Jupyter Notebooks

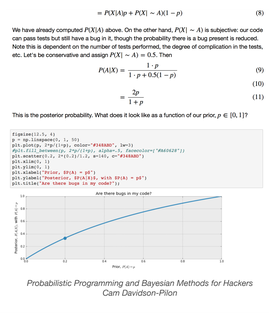
Facilitating Productivity, Sustainability, and Accessibility of Data Science

Min Ragan-Kelley, Project Jupyter Contributors*

**Simula Research Laboratory*

What is a Jupyter Notebook?

- Document format including prose, code, and rich media output
- Web application for creating notebook documents
- Interactive environment for exploratory computing
- Language agnostic with kernels in many languages, including Python, R, Julia, Haskell, JavaScript, C++, and more
- Convertible to other formats, including HTML, LaTeX, PDF, Markdown, etc.



Probabilistic Programming and Bayesian Methods for Hackers
Cam Davidson-Pilon

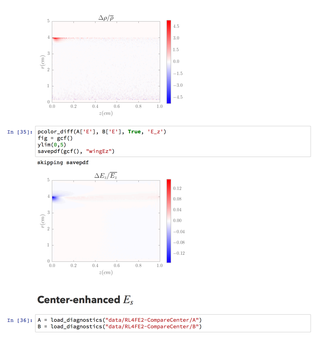
What is IPython?

- Tools for improved interactivity in the Python language
- Introspection tools
- Tab completion
- Tooltips
- Improved tracebacks

```
np.arccos        np.arctanh       np.around        np.array_split
np.arccosh       np.argmax        np.array         np.array_str
np.arcsin        np.argmin        np.array2string
np.arcsinh       np.argpartition  np.array_equal
np.arctan        np.argsort       np.array_equiv

In [2]: np.arr
np.array         np.array_equal   np.array_repr    np.array_str
np.array2string  np.array_equiv   np.array_split

In [3]: np.array?

In [3]: get_ipython??
Type:       method
String form: <bound method TerminalInteractiveShell.get_ipython of <IPython.t
inal.interactiveshell.TerminalInteractiveShell object at 0x101a4a7f8>>
File:       /Users/minrk/dev/ip/mine/IPython/core/interactiveshell.py
Definition: get_ipython()
Source:
    def get_ipython(self):
        """Return the currently running IPython instance."""
        return self

In [4]:
```

Productivity

Interactive tools such as IPython and Jupyter improve productivity by enabling rapid, iterative prototyping and exploration.



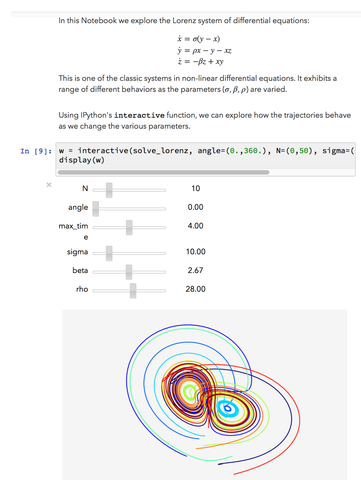
Notebooks, being both an interactive environment and a communication medium, facilitate collaboration by keeping the researcher in one environment for both performing computational explorations and communicating the current status with colleagues.

By including code, prose, and rich media outputs, notebooks enable collaboration with technical and non-technical colleagues alike, reducing the need to work separately in coding environments and presentation software or static documents.

Life-cycle of a Computational Idea

Jupyter provides tools for the life-cycle of a computational idea, from interactive exploration to collaboration to testing to communication, publication, and reproducibility.

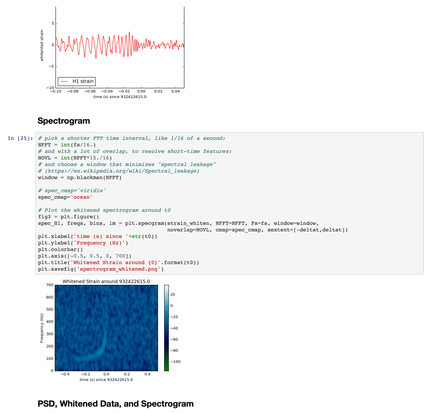
1. Explore ideas interactively in a Jupyter environment, such as IPython or a notebook.



2. Build a library or simulation based on what has been learned.
3. Describe results in an academic publication or blog post, accompanied by demonstration notebook(s).

M. G. Bobra and S. Ilonidis 2016 **ApJ** *821 127*

4. Publish companion notebooks via binder or another service, enabling readers to interact with your work and quickly reproduce results and create derivative work.

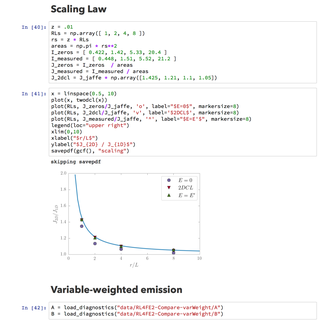


LIGO Gravitational Wave Observation
https://losc.ligo.org/tutorials/

Sustainability

notebooks

Jupyter notebooks are computational documents containing code, prose, and the outputs of computations. By performing analyses or figure generation in a notebook, the process of producing the output is automatically captured in a format that can be re-executed, making it more likely that results can be reproduced or modified later, either by the authors or by reviewers and readers.



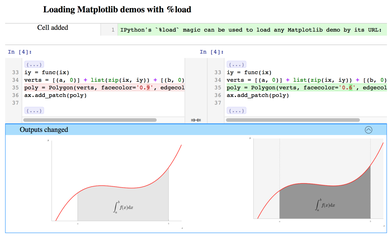
nbval

A key part of sustainability of any software is testing. The nbval package enables automatic testing and validation of notebooks by integrating with the pytest testing framework.

```
Plotting in the Notebook.ipynb ..FF.F

==================================== FAILURES ====================================
_____________________________________ cell 9 _____________________________________
Notebook cell execution failed
Cell 9: Cell outputs differ

Input:
x = np.linspace(0, 3*np.pi, 500)
plt.plot(x, np.sin(x**2))
plt.title('A simple chirp');

Traceback: mismatch 'text/plain'
<<<<<<<<<<<< Reference output from ipynb file:
<matplotlib.figure.Figure at 0x106622080>
============ disagrees with newly computed (test) output:
<matplotlib.figure.Figure at 0x10c8e01d0>
>>>>>>>>>>>>
```

nbdime

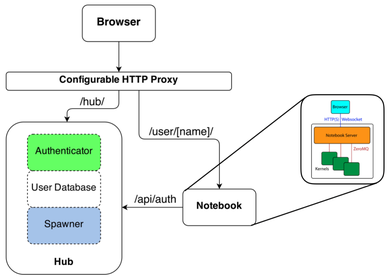
Diffing and merging content is an important part of collaboration, in both software and communication. nbdime enables diffing and merging notebooks, making intelligent decisions based on the structure of notebook documents.



Accessibility

There is a growing wealth of software tools for data science, and a growing diversity of students and researchers who need data science tools but may not have the programming or Unix skills expected by traditional tools.

Jupyter's web-based tools allow local deployment as a desktop application, or hosted deployment on shared computational resources with **JupyterHub**—including in the cloud or on traditional HPC systems—managed by instructors or system administrators.



Jupyter deployments enable students to run computations on large scale computational resources and learn computational concepts without the need to install any software on their own computers, or to resort to terminal-based environments.

Jupyter Ecosystem

- **JupyterHub**: multi-user server for exposing hosted compute resources to students and researchers, such as cloud services or HPC
- **nbviewer**: Web service for rendering notebooks to HTML for easy sharing on the web
- **binder**: Web service for hosting demonstration and example notebooks based on GitHub repositories for zero-install demonstrations.
- **nteract**: Desktop application for running local notebooks
- **JupyterLab**: next generation web application and toolkit for building Jupyter-based applications
- **nbval**: pytest plugin for testing and validating notebooks
- **nbdime**: toolkit with git integration for diffing and merging notebooks
- **Kernel Gateway**: Server for providing hosted kernels, which can be used by non-notebook applications such as dashboards

Support

Alfred P. Sloan Foundation
Gordon and Betty Moore Foundation
The Leona M. and Harry B. Helmsley Charitable Trust
Open DreamKit
Horizon 2020 European Union funding for Research & Innovation