

# Tools

Keith Perkins

# Outline

- Optional useful tools
- Anaconda
- Virtual Environments
- Jupyter Lab

# Useful tools

**Optional Content** - Useful for keeping track of things your interested in



Weava Highlighter - PDF & Web

Offered by: [weavatools.com](http://weavatools.com)



**Evernote**

Remember everything important.



Mendeley

**Mendeley Reference  
Manager for Desktop**



Cornell University

arXiv.org

**arxiv-sanity-lite**

A much lighter-weight arxiv-sanity from-scratch re-write. Periodically polls arxiv API for new papers. Then allows users to tag papers of interest, and recommends new papers for each tag based on SVMs over tfidf features of paper abstracts. Allows one to search, rank, sort, slice and dice these results in a pretty web UI. Lastly, arxiv-sanity-lite can send you daily emails with recommendations of new papers based on your tags. Curate your tags, track recent papers in your area, and don't miss out!

I am running a live version of this code on [arxiv-sanity-lite.com](http://arxiv-sanity-lite.com).


# Anaconda

<https://docs.anaconda.com/anaconda/install/index.html>

See the installation section for your OS

# Anaconda- helpful bits

Go to anaconda cheatsheet on course website - gives helpful commandline utilities

CONDA 4.6 CHEAT SHEET	
	Take a conda test drive at <a href="https://bit.ly/tryconda">bit.ly/tryconda</a> . Windows, macOS, Linux: Same commands for all platforms.
For full documentation of any command, add <code>--help</code> to the command. EXAMPLE: <code>conda create --help</code>	
Getting Started	
Verify Conda is installed, check version number	<code>conda info</code>
Update Conda to the current version	<code>conda update -n base conda</code>
Update all packages to the latest version of Anaconda. Will install stable and compatible versions, not necessarily the very latest.	<code>conda update anaconda</code>
Working with Environments	
Create a new environment named ENVNAME with specific version of Python and packages installed.	<code>conda create --name ENVNAME python=3.6 "PKG1&gt;7.6" PKG2</code>
Activate a named Conda environment	<code>conda activate ENVNAME</code>
Activate a Conda environment at a particular location on disk	<code>conda activate /path/to/environment-dir</code>
Deactivate current environment	<code>conda deactivate</code>
List all packages and versions in the active environment	<code>conda list</code>
List all packages and versions in a named environment	<code>conda list --name ENVNAME</code>
List all revisions made within the active environment	<code>conda list --revisions</code>
List all revisions made in a specified environment	<code>conda list --name ENVNAME --revisions</code>
Restore an environment to a previous revision	<code>conda install --name ENVNAME --revision REV_NUMBER</code>
Delete an entire environment	<code>conda remove --name ENVNAME --all</code>

```
keith@keith-Precision-5540: ~  
File Edit View Search Terminal Help  
(data301) >conda info --envs  
# conda environments:  
#  
base /home/keith/anaconda3  
CTEnviro /home/keith/anaconda3/envs/CTEnviro  
data301 * /home/keith/anaconda3/envs/data301  
fastbook /home/keith/anaconda3/envs/fastbook  
juplabTEST /home/keith/anaconda3/envs/juplabTEST  
xeus-python /home/keith/anaconda3/envs/xeus-python  
  
(data301) >
```

# Anaconda- virtual environments

“A named, isolated, working copy of Python that that maintains its own files, directories, and paths so that you can work with specific versions of libraries or Python itself without affecting other Python projects.

Virtual environments make it easy to cleanly separate different projects and avoid problems with different dependencies and version requirements across components.”\*

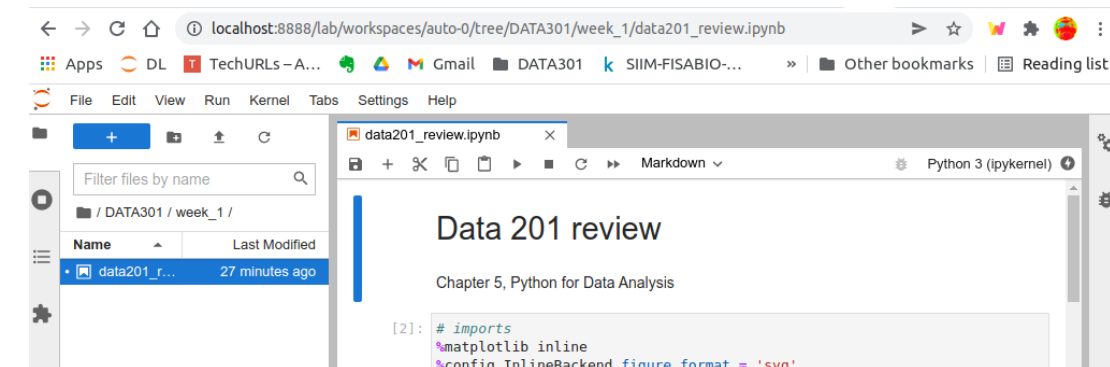
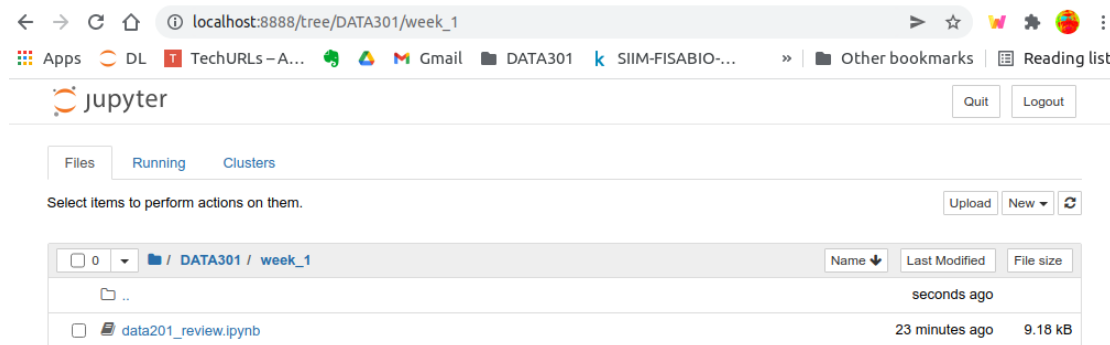
```
#run these at a command line
conda update -n base conda                #update anaconda
conda create -n data301 python=3.9 anaconda #create a new environment
conda activate data301                    #switch to it

#when done with this environment you can exit with
#conda deactivate
```

They also come in handy when you want to share your configuration with others (see “Sharing an Environment” at <https://docs.conda.io/>)

\* <https://uoa-ereseach.github.io/ereseach-cookbook/recipe/2014/11/20/conda/>

# Jupyter Lab



Jupyter Notebook: The old way, launch from commandline, Automatically starts in browser

```
(data301) >jupyter notebook
```

Jupyter Lab: The new way, launch from commandline, Automatically starts in browser

```
(data301) >jupyter lab
```

Can show multiple files at a time  
Lots of extensions (careful they are not reviewed)  
Seems to be the way Jupyter is going

But..

Debugging is primitive (ipykernel – uses default python, Or python in virtual environment that we launched Jupyter lab from)