**Jupyter Notebook**

[Jupyter](http://jupyter.org/) notebooks are a way that you can have code, text, images, and math all live together in harmony. This concept can be called "[Literate Proramming](https://en.wikipedia.org/wiki/Literate_programming)", where all code that is written also has comments or an explanation of *why* it was written (which is not always obvious). In particular, the Jupyter notebook allows you to write some code, run (evaluate) it, and see the output all in once. This is called a "read-eval-print loop" or [REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop).

What we will need:

* Anaconda
* R
* Jupyter

To use jupyter notebook we first need to install anaconda if you don't already have it.

**Install Anaconda Python, R, and Jupyter to your computer**

You can look up instructions on how to download Anconda depending on your operating system, it is pretty straight foward:

<https://docs.continuum.io/anaconda/install>

Anaconda:

<http://jupyter.readthedocs.org/en/latest/install.html#new-to-python-and-jupyter>

You will know if your conda is working properly by typing either:

**which conda**

and getting an output that looks like this

**/Users/Pedro\_Torres/anaconda2/bin/anaconda**

If you do not get an output but see an anaconda2 folder in your directory you will have to set the path to anaconda on your own like this:

1. Lets pretend I type ‘pwd’ and it the output is ‘/Users/Pedro\_Torres/’ and then type ‘ls’ and one of the folders in my directory is ‘anaconda2’
2. Type ‘cd anaconda2/bin/’
3. You are in your anaconda folder, now if I type pwd output **For my computer** should be ‘/Users/Pedro\_Torres/anaconda2/bin’
4. Set your path to anaconda by typing one of two codes:
   1. echo $PATH = “PATH:/Users/Pedro\_Torres/anaconda2/bin/anaconda”
   2. export PATH = “/Users/Pedro\_Torres/anaconda2/bin/anaconda:$PATH”
5. Now when you type which anaconda you should get something that looks like this:

**/Users/Pedro\_Torres/anaconda2/bin/anaconda**

**You are good to continue! Be advised that the above code is for MAC different approaches must be taken when working with a PC.**

**Install some more packages**

• R

* IRKernel - Use R in Jupyter notebook
* [seaborn](http://stanford.edu/~mwaskom/software/seaborn/) (nicer plots in Python)

By specifying these packages, we're also specifying all their dependencies (the other packages that each of these requires)

conda install --channel r r r-irkernel seaborn

Here's that big command broken down:

* conda - the base command (like how git was the base command you used for git stuff). Every conda subcommand is actually conda-subcommand e.g.conda-create under the hood, but we use it with just the spaces for convenience.
* install - The conda subcommand to create an environment
* --channel r - A "channel" is a URL to a folder that contains packages that you can install. Anaconda doesn't come with the R channel by default so we have to specify it here.
* r r-irkernel seaborn - The packages to install.

The output is quite big, it will look something like this:

$ conda install --channel https://conda.anaconda.org/r r r-irkernel seaborn

Warning: could not import binstar\_client (invalid token (pkg\_resources.py, line 44))Fetching package metadata: ......

Solving package specifications: ...................................................................

Package plan for installation in environment /home/ucsd-train01/anaconda3:

The following packages will be downloaded:

package | build

---------------------------|-----------------

glib-2.43.0 | 2 7.4 MB r

decorator-4.0.6 | py35\_0 6 KB defaults

numpy-1.10.2 | py35\_0 5.8 MB defaults

pyzmq-15.1.0 | py35\_0 782 KB defaults

requests-2.9.0 | py35\_0 647 KB defaults

setuptools-19.1.1 | py35\_0 348 KB defaults

cairo-1.12.18 | 6 594 KB defaults

conda-3.19.0 | py35\_0 180 KB defaults

scipy-0.16.1 | np110py35\_0 23.3 MB defaults

harfbuzz-0.9.35 | 6 1.1 MB r

pango-1.36.8 | 3 796 KB r

nbconvert-4.1.0 | py35\_0 275 KB defaults

r-base-3.2.2 | 0 20.6 MB r

seaborn-0.6.0 | np110py35\_0 257 KB defaults

r-3.2.2 | 0 2 KB r

r-base64enc-0.1\_3 | r3.2.2\_0 25 KB r

r-boot-1.3\_17 | r3.2.2\_0 575 KB r

r-cluster-2.0.3 | r3.2.2\_0 466 KB r

r-codetools-0.2\_14 | r3.2.2\_0 45 KB r

r-digest-0.6.8 | r3.2.2\_2 93 KB r

r-foreign-0.8\_66 | r3.2.2\_0 220 KB r

r-jsonlite-0.9.17 | r3.2.2\_0 927 KB r

r-kernsmooth-2.23\_15 | r3.2.2\_0 84 KB r

r-lattice-0.20\_33 | r3.2.2\_0 698 KB r

r-magrittr-1.5 | r3.2.2\_1 154 KB r

r-mass-7.3\_45 | r3.2.2\_0 1.0 MB r

r-nnet-7.3\_11 | r3.2.2\_0 99 KB r

r-repr-0.3 | r3.2.2\_0 44 KB r

r-rpart-4.1\_10 | r3.2.2\_0 861 KB r

r-rzmq-0.7.7 | r3.2.2\_3 60 KB r

r-spatial-7.3\_11 | r3.2.2\_0 122 KB r

r-stringi-1.0\_1 | r3.2.2\_0 10.7 MB r

r-survival-2.38\_3 | r3.2.2\_0 4.4 MB r

r-uuid-0.1\_2 | r3.2.2\_0 18 KB r

r-class-7.3\_14 | r3.2.2\_0 82 KB r

r-irdisplay-0.3 | r3.2.2\_0 23 KB r

r-matrix-1.2\_2 | r3.2.2\_0 3.1 MB r

r-nlme-3.1\_122 | r3.2.2\_0 2.0 MB r

r-stringr-1.0.0 | r3.2.2\_0 78 KB r

r-evaluate-0.8 | r3.2.2\_0 39 KB r

r-mgcv-1.8\_9 | r3.2.2\_0 1.8 MB r

r-irkernel-0.5 | r3.2.2\_1 71 KB r

r-recommended-3.2.2 | r3.2.2\_0 707 B r

------------------------------------------------------------

Total: 89.7 MB

The following NEW packages will be INSTALLED:

cairo: 1.12.18-6 defaults

glib: 2.43.0-2 r

harfbuzz: 0.9.35-6 r

libgcc: 4.8.5-1 r

ncurses: 5.9-4 r

pango: 1.36.8-3 r

pcre: 8.31-0 defaults

pixman: 0.32.6-0 defaults

r: 3.2.2-0 r

r-base: 3.2.2-0 r

r-base64enc: 0.1\_3-r3.2.2\_0 r

r-boot: 1.3\_17-r3.2.2\_0 r

r-class: 7.3\_14-r3.2.2\_0 r

r-cluster: 2.0.3-r3.2.2\_0 r

r-codetools: 0.2\_14-r3.2.2\_0 r

r-digest: 0.6.8-r3.2.2\_2 r

r-evaluate: 0.8-r3.2.2\_0 r

r-foreign: 0.8\_66-r3.2.2\_0 r

r-irdisplay: 0.3-r3.2.2\_0 r

r-irkernel: 0.5-r3.2.2\_1 r

r-jsonlite: 0.9.17-r3.2.2\_0 r

r-kernsmooth: 2.23\_15-r3.2.2\_0 r

r-lattice: 0.20\_33-r3.2.2\_0 r

r-magrittr: 1.5-r3.2.2\_1 r

r-mass: 7.3\_45-r3.2.2\_0 r

r-matrix: 1.2\_2-r3.2.2\_0 r

r-mgcv: 1.8\_9-r3.2.2\_0 r

r-nlme: 3.1\_122-r3.2.2\_0 r

r-nnet: 7.3\_11-r3.2.2\_0 r

r-recommended: 3.2.2-r3.2.2\_0 r

r-repr: 0.3-r3.2.2\_0 r

r-rpart: 4.1\_10-r3.2.2\_0 r

r-rzmq: 0.7.7-r3.2.2\_3 r

r-spatial: 7.3\_11-r3.2.2\_0 r

r-stringi: 1.0\_1-r3.2.2\_0 r

r-stringr: 1.0.0-r3.2.2\_0 r

r-survival: 2.38\_3-r3.2.2\_0 r

r-uuid: 0.1\_2-r3.2.2\_0 r

seaborn: 0.6.0-np110py35\_0 defaults

The following packages will be UPDATED:

conda: 3.18.8-py35\_0 defaults --> 3.19.0-py35\_0 defaults

decorator: 4.0.4-py35\_0 defaults --> 4.0.6-py35\_0 defaults

nbconvert: 4.0.0-py35\_0 defaults --> 4.1.0-py35\_0 defaults

numpy: 1.10.1-py35\_0 defaults --> 1.10.2-py35\_0 defaults

pyzmq: 14.7.0-py35\_1 defaults --> 15.1.0-py35\_0 defaults

requests: 2.8.1-py35\_0 defaults --> 2.9.0-py35\_0 defaults

scipy: 0.16.0-np110py35\_1 defaults --> 0.16.1-np110py35\_0 defaults

setuptools: 18.5-py35\_0 defaults --> 19.1.1-py35\_0 defaults

Proceed ([y]/n)?

Press "y" to proceed

Fetching packages ...

r-base64enc-0. 100% |######################################################| Time: 0:00:00 347.71 kB/s

r-digest-0.6.8 100% |######################################################| Time: 0:00:01 81.73 kB/s

r-jsonlite-0.9 100% |######################################################| Time: 0:00:18 52.38 kB/s

r-magrittr-1.5 100% |######################################################| Time: 0:00:00 436.33 kB/s

r-repr-0.3-r3. 100% |######################################################| Time: 0:00:00 75.39 kB/s

r-rzmq-0.7.7-r 100% |######################################################| Time: 0:00:00 294.29 kB/s

r-stringi-1.0\_ 100% |######################################################| Time: 0:00:03 3.71 MB/s

r-uuid-0.1\_2-r 100% |######################################################| Time: 0:00:00 247.64 kB/s

r-irdisplay-0. 100% |######################################################| Time: 0:00:00 329.61 kB/s

r-stringr-1.0. 100% |######################################################| Time: 0:00:00 366.92 kB/s

r-evaluate-0.8 100% |######################################################| Time: 0:00:00 381.25 kB/s

r-irkernel-0.5 100% |######################################################| Time: 0:00:00 342.46 kB/s

Extracting packages ...

[ COMPLETE ]|#########################################################################| 100%

Linking packages ...

[ COMPLETE ]|#########################################################################| 100%

**My First Jupyter Notebook**

Start jupyter notebook server:

There are two ways to do this

1. Open up terminal and simply type ‘ jupyter notebook’ a browser should automatically open up and it will say jupyter on top

2. You could also just have jupyter running in the background of your terminal this is done by opening up you terminal and typing’ jupyter notebook —no-browser -port #### (any four numbers) & ’

something similar to the bottom will show up.

[I 12:06:38.857 NotebookApp] The port 8889 is already in use, trying another random port.

[I 12:06:38.867 NotebookApp] Serving notebooks from local directory: /Users/Pedro\_Torres/Desktop

[I 12:06:38.867 NotebookApp] 0 active kernels

[I 12:06:38.867 NotebookApp] The Jupyter Notebook is running at: http://localhost:8890/

[I 12:06:38.867 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirma

then you can copy the http address http://localhost:8890/ put it on your browser, i prefer safari, and voila.

Finished and want to kill program:

**$ ps**

PID TTY TIME CMD

6593 pts/44 00:00:00 bash

12583 pts/44 00:00:02 jupyter-noteboo

12592 pts/44 00:00:00 links

14198 pts/44 00:00:01 jupyter-noteboo

14208 pts/44 00:00:00 ps

$kill -9 14198

**Possible errors**

***"The port #### is already in use, try another random port"***

This can happen if you ran jupyter notebook and killed it but wanted to run it again. This is because even though the jupyter notebook process died, there's still a "zombie" (real computer term) process running. To kill it, do:

$ ps

PID TTY TIME CMD

6593 pts/44 00:00:00 bash

12583 pts/44 00:00:02 jupyter-noteboo

12592 pts/44 00:00:00 links

14198 pts/44 00:00:01 jupyter-noteboo

14208 pts/44 00:00:00 ps

See which processes are associated with jupyter and then use kill -9 to stop them ("kill" stops the process and -9 means the meanest form like premeditated murder of the program)

$ kill -9 12583

$ kill -9 14198

***Big weird screen opens up***

]

Then you forgot the --no-browser flag. Try again:

jupyter notebook --no-browser --port ####

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_