "Markdown":



Now click inside the blank cell and type: `# Python Warmup`. Then press both ***shift*** and ***enter***. Your notebook will now look like this:



To edit the header, just double-click on it, and the cell will re-open. Try it – add another **#** to the text in the cell, to get `## Python Warmup` (followed by *shift enter*). What happened to the header formatting?

After you finish entering text, a new code cell appears below. You can tell that it is a code cell because the drop-down menu above says "code", and it has `In [ ]:` on the left. The next few a few exercises show how to work with code cells.

Math: Doing basic math is much like doing it on a calculator (with some important exceptions which we'll discuss during the course).

In the next cell type `10 + 6` and hit *shift enter*. Python will return the answer on an output line below the code cell. (Notice what happens to the `In [ ]:` on the left.)

Print: As you write code, after some operations you'll want Python to display the result on your screen. To display a result, use the use the `print()` function.

To see how the print function works, type `print('Hello World')` into a cell and hit *shift enter*.

How is using **`print()`** different from just typing **`Hello World`**? Try the following and see what happens:

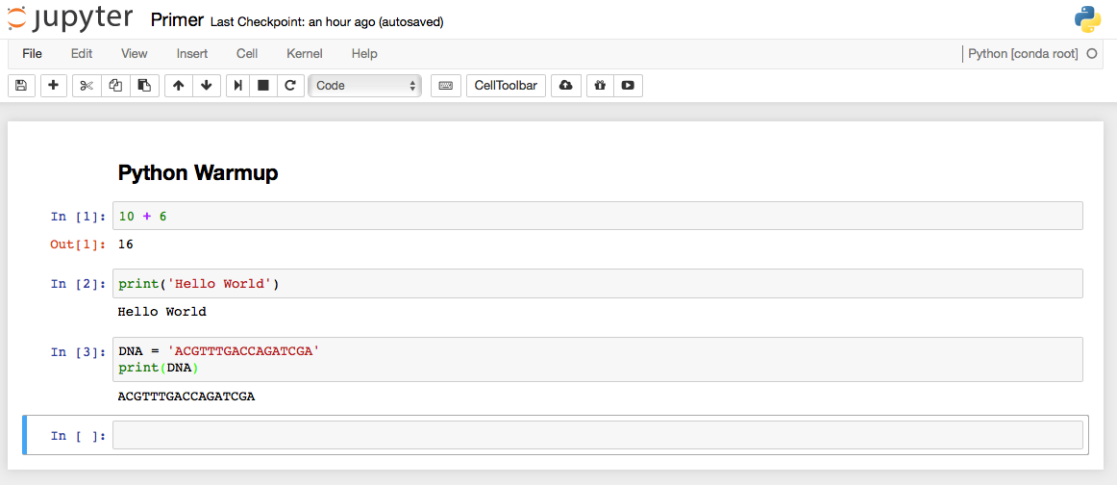
1) Type **`'Hello World'`** then *shift enter*. How is the output different from **`print('Hello World')`**?
2) Type **`Hello World`** (without the single quotes) then *shift enter*. What happens now?

Confused? In class, we'll discuss why this happens in class.

Let's try another example of **`print()`**. Type **`DNA = 'ACGTTTGACCAGATCGA'`** (or any random sequence**`)`** into the next cell. This assigns the sequence to a variable named **DNA**.

Then start a new line in the same cell by hitting ***enter*** (not *shift enter*).

Finally, type **`print(DNA)`**. Now hit *shift enter*, and python will print the value of the variable **dna** to your screen:



Make a plot: There are some powerful graphic functions available for Python. Here we will make a simple graph.

To make a plot, we need to load some additional software tools. These come from a Python *package* called **matplotlib**.

Type the following two commands. (Hit *enter* after the first line, type the second line, then hit *shift enter* to execute both commands.)

```
%matplotlib inline
import matplotlib.pyplot as plt
```

The first command prepares Jupyter Notebook to display plots in the notebook window. The second line loads a subset of the **matplotlib** package called **pyplot**. Note that no output is displayed after running these commands, but a new blank cell appears. Unless you get an error message, assume the commands ran successfully.

Now let's plot some fake data. Type the following lines into the new cell, then hit *shift enter*:

```
drug_concentration = [0,1,2,4,8]
response = [0.2,15.1,39.8,90.5,100.2]
```

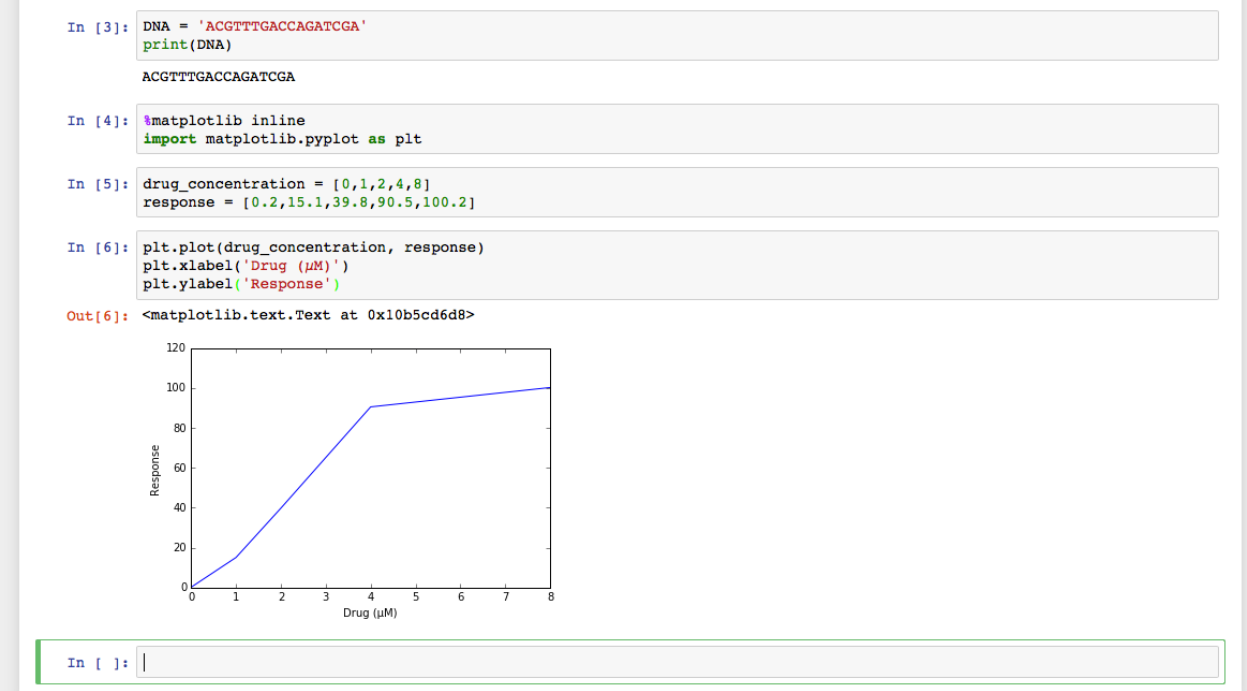
These commands create *lists* of data values and assign those lists to variables. We'll discuss lists and variables during the first class. To plot the data and create axis labels, type the following commands, then type *shift enter* to run them:

```
plt.plot(drug_concentration, response)
plt.xlabel("Drug (μM)")
plt.ylabel("Response")
```

Pro tip: *You don't have to type out the full names of the variables – type the first few letters, then hit tab and see what happens. This is called* ***tab completion****.*

Note the prefix used with functions like `plt.ylabel`. This means we're calling the function `ylabel` from the `pyplot` package.

You should see a plot like this (it may take a moment to appear):



If you would like to go back and change any of your code, it's easy – just click inside the cell you want to change and begin typing.

4. Save and quit: Everything in your Notebook session will be saved as a notebook file called Primer.ipynb.

To save, select "Save and Checkpoint" from the Jupyter **File** menu. Alternately, click the button with the floppy disk icon (right under **File**). Then select "Close and Halt" from the file menu. The notebook browser window will close, and you'll return to the Jupyter dashboard.

Notice that in the dashboard you now have a new file in your course folder called “Primer.ipynb.” Everything you did in the notebook is saved there. During your next session, you can re-open the notebook and start right where you left off.

NOTE: Notebook files must be opened within the Jupyter dashboard. If you find the file in the Finder or Windows Explorer and then double-click it, you won’t see anything intelligible.

To completely quit Jupyter, click the **Quit** button in the upper right (next to **Logout**). Congratulations, you’ve completed your first Jupyter Notebook session! If it seems a little overwhelming, don’t worry – you’ll get plenty of practice with notebooks in class. If this seems easy, don’t worry – the class will get harder!