Sustainable development of scientific open source tools: A view from Jupyter

Fernando Pérez
fernando.perez@berkeley.edu
Why?
Why?

- **Ethical**: openness as fairness
- **Human/social**: openness fosters collaboration.
- **Epistemological**: proprietary science is an oxymoron.
- **Technical**: Python was cool :)

Python - The Beginning

the most important lesson I learned was about sharing

– Guido van Rossum

http://neopythonic.blogspot.com/2016/04/kings-day-speech.html

Slide credit: C. Willing
In reality, programming languages are how programmers express and communicate ideas — and the audience for those ideas is other programmers, not computers.

– Guido van Rossum

http://neopythonic.blogspot.com/2016/04/kings-day-speech.html
What?
IP[y]: IPython

jupyter
IPython: Interactive Python, 2001

A humble start: IPython 0.0.1, 259 LOC

“Just an afternoon hack”

https://gist.github.com/fperez/1579699
Team today: where *all the credit* goes

Plus ~ 1500 more Open source contributors!
The IPython/Jupyter Notebook

- Rich web client
- Text & math
- Code
- Results
- Share, reproduce.
Jupyter Protocol
web-age capture of the process of interactive computing

any mime-type output

- text
- svg, png, jpeg
- latex, pdf
- html, javascript
- interactive widgets
Jupyter Protocol is language agnostic

~100 different kernels: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels
Did you know there is a SPARQL kernel for @ProjectJupyter notebooks? I didn’t! So much fun to be had querying wikidata ... try it: mybinder.org/v2/gh/betatim/ ...

What do US presidents die of?

```cpp
#include <iostream>
#include "xtensor/xarray.hpp"
#include "xtensor/xio.hpp"

int main()
{
    xt::xarray<double> arr1 = xt::ones(3, 3);
    xt::xarray<double> arr2 = xt::ones(3, 3);
    xt::xarray<int> arr = xt::ones(3, 3);
    std::cout << xt::view(arr1, 1) + arr2;
    std::cout << arr;
    return 0;
}
```

Initialize a 2-D array and compute the sum of one of its rows and a 1-D array

Initialize a 1-D array and reshape it inplace
Classic ‘Notebook’...
JupyterLab: a grand unified theory of Jupyter

Huge Team Effort!
C. Colbert, S. Corlay, A. Darian, B. Granger, J. Grout, P. Ivanov, I. Rose, S. Silvester, C. Willing, J. Zosa-Forde…
Live Demo!
JupyterHub: multiuser support

JupyterHub is a multiuser version of the notebook designed for centralized deployments in companies, university classrooms and research labs.

- **Pluggable authentication**: Manage users and authentication with PAM, OAuth or integrate with your own directory service system. Collaborate with others through the Linux permission model.
- **Centralized deployment**: Deploy the Jupyter Notebook to all users in your organization on centralized servers on- or off-site.
- **Container friendly**: Use Docker containers to scale your deployment and isolate user processes using a growing ecosystem of prebuilt Docker containers.
- **Code meets data**: Deploy the Notebook next to your data to provide unified software management and data access within your organization.
CODING ENVIRONMENT

AUTHENTICATION

Slides credit: C. Holdgraf
What does this mean for science + education?

- Can utilize...
  - ...shared hardware/compute for running code
  - ...shared data storage for big datasets
  - ...shared environments for doing work
  - ...shared workflows, ideas, and results
CODING ENVIRONMENT

FANCY HARDWARE

AUTHENTICATION

jupyterhub
CODING ENVIRONMENT

CONTENT ON THE WEB

AUTHENTICATION

jupyterhub
NERSC

- Mission HPC center for US Dept. of Energy
  - 7000+ diverse users across science (e.g. cosmology, climate, biosciences)
- Cori – Cray XC40 (31.4 PF Peak)
  - 9668 Intel Knights Landing (KNL), 2388 Haswell nodes
- Deep learning: Data and analytics (DAS) group:
  - Tools for machine learning; optimized for scale
  - Cutting-edge methods/Collaborations/Training
- Interactive Computing at NERSC:
  - Modifications to SLURM including real-time and interactive queues with dedicated resource
  - Also other interactive features not described here: (visualization; science gateways etc.)

Slides credit: S. Farrell et al, LBNL
Jupyter architecture

Cori Compute Nodes

- ipyparallel or Dask Controller
- Engine/kernel

Cori Login Node

- Notebook Server Process
- kernel/ipyparallel client

Cori Filesystems

JupyterHub Web Server
Interactive Distributed Deep Learning

• View live updating results from all HPO trials
• Select best and worst performing runs
• Perform further analysis within notebook
• Stop
Welcome to the Jupyter at Research Facilities group!

There are a wide range of research facilities (such as X-ray light sources, astronomical observatories, genetics centers, etc.) that are users or potential users of the Jupyter ecosystem to support their research. Such facilities have some common needs in the Jupyter ecosystem:

1. Large data sets that are expensive to produce and prohibitive to transport need to be stored and analyzed for long periods of time. Often these data sets must be analyzed remotely.

2. Large computational resources (such as clusters or supercomputers) need to be scheduled and shared among facility users and accessed remotely.

3. Remote teams need to work in collaboration with facility personnel or on-site team members to collect and analyze data. This includes both interactive collaboration as the data is being collected and analysis long after the data has been collected.

4. Reproducible computational results, including narrative, data, code, and output, need to be stored and accessible remotely for potentially long periods of time.

5. Results need to be converted to formats suitable for publication and sharing with the broader community.

In this group, we welcome questions and discussions relating to the use of Jupyter at research facilities (also commonly called "user facilities") to address these and other needs.

https://groups.google.com/forum/#!forum/jupyter-research-facilities
mybinder.org: shareable reproducibility

Turn a GitHub repo into a collection of interactive notebooks

Origins

Explicit Dependencies

github.com/freeman-lab
BinderHub

ON-DEMAND ENVIRONMENTS

CONTENT ON THE WEB
Binder: reproducible, executable scholarship

from averaging ~150 people per week to averaging ~2,900 people per week

Berkeley: Yuvi Panda, Chris Holdgraf
Cal Poly: Carol Willing
Simula: Min Ragan-Kelley
Jessica Zosa-Forde, Tim Head
Reproducible Research

An article about computational science in a scientific publication is **not** the scholarship itself, it is merely **advertising** of the scholarship. The **actual scholarship** is the complete software development environment and the complete set of instructions which generated the figures.

*Buckheit and Donoho, WaveLab and Reproducible Research, 1995*
Collaborative and Reproducible Data Science

STAT 159 @ Berkeley, Fall 2017

- **Schedule:** 2 Lectures (80 min), 1 lab (2 h)
- **Prerequisites:** foundations in computation, probability and statistical modeling
- **Enrollment:** 50 undergrads, 10 grads
- **Graduate Student Instructor:** Eli Ben-Michael, Stats PhD student
- **Grading:** weekly readings, quizzes, homework and 3 projects.

Goals

❖ **What?**
  ❖ Core ideas: data access, computation, statistical analysis and publication.

❖ **Why?**
  ❖ An essential concern of modern computational research.
  ❖ **Social** and **scientific** implications of lack of reproducibility.
  ❖ Frame the problem in terms **practical, ethical and epistemological**.

❖ **How?**
  ❖ **Skills** and **habits** necessary to make a practice of reproducibility.
  ❖ An everyday practice, **not a “publication time” concept**.
Core skills

- **Version control:** Git and GitHub
- **Programming:** Python
- **Process automation:** Make
- **Data analysis:** Numpy, Pandas, Matplotlib, NLTK, Scikit-Learn, …
- **Documentation:** Sphinx
- **Software testing:** PyTest
- **Continuous Integration:** Travis
- **Reproducible containers:** Binder
Computational hygiene: a daily habit
picogit: git in 9 lines of Python

```python
from hashlib import sha1

# Our first commit
data1 = b'This is the start of my paper.'
meta1 = b'date: 1/1/17'
hash1 = sha1(data1 + meta1).hexdigest()
print('Hash:', hash1)

Hash: 3b32905baabd5ff22b3832c892078f78f5e5bd3b

# Our second commit, linked to the first
data2 = b'Some more text in my paper...'
meta2 = b'date: 1/2/1'
# Note we add the parent hash here!
hash2 = sha1(data2 + meta2 + hash1.encode()).hexdigest()
print('Hash:', hash2)

Hash: 1c12d2aad51d5fc33e5b83a03b8787dfadde92a4

And this is pretty much the essence of Git!
```
Automation and Testing: SW Carpentry

Automation and Make

Prerequisites

In this lesson we use `make` from the Unix Shell. Some previous experience with using the shell to let directories, create, copy, and move files, and run simple scripts is necessary.

Setup

In order to follow this lesson, you will need to download some files. Please follow instructions on the setup page.

Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td>Introduction</td>
</tr>
<tr>
<td>1:10</td>
<td>Module 1</td>
</tr>
<tr>
<td>1:30</td>
<td>Module 2</td>
</tr>
<tr>
<td>1:40</td>
<td>Module 3</td>
</tr>
<tr>
<td>2:00</td>
<td>Module 4</td>
</tr>
<tr>
<td>2:10</td>
<td>Module 5</td>
</tr>
<tr>
<td>2:30</td>
<td>Module 6</td>
</tr>
<tr>
<td>2:40</td>
<td>Module 7</td>
</tr>
<tr>
<td>3:00</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

Python Testing and Continuous Integration

Prerequisites

You only know what you test

In this lesson we use a Python library called `pytest`. Basic understanding of Python and its syntax is a prerequisite. Some previous experience with the shell is expected, but not mandatory.

Managing Expectations

In the same way that your scientific domain has expectations concerning experimental accuracy, it also has expectations concerning allowable computational accuracy. These considerations should play a role when you evaluate the acceptability of your own or someone else’s software.

Code without tests is legacy code!
Project 3: original analysis

- **Data:** included in repo or linked if too large.
- **Clean, tested code.**
- **Analysis notebooks and supporting code**
  - Break down your analysis into as many notebooks as is reasonable for convenient reading and execution.
- **Main narrative notebook:** summarizes and discusses results.
- **Reproducibility support:** Makefile and environment.yml
- **Good repository practices:** README.md, LICENSE, .gitignore.

- Use Victoria Stodden's *ENABLING REPRODUCIBLE RESEARCH: LICENSING SCIENTIFIC INNOVATION.*
Brief Analysis on the Marginal Effects of Studying

As students, we often wonder what effect an extra hour of studying will have on our grades. When trying to determine whether staying up an extra hour to study for that final exam is truly worth it, we are usually limited by imperfect information and our own superstitions. In this project, we attempt to estimate the "true" marginal effect of studying on students' grades. We try to model the effects first using OLS and then various instruments and 2 stage least squares. This repository is also meant to serve as an example of what a reproducible econometric analysis would look like.

Required Installations

The only installation needed to run this repo is Anaconda. Click here to learn about how to install Anaconda. Once installed, you should be good to go!

Using Binder

We've enabled Binder for this project which allows you to view jupyter notebooks in an executable environment. Feel free to click the link at the top of this README to launch the binder.

Getting Started

Download the repo onto your local machine and open your command prompt. Simply type in the following commands to run the analysis:

```
make clean
make env
source activate study
make run
```

After all your notebooks have run you should see new files in the results, fig, and data directories. Read about our approach and results in main.ipynb. All the figures from our analysis are saved in the fig directory and our regressions are saved in the results directory as dataframes. You can load these dataframes and work with them as regression instances (i.e. you can call .summary(), .params() etc). Click here for OLS documentation and here for 2SLS documentation.

Licensing

In an effort to enable reproducible, collaborative research our project is subject to the MIT License which allows you to modify and distribute the above code for both private and commercial usage. See LICENSE to learn more.

<table>
<thead>
<tr>
<th>File</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>added reproducibility aspects, split model fitting into 2 rtbs; NOTE...</td>
</tr>
<tr>
<td>fig</td>
<td>Fix typos in data_exploration.ipynb</td>
</tr>
<tr>
<td>results</td>
<td>Fix typos in model_fitting_2.ipynb</td>
</tr>
<tr>
<td>.gitignore</td>
<td>Add caches to .gitignore</td>
</tr>
<tr>
<td>.mailmap</td>
<td>adding mailmap to account for config issues</td>
</tr>
<tr>
<td>.travis.yml</td>
<td>Add pandas install to Travis</td>
</tr>
<tr>
<td>LICENSE</td>
<td>Added LICENSE</td>
</tr>
<tr>
<td>Makefile</td>
<td>added reproducibility aspects, split model fitting into 2 rtbs; NOTE...</td>
</tr>
<tr>
<td>README.md</td>
<td>added reproducibility sentence to README</td>
</tr>
<tr>
<td>data_exploration.ipynb</td>
<td>Minor add to data_exploration.ipynb</td>
</tr>
<tr>
<td>environment.yml</td>
<td>added reproducibility aspects, split model fitting into 2 rtbs; NOTE...</td>
</tr>
<tr>
<td>instructions.md</td>
<td>Add note about grades in team work</td>
</tr>
<tr>
<td>main.ipynb</td>
<td>correction to instruments justification</td>
</tr>
<tr>
<td>model_fitting_1.ipynb</td>
<td>Fix typos in model_fitting_1.ipynb</td>
</tr>
<tr>
<td>model_fitting_2.ipynb</td>
<td>Fix typos in model_fitting_2.ipynb</td>
</tr>
<tr>
<td>p3functions.py</td>
<td>Add two-way function</td>
</tr>
<tr>
<td>tests.py</td>
<td>Add two-way function</td>
</tr>
</tbody>
</table>
Analysis notebooks

Code and tests
Student feedback

Anyway, I would like to meet with you in the coming weeks to update you about the progress I’ve made in my jump into reproducibility, especially my experience with contributing to pandas and the few chapters of “The Practice of Reproducible Research” I got to read.

assistance. I was mainly interested in having you as an advisor because I’m interested in the idea of responsible research practices in this type of setting where the data cannot be shared - what do responsible research practices look like for analysis like this? How do I present the results in a way that shows all the steps taken and all the analyses run without giving too much information about the data?

Journalist who now is applying to Data Science graduate programs (just admitted to Columbia!)

Your class still exert a great influence on my current projects. I’ve been working on create detailed buyer personas since I came back to China and using the method you taught in class to develop pricing and operating algorithm with Python, establishing a price estimation model and optimizing the valuation system of Airbnb with modified AeroSolve Module.

To be honest, I was hesitating before whether I could do a good job in data analysis given that I originally majored in journalism. Thanks to your encourage, now I feel more confident and develop a clear career.
The world of science and education wants open platforms

~2.4M notebooks on GitHub in Jan 2018

https://github.com/parente/nbestimate
"To me, one of the patterns we see that makes the world go wrong is when somebody acts as if they aren’t powerful when they actually are powerful. So if you’re still reacting against whatever you used to struggle for, but actually you’re in control, then you end up creating great damage in the world. […]"

**We used to be kind of rebels**, like, if you go back to the origins of Silicon Valley culture, there were these big traditional companies like IBM that seemed to be impenetrable fortresses. And we had to create our own world. **To us, we were the underdogs and we had to struggle. And we’ve won.** I mean, we have just totally won. We run everything. […] But we don’t act like it.

We have no sense of balance or modesty or graciousness having won. **We’re still acting as if we’re in trouble and we have to defend ourselves,** which is preposterous. And so in doing that we really kind of turn into assholes, you know?"

Jaron Lanier, New York Magazine interview, April 2018.
Responsibilities? Us, but also partners!

❖ We’re not the underdogs anymore - behave like it!
  ❖ Better at community, inclusion, sustainability.

❖ Our industrial partners also have a responsibility
  ❖ Many in industry behave purely like extractive miners:
    ❖ Of technology (that is free to use)
    ❖ Of people (who are “trained for free”)
    ❖ Projects suffer, people too (who can “train for free”? not everyone….)
Complex tensions and competing incentives

- We’re successful, yet our sustainability is in question.
- Deep academic connections, yet fundamental conflict with academic incentive model.
- Sources of technology and economic value, yet no “business model”.
- Pieces of the puzzle:
  - Grad students and postdocs (with changes!)
  - Research Software Engineers (UK, Netherlands)
  - National Laboratories (US)
  - Better industry partnership arrangements?
  - Professorships with explicit software/tools/data mandate?
Inspired by Elinor Ostrom’s Governing the Commons

Cooperative, high-interaction, self-governed communities (case studies)

Essential conditions:

High mutual trust and long-term interest in the community

Note: Titus was also reading it, so we’re probably on to something :)
Thank You!