JupyterHub for domain-focused integrated learning modules
Mariah Rogers  
Data Science Undergrad Advisor

Julian Kudszus  
Curriculum Development Lead
What is a “module”?

- Any course, any department, any instructor
- Anywhere from one class session to weekly
- Set of Jupyter notebooks for:
  - Lecture presentations
  - In-class labs
  - Assignments
  - Extended projects
- Run on Berkeley’s DataHub (JupyterHub for Data 8)
- Developed by students (undergrads with data science / dev skills)
tinyurl.com/xenglis-31ac
XENGLIS 31AC - Literature in American Cultures

Chinese immigrants detained at Angel Island - poetry carved on barrack walls

Individual thematic study of the poems for 3-4 weeks

One class period, computational text analysis of all 49 poems
Faculty goals for the student experience

- Must be accessible to all students
- Assume no computer science background and only high school algebra
- Students interact immediately with data by writing and running code
- Can't assume all students have personal computers
- Can't require students to figure out a local installation
- Provide a platform students can build on throughout their college careers
Facilitating data literacy in hopes of increasing computing diversity

- Remove "command line tax"
- Make computing relevant for classes that are not CS/Stats
- Have content that appeals to all demographics

Increase computing diversity

Adapted from slide made by Gunjan Baid & Vinitra Swamy
The first Berkeley JupyterHub

- Computational Models of Cognition (CogSci 131), 220 students
- Formerly taught in Matlab
- Jessica Hamrick
  - Ported class to Python
  - Devised browser computing environment with Jupyter notebooks on Rackspace
- Docker Spawner extension (written by Jess) for dynamic authenticated user image creation

Modified from a diagram by Jessica Hamrick
Berkeley's DataHub

- OAuth - CalNet login
  - Usable by anyone on campus, no account setup necessary
- Kubernetes:
  - Powerful, Scalable
  - Abstracts away most detail about hardware / cloud service
- 1 GB / user
DataHub evolutions

- Fall '18 on Microsoft Azure
- Summer '18 on Google Cloud
- Spring '18 on Microsoft Azure
- Fall '17 on Microsoft Azure
- Summer '17 on Haas cluster
- Spring '17 on Google Cloud

Service agnostic - can run on any cluster

Adapted from slide by Ryan Lovett, UC Berkeley
Hubs are popping up everywhere!

- Future goals of an integrated campus Hub management system
- Single log-in, students can select which course to view

“Our ultimate goal is a self-service portal for instructors to launch their own course-specific hub. They should be able to define the software stack and resource requirements with no intervention from an administrator. This kind of automation requires that we integrate a container building feature akin to what Binder uses.”
Tools essential to the modules workflow

- github
  - Open source notebook directories, pulled into user environment by Interact links

- nbGitPuller - Interact links
  - Formerly nbInterract; pulls files from a git repository
  - In Binder, Interact links spin up the node and install the dependencies

- nbServerProxy
  - Notebook server extension to proxy web services for visualizations not native to Jupyter interface (eg. interactive 3D brain visualization)

- nbRSessionProxy - built on top of nbServerProxy
  - RStudio interface in the browser, rather than R kernel in Jupyter interface
Interact Links - nbGitPuller

https://datahub.berkeley.edu/user-redirect/interact?account=ds-modules&repo=XENGLIS-31AC&branch=master&path=
What JupyterHub enables

- For students:
  - Ready-to-go working environment when class starts
  - No need to self-troubleshoot/debug installation, dependency issues
  - No need to find or download necessary files, scripts, or datasets

- For instructors and admins:
  - Dynamic load balancing
  - Shared compute and storage resources
  - Streamlined troubleshooting
  - Consolidated, identical working environments for every student (minimized 1-off errors)
Streamlining the experience

- Centralized control of library and dependency installations
  - Done once, then usable by everyone
- ONE (or a few) admins that can make (or break) production
- Abstractions from complications for students
  - See what they need for their class, nothing else
- Use of different kernels when needed without install/setup headache
Motivations for modules

- Reach students across wide variety of majors
- Make data science accessible
- Encourage future data science learnings
- Provide useful, domain-relevant skills
Program reach in numbers

1700+ students reached

40+ courses supported

40+ undergraduate curriculum developers

Over the past academic year (Fall 2017-Spring 2018)
Faculty
Direct the analysis, learning about DS potential within their field

Students
Interact with and see how Data Science is used in a field they are interested in

Curriculum Devs
Gain experience working in teams, with real data, and presenting findings

Learn DS techniques relevant to their field

Integrate DS with existing domain knowledge

Hands-on support in DS techniques
Supporting modules participants

**Instructors**
- pedagogy workshop
- support from curriculum devs

**Curriculum Devs**
- workshops and talks
- feedback from students, professors, other team members

**Students**
- data science office hours
- outside resources
Economics Module

- First notebook for the course
  - Course background info
  - Intro to jupyter notebooks

Economics 101B Spring 2018 Pre-Semester Exercises

Professor DeLong

Welcome to 101B! This introductory notebook will familiarize you with some of the basic strategies for data analysis that will be useful to you throughout the course. It will cover an overview of our software, an introduction to programming, and some economics.

Table of Contents

1 - [Computing Environment]
2 - [Introduction to Coding Concepts]
   1 - [Python Basics]
   2 - [Pandas]
   3 - [Visualization]
3 - [Macroeconomics]
   1 - [Chapter 1]
   2 - [Chapter 2]

Our Computing Environment, Jupyter notebooks

This webpage is called a Jupyter notebook. A notebook is a place to write programs and view their results.

Text cells

In a notebook, each rectangle containing text or code is called a cell.

Text cells (like this one) can be edited by double-clicking on them. They’re written in a simple format called Markdown to add formatting and section headings. You don’t need to learn Markdown, but you might want to.
Question 5: Plotting

Try plotting the total percent of people unemployed vs those unemployed for more than 15 weeks.

```python
total_unemployed = ...
unemp_15_weeks = ...

plt.scatter(total_unemployed, unemp_15_weeks)
plt.xlabel(...)
plt.ylabel(...)

# note: plt.show() is the equivalent of print, but for graphs
plt.show()
```

- Code skeleton provided
- Comments to explain individual lines of code
- Course material, instructions, and problem set all in one place
In [13]:
plt.plot(gdp, pe, label="PE as a function of GDP")
plt.xlabel("Real GDP (Billions of $)")
plt.ylabel("Planned Total Expenditure (Billions of $)")
plt.plot(gdp, gdp, label="Equilibrium line")
plt.legend()

Out[13]: <matplotlib.legend.Legend at 0x11ad4978>

5. What is the plot that you just made called? What is the name and significance of the intersection point of the two lines?

   **Answer**: 

6. Now, remember that when prices are sticky, planned total expenditure is a good measure of real GDP. Why is this? When prices are not sticky, why would this no longer hold true? What does Keynes have to say about the role that planned expenditure plays in potential production?

   **Answer**: 

Good job, you’re done now!
Jupyter notebooks for teaching

- Explanations of data science material, problem set instructions, and problem set all in one place
- No need for students to switch between windows
- For those new to data science, helpful to have all the info in one place
Rhetoric Module

Data Analysis: Moral Foundations Theory

Professor Amy Tick
- Key results from module
- Words grouped into categories
- Old way: by hand
- New way: using Python
Part 4: Run Analysis with Your Dictionary

One of the advantages of coding is how easy it is to repeat one method of analysis with different parameters. For instance, changing a single line of code means that all of the word counts, proportions, and graphs in the above sections can be recalculated using a different dictionary of Moral Foundations words.

To change what dictionary is loaded to the `mft_dict` variable, go to Part 1.2: Moral Foundations Dictionary and follow the instructions in the first code cell.
Building a module

- Requires collaboration between curriculum developers and course staff
  - Professor/TAs provide domain expertise
  - Data science curriculum developers provide data science knowledge and notebook building skills

- Optimizing for our goals
  - Providing detailed development guidelines
  - Reteaching similar concepts in different contexts
Our Computing Environment, Jupyter notebooks

This webpage is called a Jupyter notebook. A notebook is a place to write programs and view their results.

Text cells

In a notebook, each rectangle containing text or code is called a cell.

Text cells (like this one) can be edited by double-clicking on them. They're written in a simple format called Markdown to add formatting and section headings. You don't need to learn Markdown, but you might want to.

After you edit a text cell, click the "run cell" button at the top that looks like ▶️ to confirm any changes. (Try not to delete the instructions of the lab.)

**Understanding Check 1** This paragraph is in its own text cell. Try editing it so that this sentence is the last sentence in the paragraph, and then click the "run cell" ▶️ button. This sentence, for example, should be deleted. So should this one.

Code cells

Other cells contain code in the Python 3 language. Running a code cell will execute all of the code it contains.

To run the code in a code cell, first click on that cell to activate it. It'll be highlighted with a little green or blue rectangle. Next, either press ▶️ or hold down the shift key and press return or enter.

Try running this cell:

```
In [ ]: print("Hello, World!")
```
Guiding developers

- ‘Core resources,’ notebooks that teach a specific skill such as text parsing or data visualization
  - Built to be easily adaptable, can plug in domain specific data sets
  - Modify explanations for context
  - Work with course staff to incorporate domain-relevant questions
- Follow module notebook format guidelines
- Follow PEP8 guidelines
- Include link to feedback form
Key feedback from students

70% of students want to see modules in more courses

40% of students want to take Data 8 after a module

*10% had already taken Data 8
Future plans

- Improve student experience based on feedback
  - Difficulty level
  - Clarity of instructions
  - Feedback for instructors

- Expand program offerings
  - Bring modules to more classrooms and courses on campus
  - Encourage more students to study data science
More info on the program

- [data.berkeley.edu/education/modules](data.berkeley.edu/education/modules)
- [ds-modules.github.io/DS-Modules/](ds-modules.github.io/DS-Modules/) to view notebooks
- [data.berkeley.edu](data.berkeley.edu) for more on Data Science at UCB
Thank You

- Ryan Lovett
- Eric Van Dusen
- Ryan Edwards
- Yuvi Panda
- Cathryn Carson
- Anthony Suen