

# Software For Data Science

SETH RUSSELL – SETH.RUSSELL@CUANSCHUTZ.EDU

ACCORDS DATA SCIENCE PROGRAM, D2V ANALYTICS CORE

[HTTPS://GITHUB.COM/MAGIC-LANTERN/BIOS6642](https://github.com/magic-lantern/bios6642)

# Jupyter Notebooks

Greatest Invention of the Modern Age:  
**spell check**

By NASA, ESA, and A. Simon (NASA Goddard), edited by  
PlanetUser -  
<https://www.nasa.gov/feature/goddard/2019/hubble-new-portrait-of-jupiter>, Public Domain,  
<https://commons.wikimedia.org/w/index.php?curid=81259729>



# Jupyter Notebooks



# Jupyter features

- ▶ Jupyter: Help > Keyboard Shortcuts
- ▶ Jupyterlab: Settings > Advanced Settings Editor > Keyboard Shortcuts
- ▶ Command Mode:
  - ▶ <Enter>: Switch to Edit mode
  - ▶ <Shift> <Enter>: run cell, select below
  - ▶ Y : change cell to code
  - ▶ M : change cell to markdown
  - ▶ A : insert cell above
  - ▶ B : insert cell below
  - ▶ X : cut selected cells
  - ▶ C : copy selected cells
  - ▶ V : paste cells below
  - ▶ Z : undo cell deletion
  - ▶ S: Save and Checkpoint
  - ▶ <Shift> L: Toggle Line Numbers

# Jupyter features

- ▶ Jupyter: Help > Keyboard Shortcuts
- ▶ Jupyterlab: Settings > Advanced Settings Editor > Keyboard Shortcuts
- ▶ Edit Mode:
  - ▶ `<ESC>`: Switch to Command mode
  - ▶ `<TAB>` : code completion or indent
  - ▶ `<Shift> <TAB>` : tooltip
  - ▶ `<Ctrl> Z`: undo
  - ▶ `<Shift> <Enter>`: run cell, select below
  - ▶ `<Ctrl> S`: Save

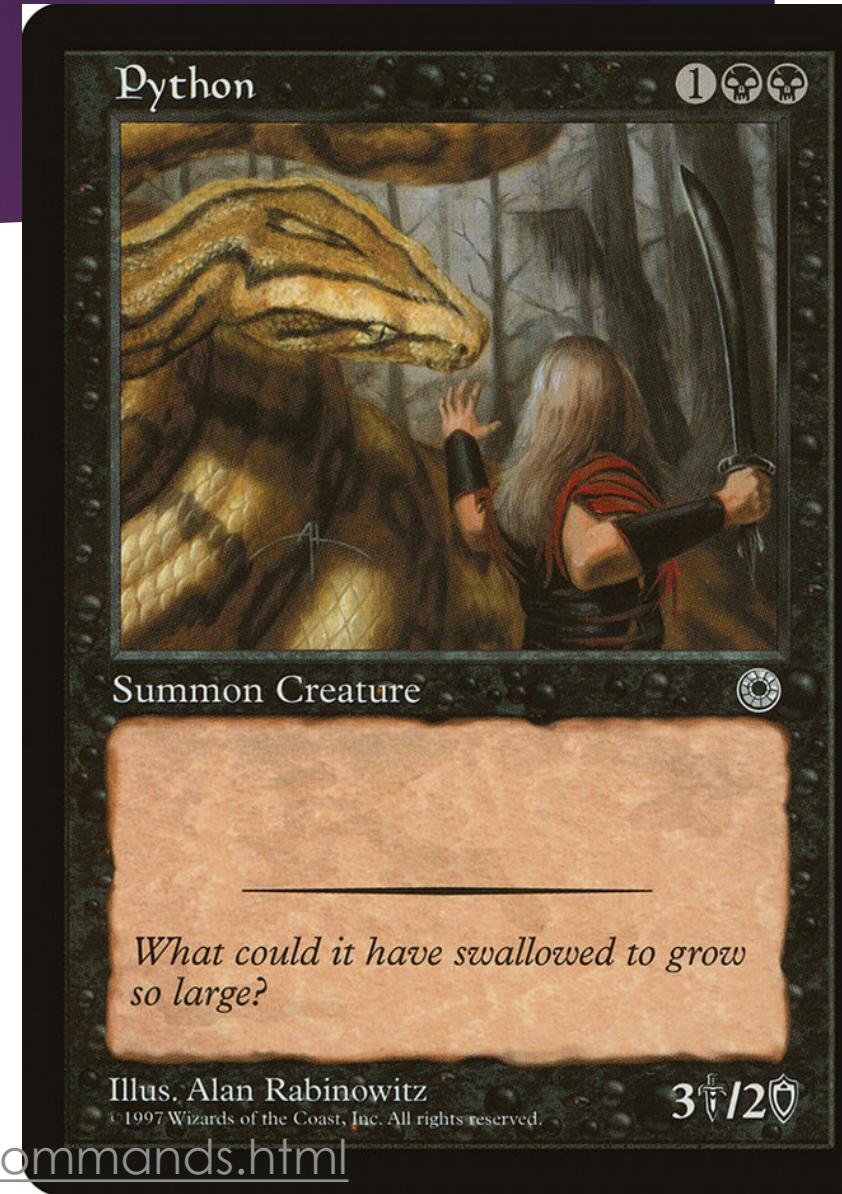
# Jupyter Magic

**MAGICS** are enhancements that IPython adds on top of the normal Python syntax.

These are known in IPython as magic commands, and are prefixed by the **% character**. These magic commands are designed to succinctly solve various common problems in standard data analysis.

Magic commands come in two flavors:

- ▶ **line magics:** Denoted by a **single %** prefix and operate on a single line of input
- ▶ **cell magics:** Denoted by a **double %%** prefix and operate on multiple lines of input.



# Jupyter Magic Magic

Like normal Python functions, IPython magic functions have docstrings, and this useful documentation can be accessed in the standard manner. So, for example, to read the documentation of the `%timeit` magic simply type this:

- ▶ `%timeit?`

Documentation for other functions can be accessed similarly. To access a general description of available magic functions, including some examples, you can type this:

- ▶ `%magic`

For a quick and simple list of all available magic functions, type this:

- ▶ `%lsmagic`

# Jupyter Demo

- ▶ See <https://github.com/magic-lantern/bios6642/tree/master/code>

# Jupyter Extensions

Must have:

- ▶ Jupytext: <https://github.com/mwouts/jupytext>
- ▶ Matplotlib: <https://github.com/matplotlib/jupyter-matplotlib>
- ▶ Offline: <https://github.com/manics/jupyter-offlinenotebook>

## DEMO

Windows: Open Anaconda Prompt (run as administrator)

macOS/Linux: Open terminal

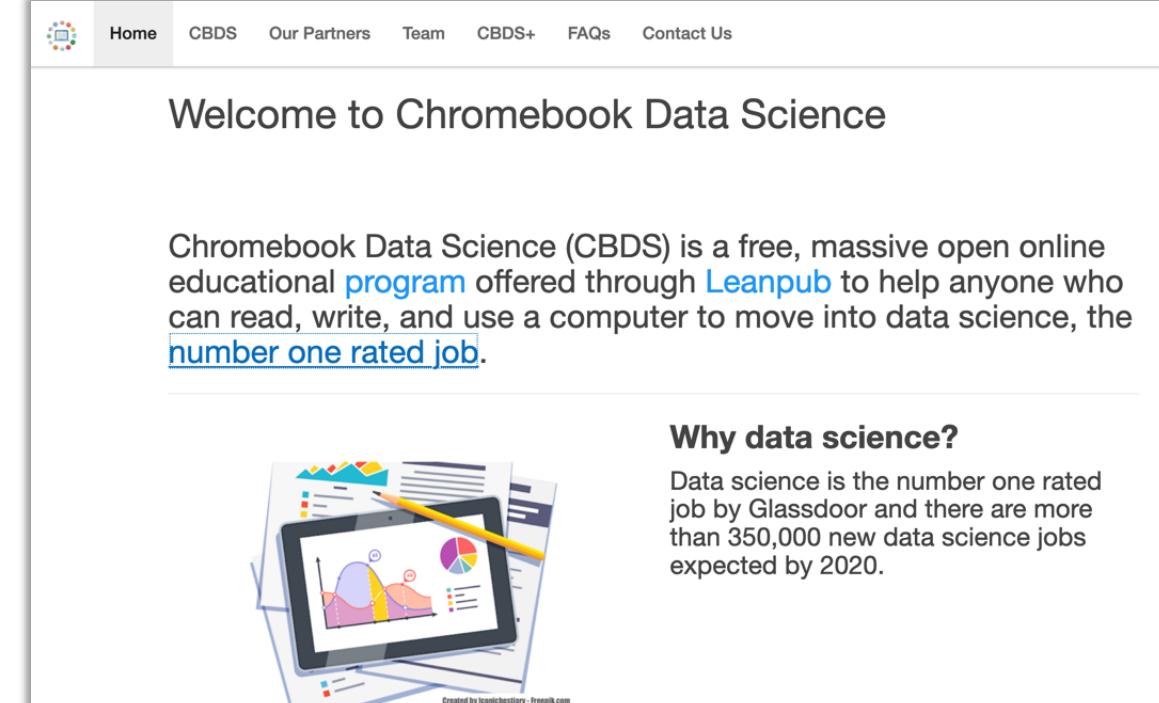
# Jupyter Extensions

Many many more:

- ▶ [https://github.com/ipython-contrib/jupyter\\_contrib\\_nbextensions](https://github.com/ipython-contrib/jupyter_contrib_nbextensions)
- ▶ <https://github.com/mauhai/awesome-jupyterlab>

# Jupyter in the cloud

- ▶ <https://www.kaggle.com/notebooks>
- ▶ <https://colab.research.google.com/notebooks/intro.ipynb>
- ▶ <https://notebooks.azure.com/>
- ▶ <https://jupyter.org/try> (Supports R, Python, Julia, Ruby, and more on mybinder.org)

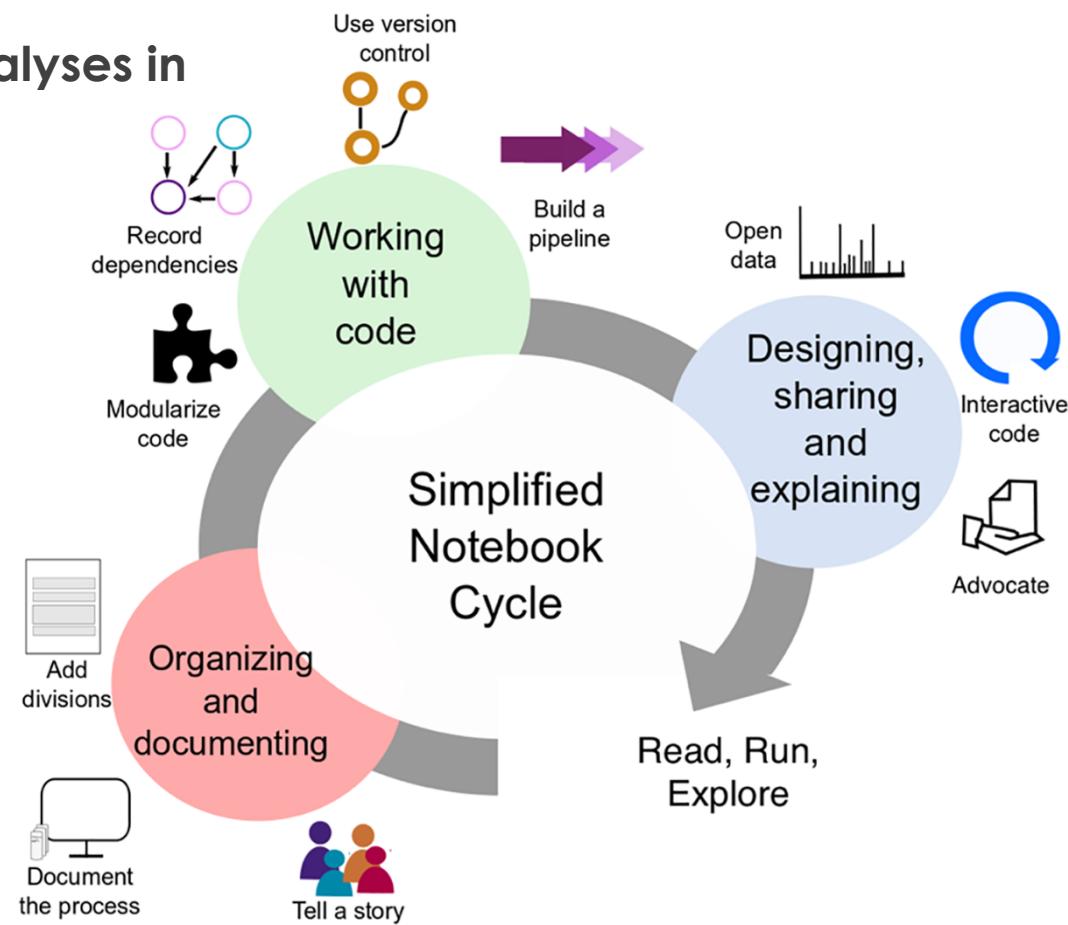


The screenshot shows the homepage of the Chromebook Data Science (CBDS) website. At the top, there is a navigation bar with links for Home, CBDS, Our Partners, Team, CBDS+, FAQs, and Contact Us. The main heading is "Welcome to Chromebook Data Science". Below the heading, a paragraph of text explains what CBDS is: "Chromebook Data Science (CBDS) is a free, massive open online educational [program](#) offered through [Leanpub](#) to help anyone who can read, write, and use a computer to move into data science, the [number one rated job](#)." To the right of this text, there is a section titled "Why data science?" which contains a brief description: "Data science is the number one rated job by Glassdoor and there are more than 350,000 new data science jobs expected by 2020." At the bottom left, there is a small graphic of a tablet displaying a chart, surrounded by papers and a pencil.

# Reproducible Science with Jupyter

## ► Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks:

- ▶ Rule 1: Tell a story for an audience
- ▶ Rule 2: Document the process, not just the results
- ▶ Rule 3: Use cell divisions to make steps clear
- ▶ Rule 4: Modularize code
- ▶ Rule 5: Record dependencies
- ▶ Rule 6: Use version control
- ▶ ...



Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks

<https://doi.org/10.1371/journal.pcbi.1007007>

# Not Jupyter Notebooks!

Joel Grus: “I Don’t Like Notebooks”

**SOLUTION: Use iPython or an IDE**

- Out of order execution
- Notebooks are difficult for beginners
- Unit testing

<https://youtu.be/7jiPeIFXb6U>

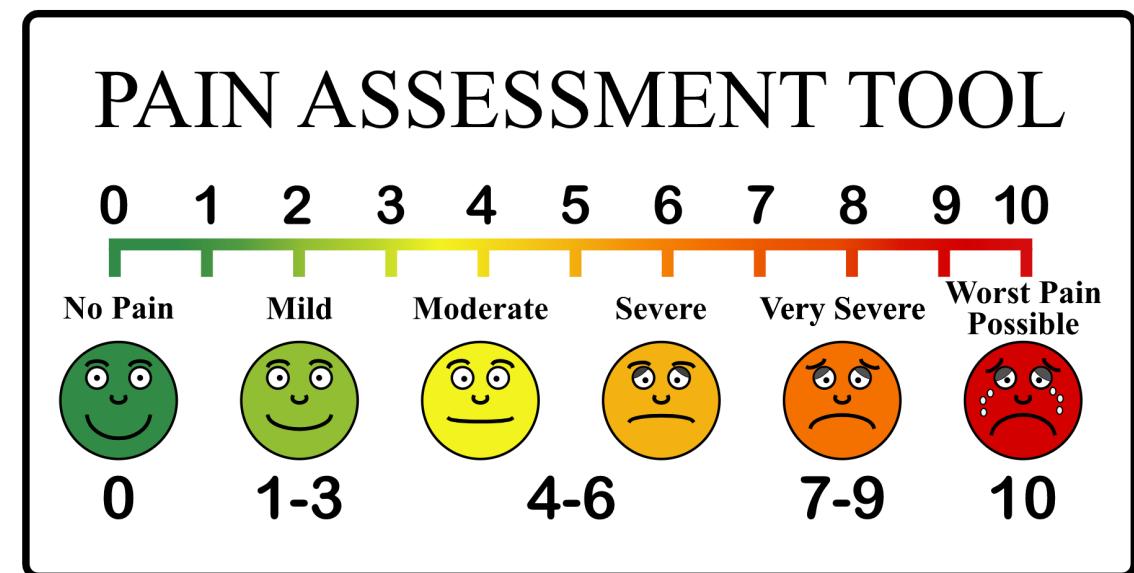
[https://docs.google.com/presentation/d/1n2RIMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit#slide=id.g362da58057\\_0\\_1](https://docs.google.com/presentation/d/1n2RIMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit#slide=id.g362da58057_0_1)

# Not Jupyter Notebooks!!

Austin Z. Henley - What's wrong with computational notebooks?

- Hard to manage code – not true IDE
- Versioning difficult
- Sharing difficult
- Can't easily be converted to a product

**SOLUTION: Fix Notebooks, add tools to solve pain points**



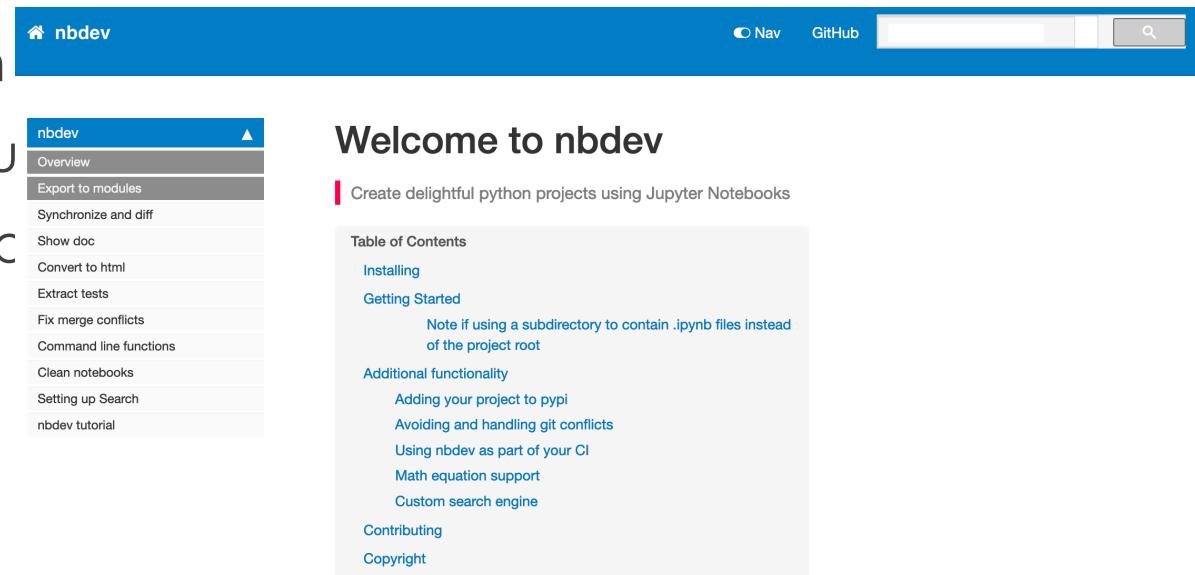
<http://web.eecs.utk.edu/~azh/blog/notebookpainpoints.html>

# Not Jupyter Notebooks!!!

Jeremy Howard - So what's **missing** in Jupyter Notebook?

- Create modular reusable code, which
- Creating hyperlinked searchable docu
- Test code (including automatically thrc
- Navigate code
- Handle version control

<https://www.fast.ai/page2/#so-whats-missing>



The screenshot shows the nbdev documentation website. The header includes a logo, navigation links (Nav, GitHub), and a search bar. The main content area has a sidebar on the left with a navigation menu:

- nbdev
- Overview
- Export to modules
- Synchronize and diff
- Show doc
- Convert to html
- Extract tests
- Fix merge conflicts
- Command line functions
- Clean notebooks
- Setting up Search
- nbdev tutorial

The main content area features a title "Welcome to nbdev" and a sub-section "Create delightful python projects using Jupyter Notebooks". It includes a "Table of Contents" with links to "Installing", "Getting Started", "Additional functionality", "Contributing", and "Copyright". A note about subdirectory usage is also present.

nbdev is a library that allows you to fully develop a library in Jupyter Notebooks, putting all your code, tests and documentation in one place. That is: you now have a true [literate programming](#) environment, as envisioned by Donald Knuth back in 1983!

Using the interactive environment, you can easily debug and refactor your code. Add `#export` flags to the cells that define the functions you want to include in your python modules. Here, for instance, is how `combined_cos` is defined and documented in the `fastai` library:

# Jupyter Data Science Demo

Give us feedback on how mybinder.org is doing and what to improve: [the mybinder.org user survey](#) (it is only three questions!). Thank you!

Starting repository: magic-lantern/bios6642/master  
Take a look at the full list of configuration files supported by [repo2docker](#).

Build logs show

Here's a non-interactive preview on nbviewer while we start a server for you. Your binder will open automatically when it is ready.

 bios6642

Name

 code

JUPYTER FAQ