Benefits and Limitations of Jupyter-based Web Applications

Nicole Brewer, Rajesh Kalyanam, Rob Campbell,
Carol Song, Lan Zhao

~~Benefits and Limitations of Jupyter-based Web Applications~~

"The Story of the Little Notebook"

Nicole Brewer, Rajesh Kalyanam, Rob Campbell,
Carol Song, Lan Zhao

Nicole Brewer
Graduate Research Assistant
PhD Student, History and Philosophy of Science
Arizona State University

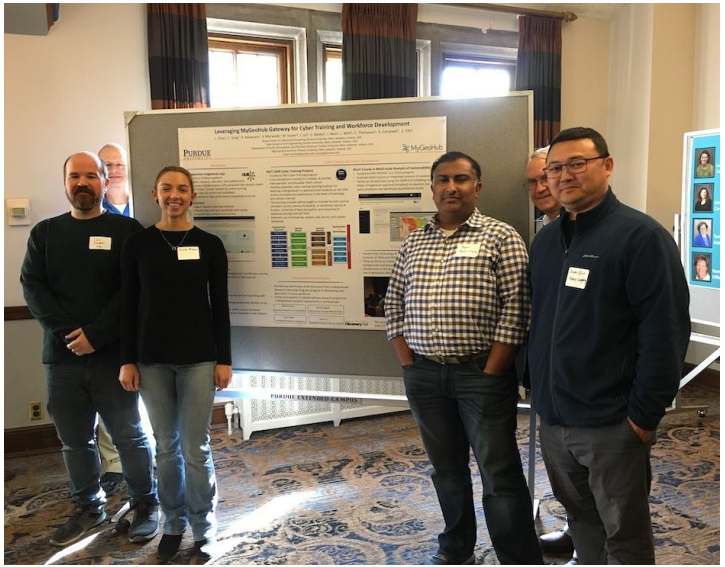
@catch_me_coding

Nicole Brewer

Research Software Engineer

Scientific Solution Group

Purdue University

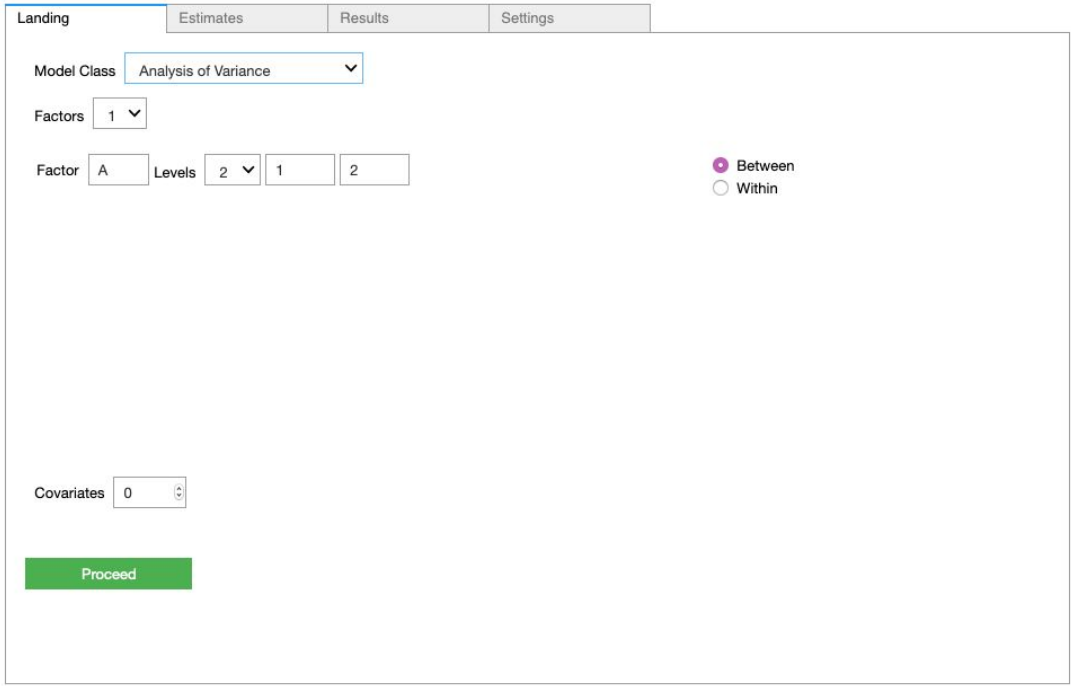


Overview

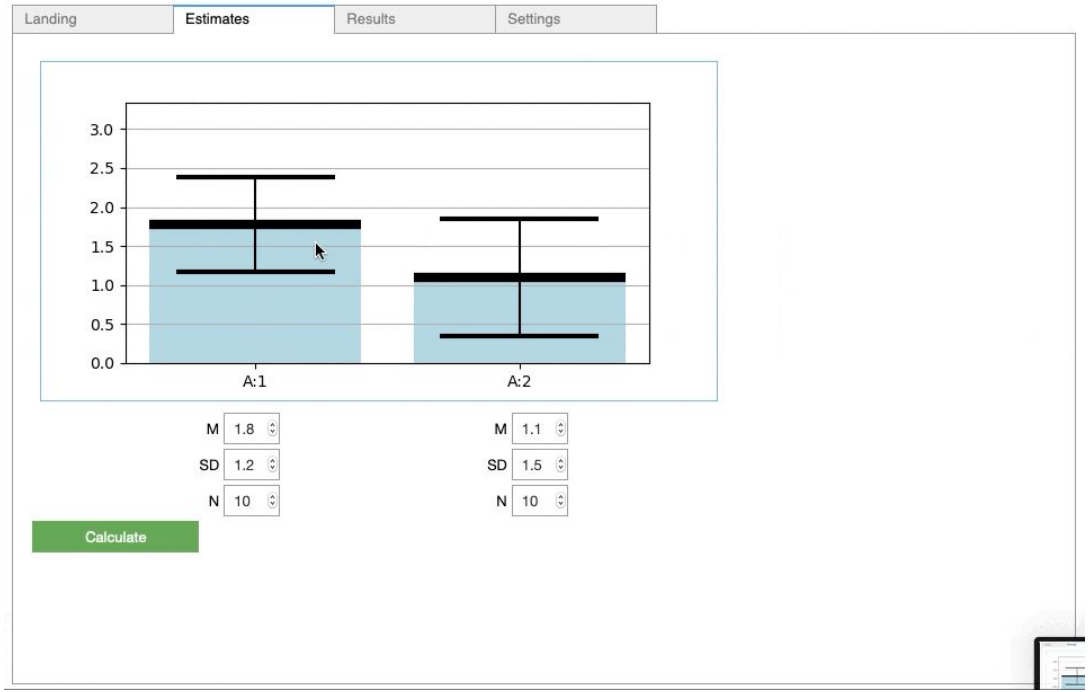
1. What is a scientific web application?
2. Why should researchers develop them?
3. How should researchers and RSEs collaborate?
4. When should RSEs build apps with Jupyter Notebooks?
    a. Benefits
    b. Deployments and supported features
    c. Limitations
    d.

What is a scientific web application?

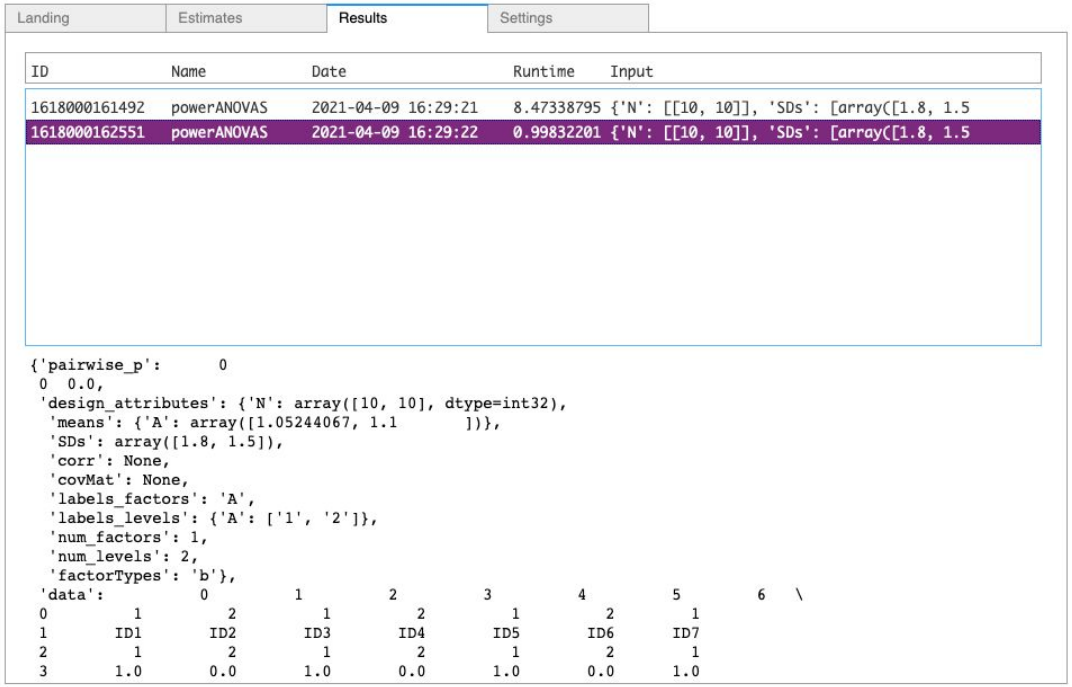
A scientific application is a graphic interface that **allows users to manage and analyze data and perform computations**



A scientific application is a graphic interface that **allows users to manage and analyze data and perform computations**



A scientific application is a graphic interface that **allows users to manage and analyze data and perform computations**



Why should researchers develop web apps?

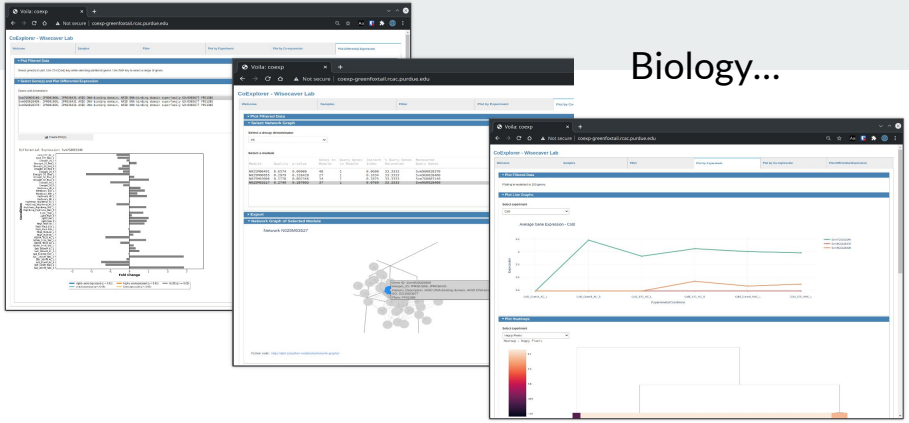
Web applications are increasingly popular scientific publication tools

- Internal tool to improve research process
- Teaching
- Outreach
- Stakeholder engagement
- Updating policy makers on the evolving state of the COVID-19 pandemic

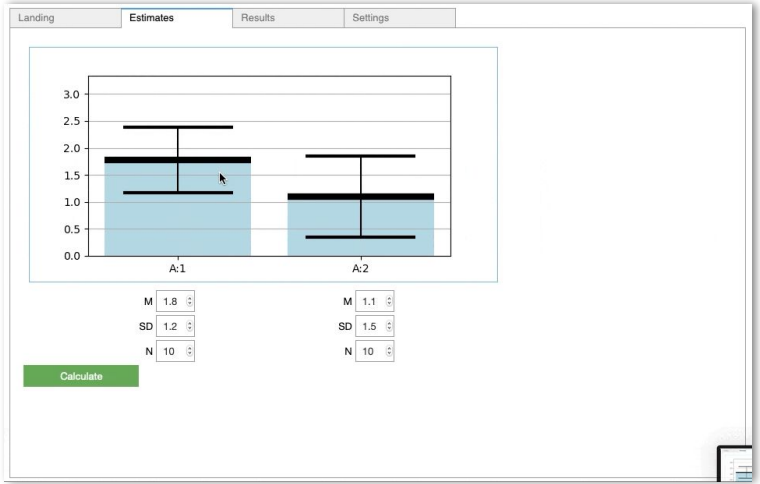
Web applications improve the quality of research data and software

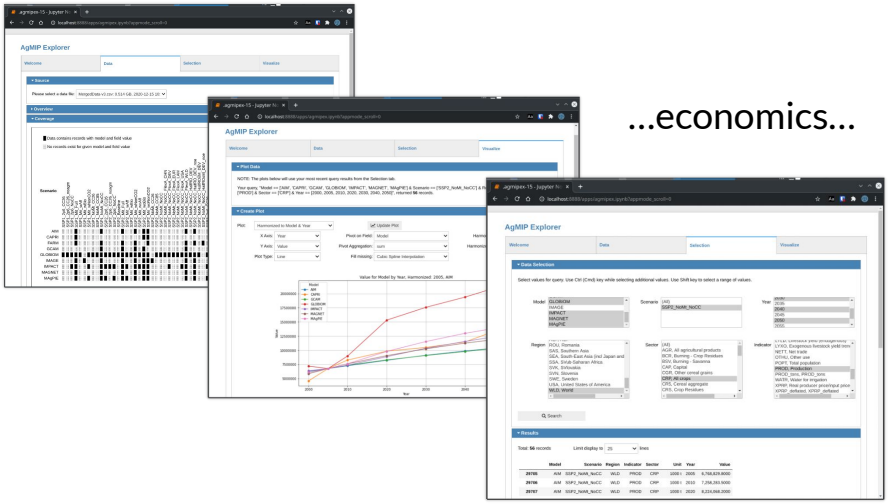
- **Accessibility** of cyberinfrastructure for non-technical users
- **Reusability** of data and software
- **Reproducibility** may be improved depending on the app design

How should RSEs and scientists collaborate?



Biology…





…economics…

…psychology,

aeronautics,

and more.

When should RSEs use Jupyter to develop web apps?

When should RSEs use a Jupyter Notebook?

Benefits	Supported Features	Limitations
1. scientists may have an existing notebook 2. easy for researchers to maintain 3. third-party widget ecosystem	• Shared Storage • User Authenication • Group Managmenet • Repeating Components	• Little stylistic control • Third-party widget ecosystem • Modularity requires OOP expertise • Many data visualization libraries aren't interactive

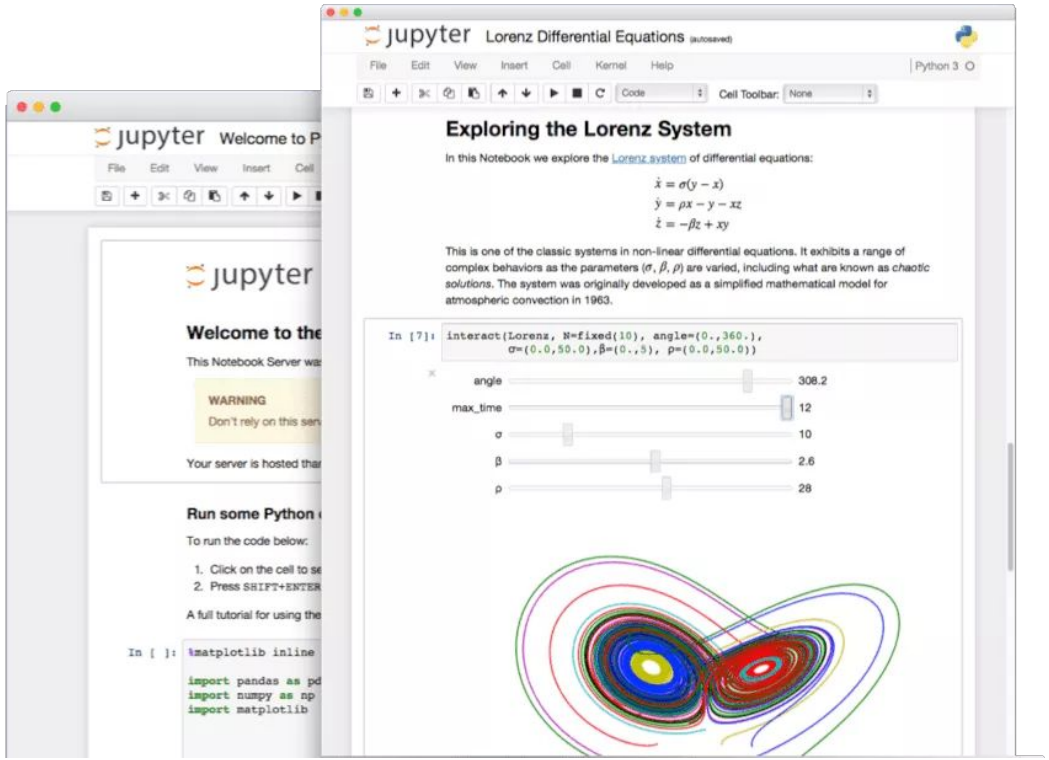
Benefits

1. Scientists may already have an exploratory Jupyter Notebook
2. Jupyter Notebooks are easy for researchers to modify and maintain
3. Well-maintained widget ecosystem for visualization

1 Scientists may have an exploratory notebook

Jupyter is a web application for creating and sharing computational documents

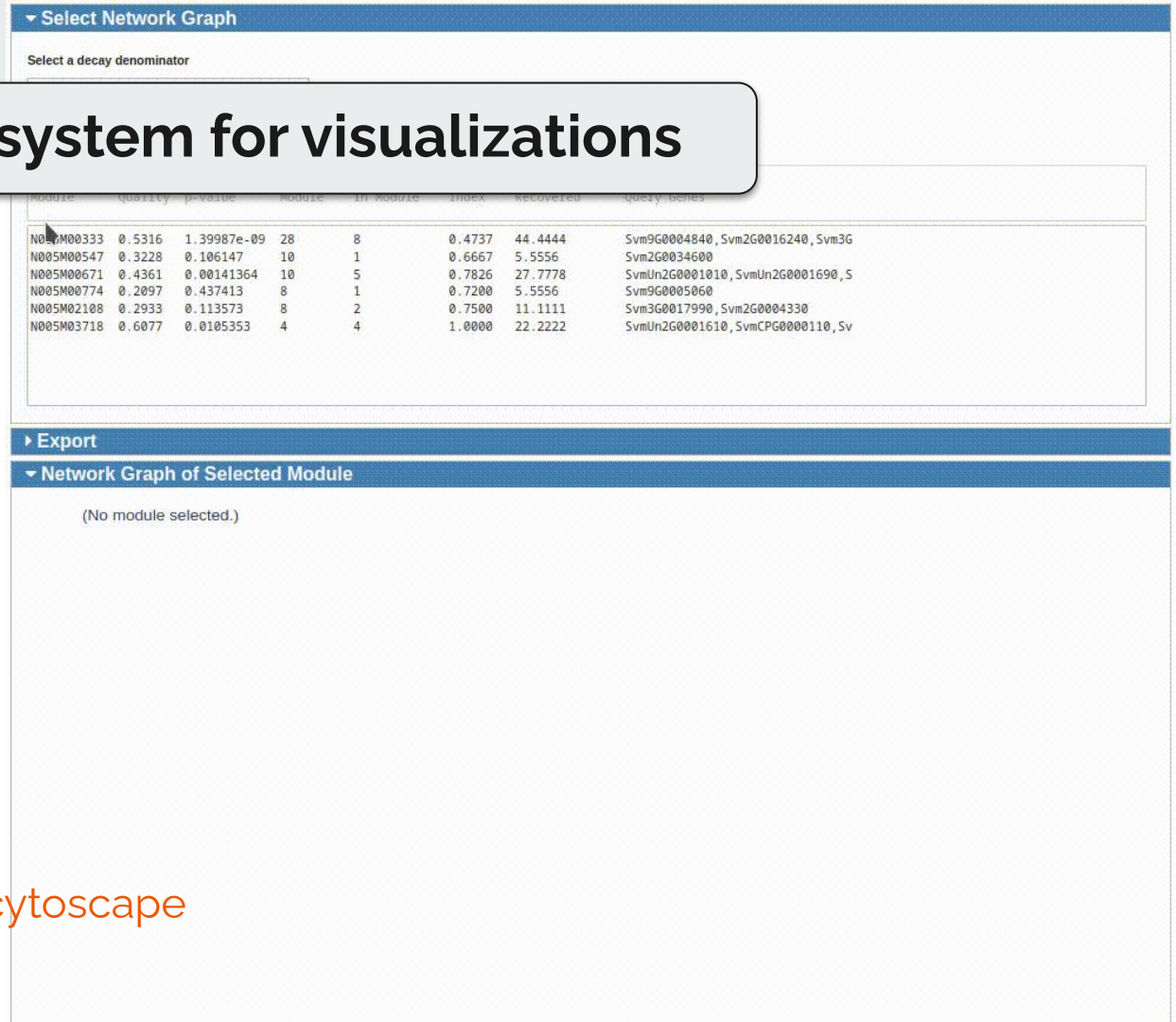
- text
- code
- visual outputs



2 Jupyter Notebooks are easy to modify & maintain

- Easy Python; no traditional web development (HTML, CSS, etc.)
- Code is centrally located - no multifile framework
- Traditional debugging available but not required

3 ipywidget ecosystem for visualizations



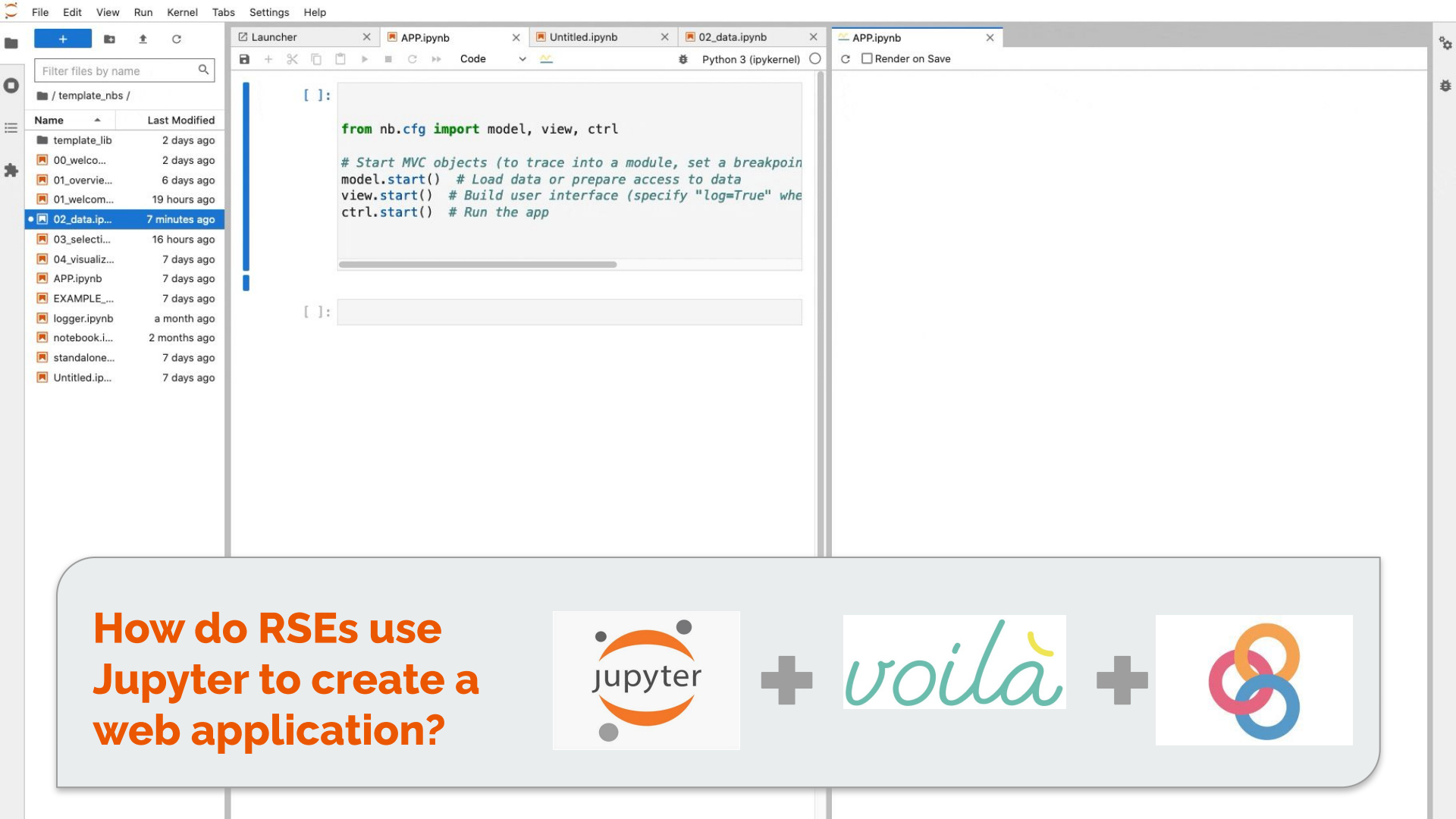
visualize networks with ipycytoscape

3 ipywidget ecosystem for visualizations

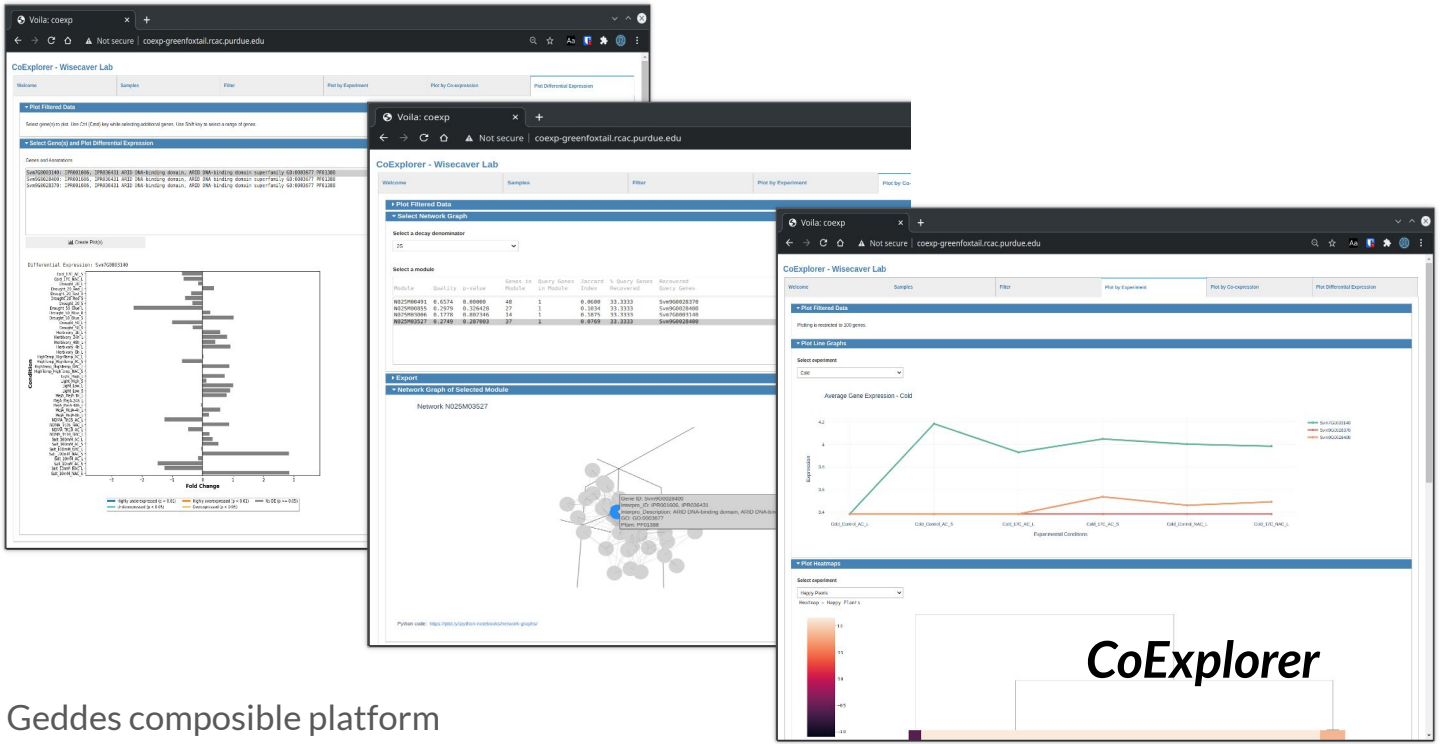
Interactive maps with ipyleaflet

Deployments and supported features

1. Shared Storage
2. User Authenication & Group Managmenet
3. Repeating Components


File Edit View Run Kernel Tabs Settings Help
Launcher
APP.ipynb
Untitled.ipynb
02_data.ipynb
Code
Python 3 (ipykernel)
Filter files by name
/ template_nbs /
Name
Last Modified
template_lib 2 days ago
00_welco... 2 days ago
01_overvie... 6 days ago
01_welcom... 19 hours ago
02_data.ip... 7 minutes ago
03_selecti... 16 hours ago
04_visualiz... 7 days ago
APP.ipynb 7 days ago
EXAMPLE_... 7 days ago
logger.ipynb a month ago
notebook.i... 2 months ago
standalone... 7 days ago
Untitled.ip... 7 days ago
[ ]:
from nb.cfg import model, view, ctrl
# Start MVC objects (to trace into a module, set a breakpoin
model.start() # Load data or prepare access to data
view.start() # Build user interface (specify "log=True" whe
ctrl.start() # Run the app
[ ]:
APP.ipynb
Render on Save
How do RSEs use Jupyter to create a web application?
jupyter
voilà

1 Shared Storage



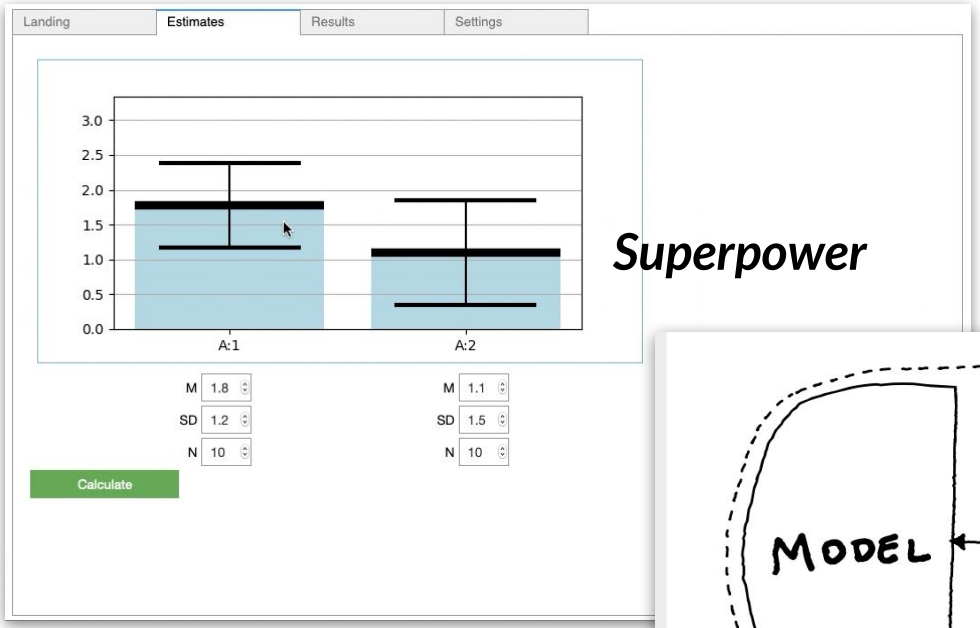
- Hosted on Purdue's Geddes composible platform
- Kubernetes to scalably deploy sessions of the application

2 Authentication and Group Management

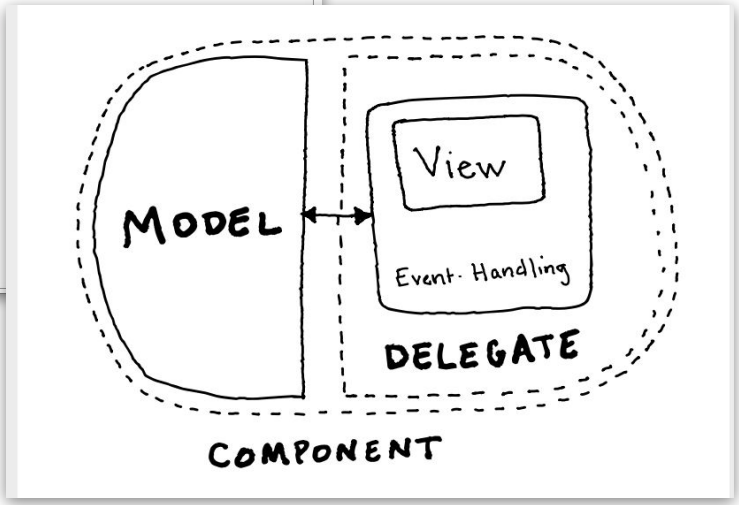
AgMIP Explorer

- Shared data storage among select users only
- Application hosted as a tool on MyGeoHub (science gateway built with HubZero)

3 Modularity and Interactive Plots



Superpower



- Repeating components
- Modified ModelDelegate design pattern for increased modularity
- Interactive plots with bidirectional communication

Limitations

- Limited stylistic control
- Widget ecosystem is largely built by third party developers
- Modularity requires OOP expertise
- Many data visualization libraries aren’t interactive and may require custom JavaScript or other “hacky” solutions

Review

Benefits	Supported Features	Limitations
1. scientists may have an existing notebook 2. easy for researchers to maintain 3. third-party widget ecosystem	• Shared Storage • User Authenication • Group Managmenet • Repeating Components	• Little stylistic control • Third-party widget ecosystem • Modularity requires OOP expertise • Many data visualization libraries aren't interactive

Future work

- User groups not managed by science gateways
- Scalability
- Access to HPC resources
- Refactoring

nicole-brewer/nbdev_app_template

- Dockerfile for deploying on a composable platform
- Docker-compose for development environment
- Extended version of the nbdev template with instructional notebooks and templates

Give the repo a  and watch for updates