

# Advancing Open and Reproducible Water Data Science by Integrating Data Analytics with an Online Data Repository

Jeffery Horsburgh<sup>1</sup>, Scott Black<sup>2</sup>, and Anthony Castronova<sup>3</sup>

<sup>1</sup>Utah State University

<sup>2</sup>Consortium of Universities for the Advancement of Hydrologic Sciences, Inc.

<sup>3</sup>Consortium of Universities for the Advancement of Hydrologic Sciences, Inc

November 21, 2022

## Abstract

Scientific and related management challenges in the water domain require synthesis of data from multiple domains. Many data analysis tasks are difficult because datasets are large and complex; standard formats for data types are not always agreed upon nor mapped to an efficient structure for analysis; water scientists may lack training in methods needed to efficiently tackle large and complex datasets; and available tools can make it difficult to share, collaborate around, and reproduce scientific work. Overcoming these barriers to accessing, organizing, and preparing datasets for analyses will be an enabler for transforming scientific inquiries. Building on the HydroShare repository's established cyberinfrastructure, we have advanced two packages for the Python language that make data loading, organization, and curation for analysis easier, reducing time spent in choosing appropriate data structures and writing code to ingest data. These packages enable automated retrieval of data from HydroShare and the USGS's National Water Information System (NWIS), loading of data into performant structures keyed to specific scientific data types and that integrate with existing visualization, analysis, and data science capabilities available in Python, and then writing analysis results back to HydroShare for sharing and eventual publication. These capabilities reduce the technical burden for scientists associated with creating a computational environment for executing analyses by installing and maintaining the packages within CUAHSI's HydroShare-linked JupyterHub server. HydroShare users can leverage these tools to build, share, and publish more reproducible scientific workflows. The HydroShare Python Client and USGS NWIS Data Retrieval packages can be installed within a Python environment on any computer running Microsoft Windows, Apple MacOS, or Linux from the Python Package Index using the PIP utility. They can also be used online via the CUAHSI JupyterHub server (<https://jupyterhub.cuahsi.org/>) or other Python notebook environments like Google Collaboratory (<https://colab.research.google.com/>). Source code, documentation, and examples for the software are freely available in GitHub at <https://github.com/hydroshare/hsclient/> and <https://github.com/USGS-python/dataretrieval>.

# Advancing Open and Reproducible Water Data Science by Integrating Data Analytics with an Online Data Repository

**Jeffery S. Horsburgh**

Utah State University

**Scott Black, Anthony Castronova**

Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)



Utah Water Research Laboratory  
UtahStateUniversity





# Reproducibility is key

*“If I have seen further it is by standing on the shoulders of Giants.”*

Isaac Newton, 1625

Building trust in scientific research requires transparency  
and reproducibility



# Collaborative (Reproducible) Data Science Workflow

- Easily create a digital instance of a dataset or data science workflow
- Quickly share it with colleagues (perhaps privately at first)
- Add value through collaboration, annotation, and iteration
- Describe with metadata
- Eventually...share publicly or formally Publish so others can reuse

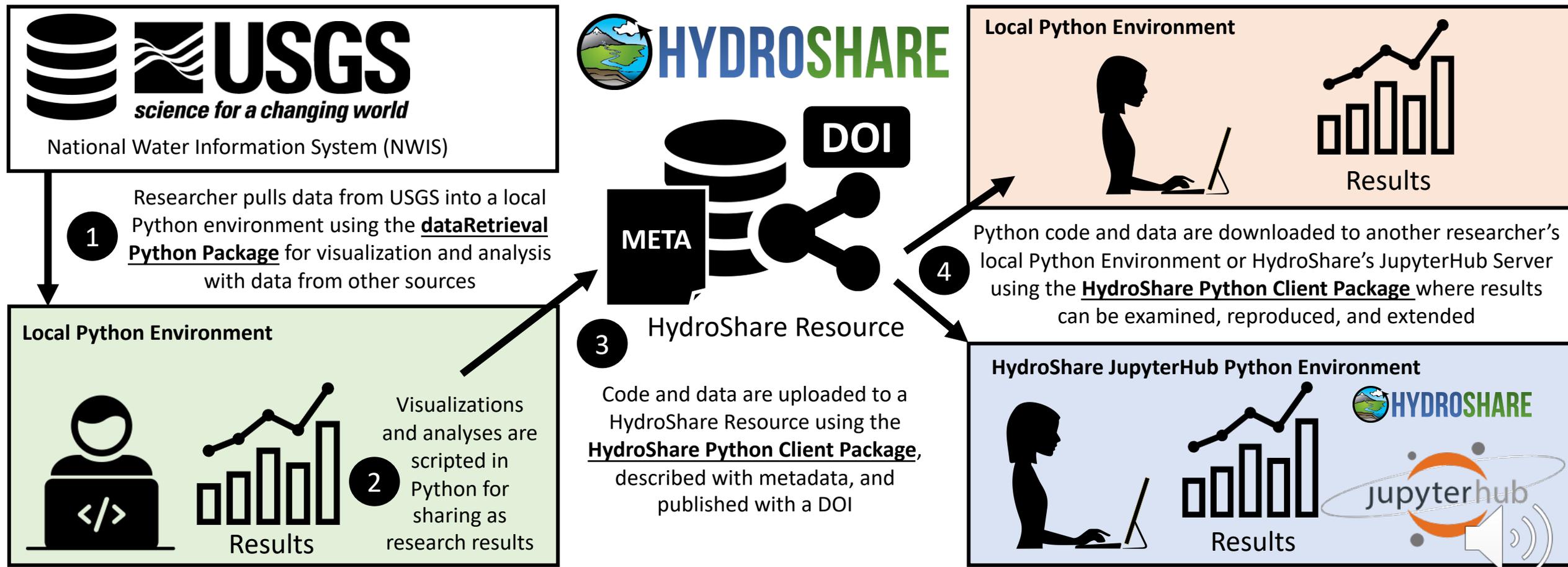


What is the role of data repositories in this scientific workflow?



# Connecting Visualization and Analysis with an Online Repository

- Better enabling collaborative data science workflows and reproducibility



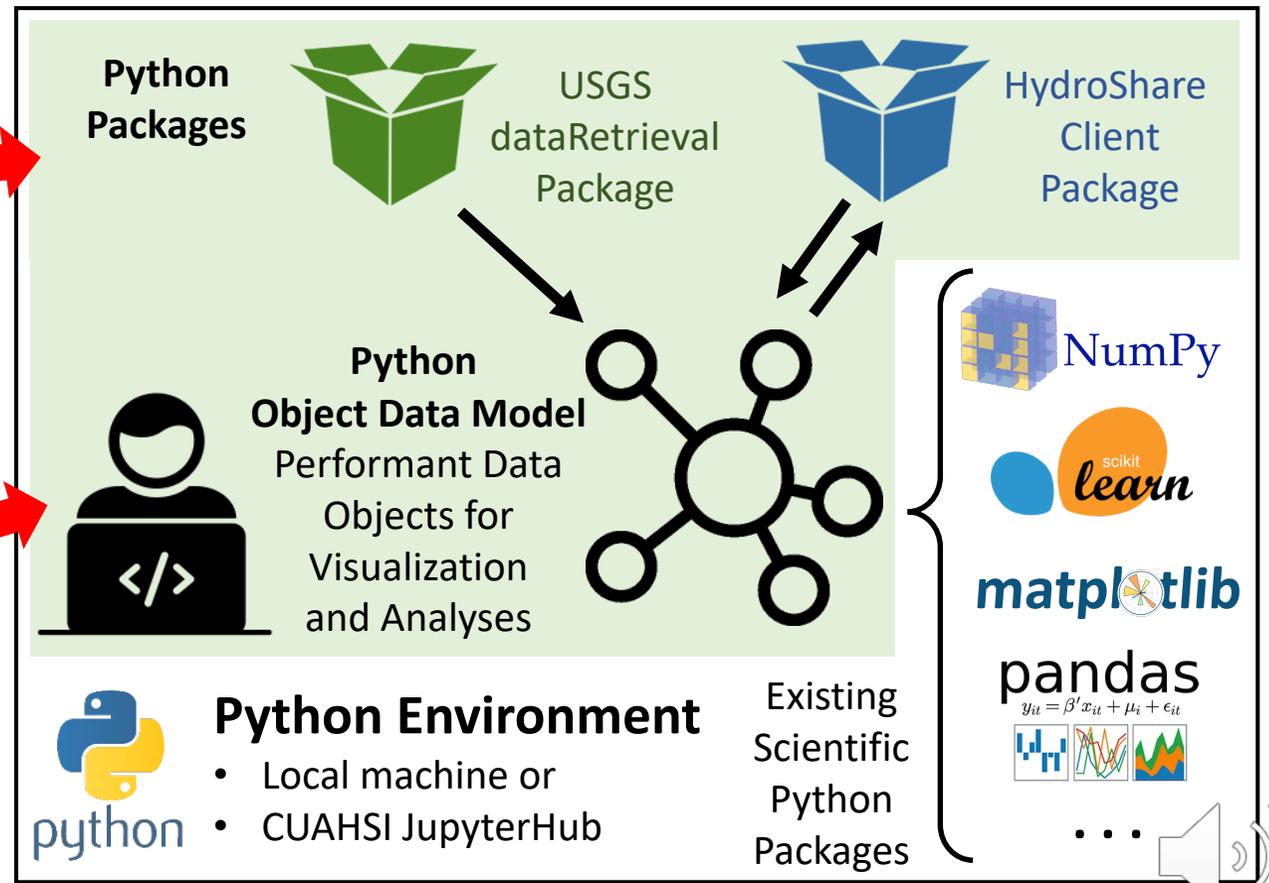
# The tools needed to make this work



Data repositories

Tools for accessing and interacting with those repositories

A Python representation of the data retrieved that can be operated on using existing data science tools





# HYDROSHARE

<http://www.hydroshare.org>

- A repository for sharing and publication that uses FAIR principles
- Operated by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)
- Creating and sharing data and models using a variety of file formats and flexible metadata
- **Public-facing REST API and Python client enabling automated interactions**

The image shows two overlapping browser windows. The top window displays the HydroShare homepage with a navigation menu (MY RESOURCES, DISCOVER, COLLABORATE, APPS, HELP, ABOUT) and a large banner image of a landscape with a rainbow. The text 'Discover' is overlaid on the banner, with a subtext: 'Discover content shared by your colleagues and other users. Access a broad range of resource types used in hydrology.'

The bottom window shows a resource page titled 'Water Temperature in the Little Bear River at Mendon Road near Mendon, UT'. The page includes the following information:

- Authors:** Jeff Horsburgh, Amber Jones
- Owners:** Jeff Horsburgh
- Resource type:** Time Series
- Created:** June 6, 2015, 3:57 a.m.
- Last updated:** June 6, 2015, 4:25 a.m. by Jeff Horsburgh

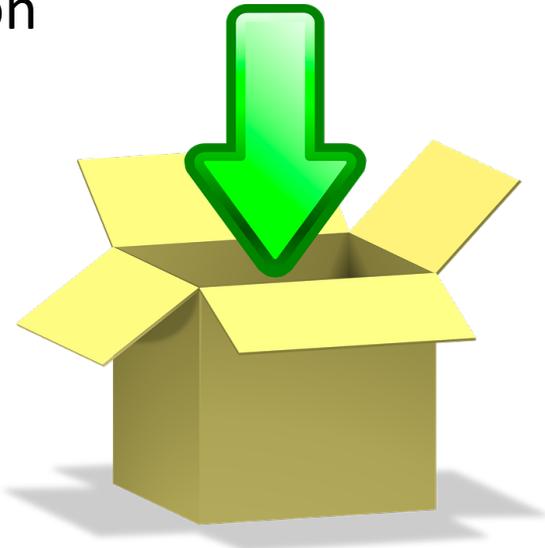
The page also features an 'Abstract' section, a 'Subject' section with tags for 'Temperature', 'Water', 'Water quality', 'Little Bear River', and 'Utah', and a 'How to cite' section with the citation: 'Horsburgh, J., A Jones (2015). Water Temperature in the Little Bear River at Mendon Road near Mendon, UT, HydroShare, <http://www.hydroshare.org/resource/1a25b11fa1354773b66c>

At the bottom, there is a 'Sharing' section with a 'Public' status and a Creative Commons Attribution CC BY license icon. A 'Manage access' button is also visible.

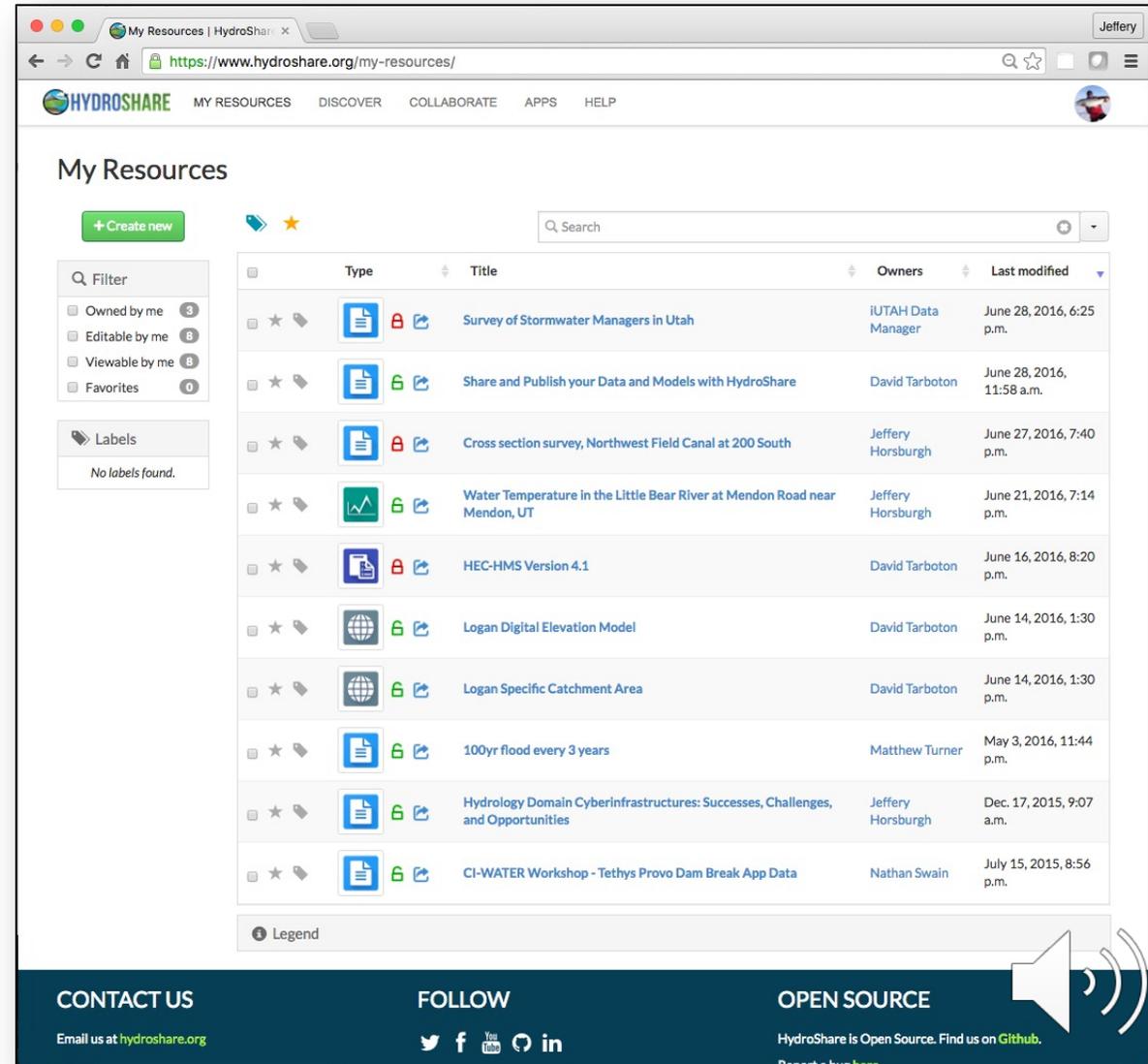
# HydroShare “Resources”

- **Resource** = primary unit of digital content
  - Create, version, copy
  - Describe
  - Own, share, access
  - Discover
  - Formal Publication

A “Resource” is a container into which users can put digital content



Resources can be datasets, models, or other research products



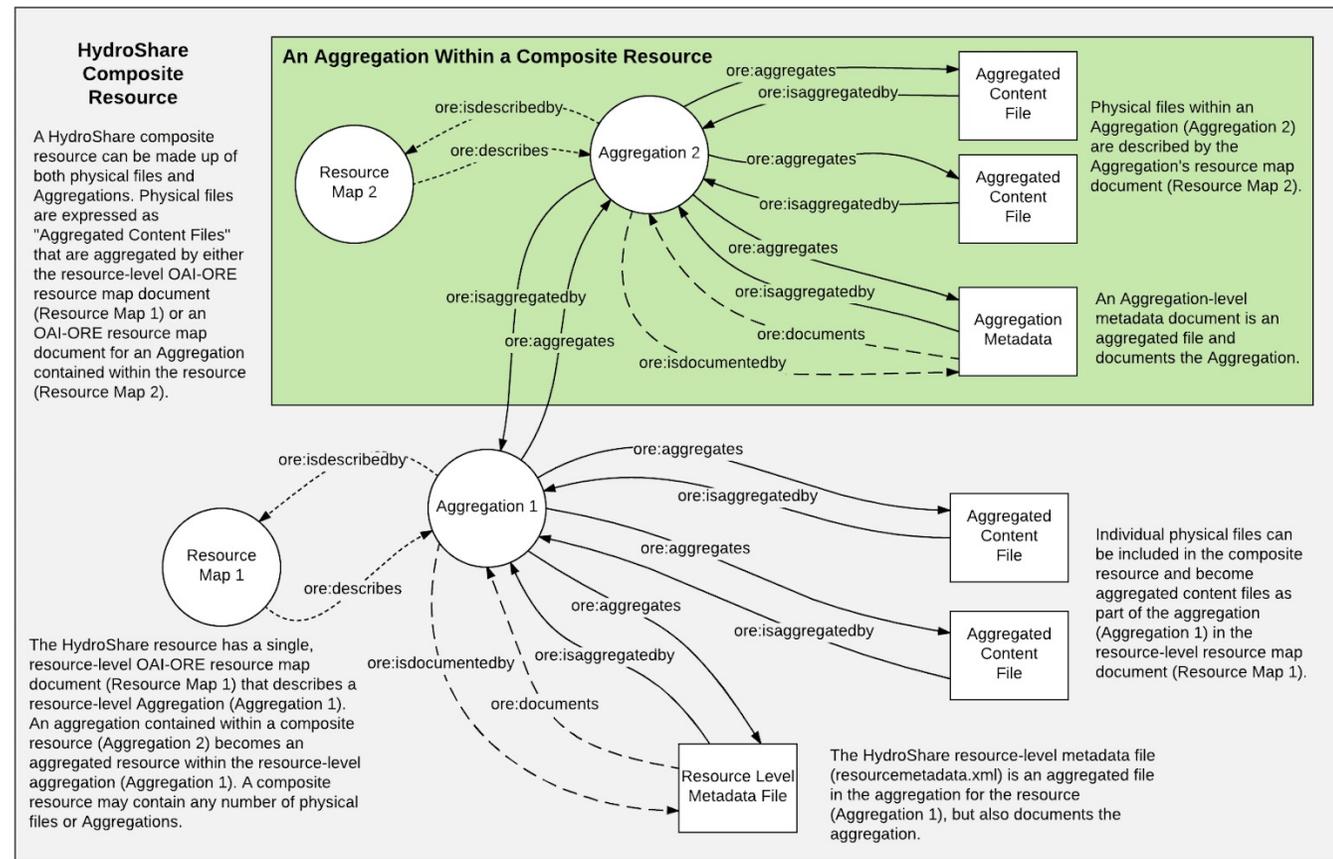
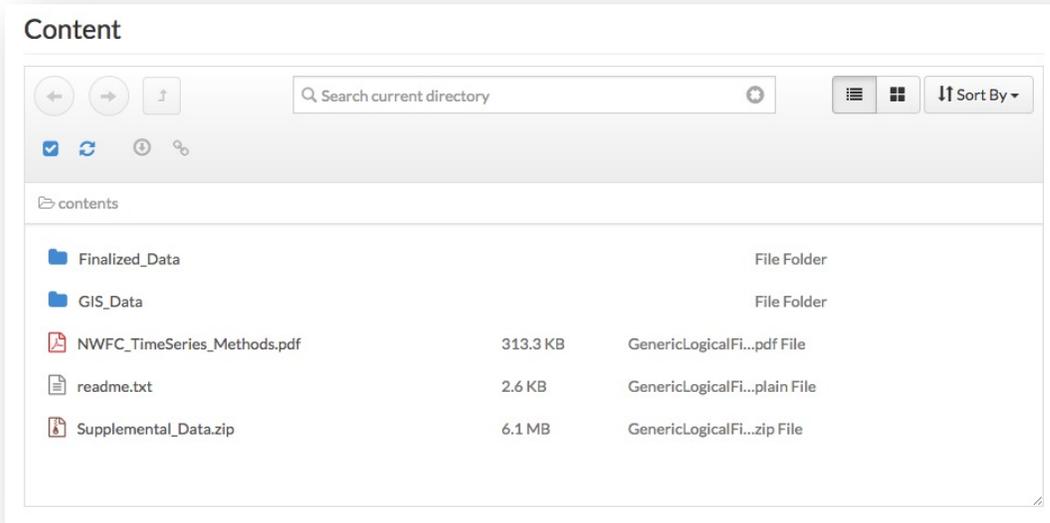
The screenshot shows the 'My Resources' page on the HydroShare website. The page has a navigation bar with 'HYDROSHARE', 'MY RESOURCES', 'DISCOVER', 'COLLABORATE', 'APPS', and 'HELP'. Below the navigation bar, there is a search bar and a '+ Create new' button. A filter sidebar on the left allows users to filter resources by ownership and visibility. The main content area displays a table of resources with the following columns: Type, Title, Owners, and Last modified. The resources listed include:

Type	Title	Owners	Last modified
Document	Survey of Stormwater Managers in Utah	IUTAH Data Manager	June 28, 2016, 6:25 p.m.
Document	Share and Publish your Data and Models with HydroShare	David Tarboton	June 28, 2016, 11:58 a.m.
Document	Cross section survey, Northwest Field Canal at 200 South	Jeffery Horsburgh	June 27, 2016, 7:40 p.m.
Figure	Water Temperature in the Little Bear River at Mendon Road near Mendon, UT	Jeffery Horsburgh	June 21, 2016, 7:14 p.m.
Document	HEC-HMS Version 4.1	David Tarboton	June 16, 2016, 8:20 p.m.
Model	Logan Digital Elevation Model	David Tarboton	June 14, 2016, 1:30 p.m.
Model	Logan Specific Catchment Area	David Tarboton	June 14, 2016, 1:30 p.m.
Document	100yr flood every 3 years	Matthew Turner	May 3, 2016, 11:44 p.m.
Document	Hydrology Domain Cyberinfrastructures: Successes, Challenges, and Opportunities	Jeffery Horsburgh	Dec. 17, 2015, 9:07 a.m.
Document	CI-WATER Workshop - Tethys Provo Dam Break App Data	Nathan Swain	July 15, 2015, 8:56 p.m.

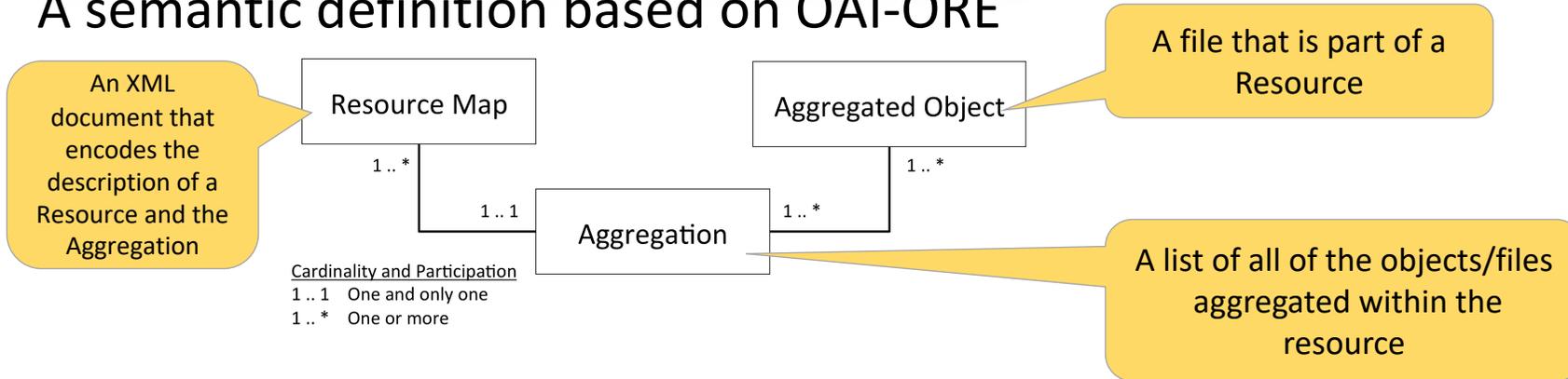
At the bottom of the page, there is a dark blue footer with 'CONTACT US' (Email us at [hydroshare.org](mailto:hydroshare.org)), 'FOLLOW' (with social media icons for Twitter, Facebook, YouTube, and LinkedIn), and 'OPEN SOURCE' (HydroShare is Open Source. Find us on [GitHub](#)).

# HydroShare “Resources”

## A file/content – based definition



## A semantic definition based on OAI-ORE



A profile of the Open Archives Initiative's Object Reuse and Exchange (OAI-ORE) standard



# HydroShare “Resource” as a Data Science Enabler

- When creating reproducible data science workflows – how to organize?
  - Eventual goal is to share the analysis
  - Need to be able to get data/code into a repository
  - Need straightforward ways to organize the content used
    - Potentially inputs, outputs, code, etc.
- The HydroShare Resource is a great organizing container
  - Think of it as a “Project Directory”
  - Existing Resource Data Model
  - Machine readable semantic representation of structure
  - Flexible
  - Existing “aggregation” types identify commonly used data
  - Can map the whole thing to Python for easy manipulation
  - Already handled by a repository!



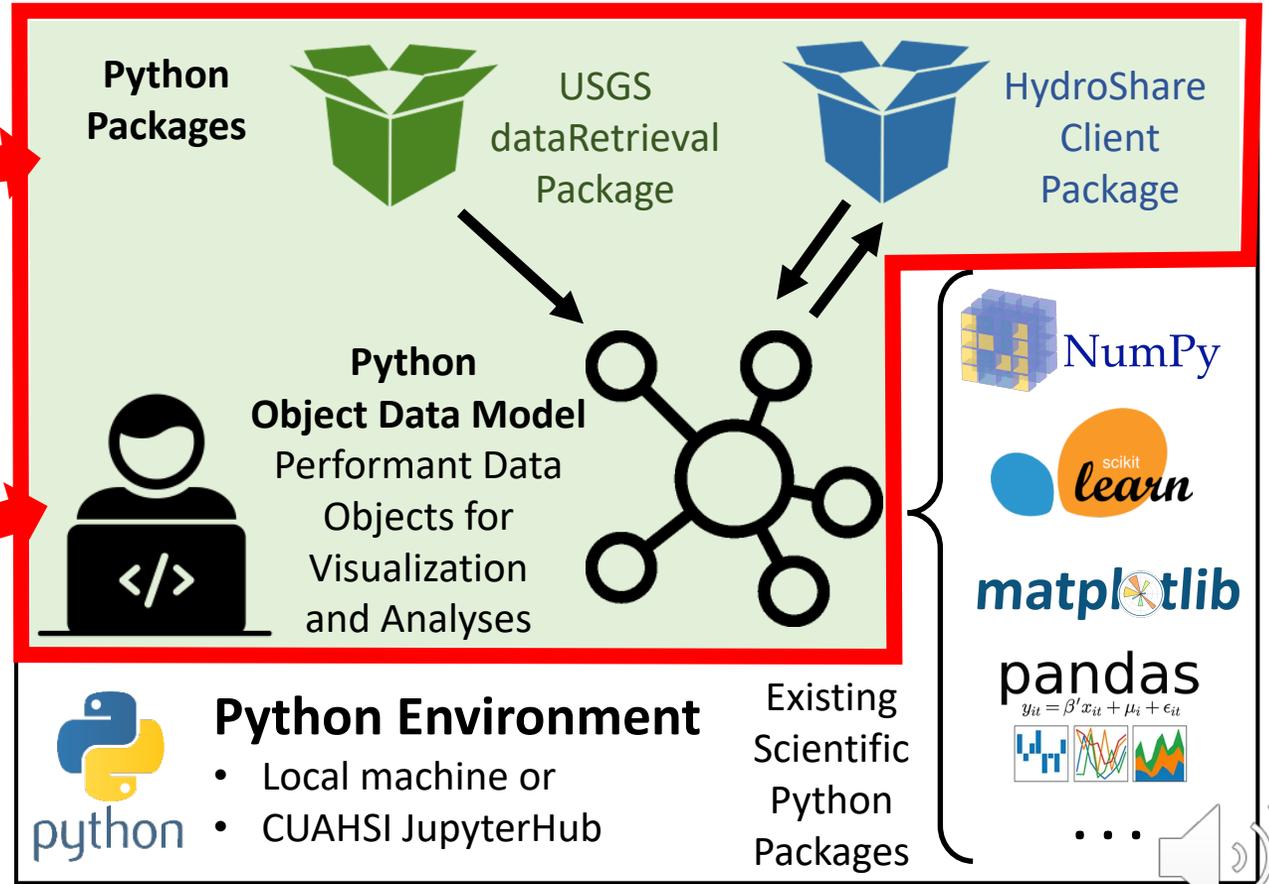
# The tools needed to make this work



Data repositories

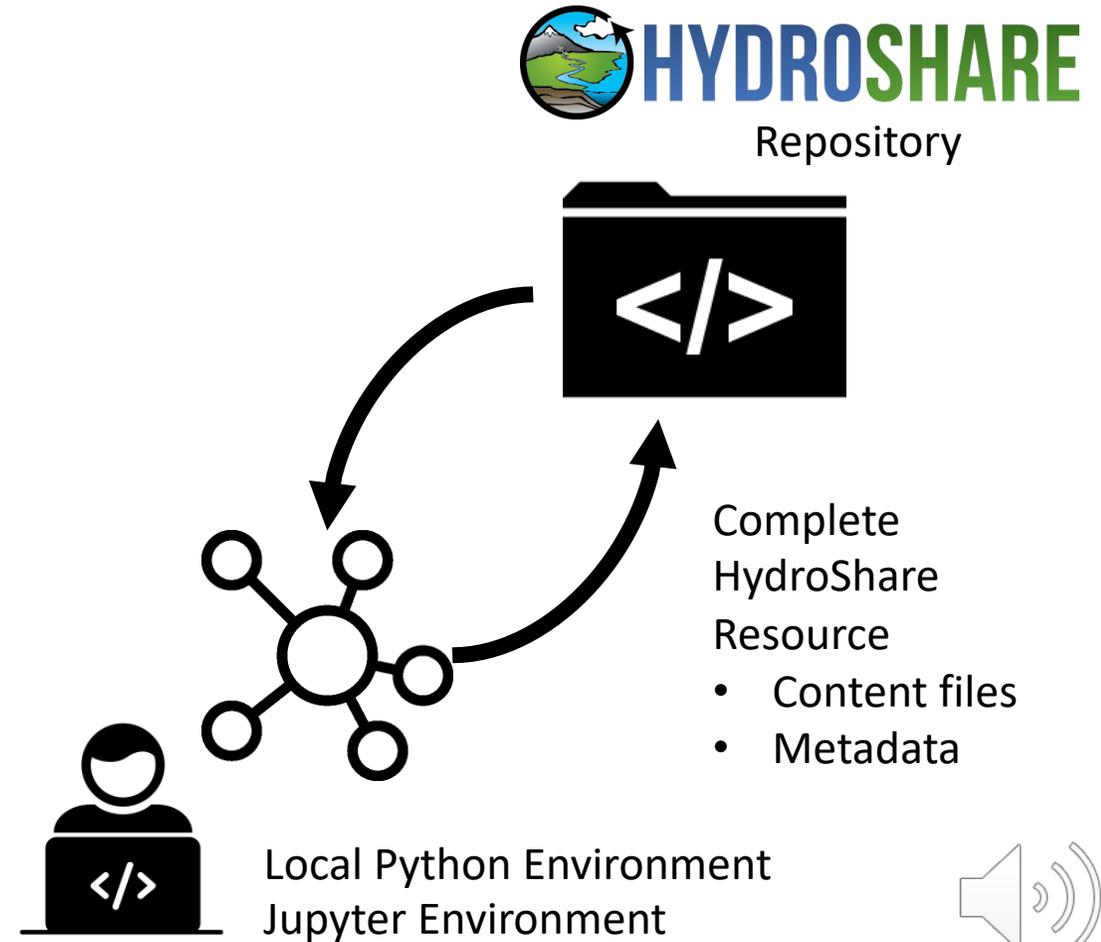
Tools for accessing and interacting with those repositories

A Python representation of the data retrieved that can be operated on using existing data science tools



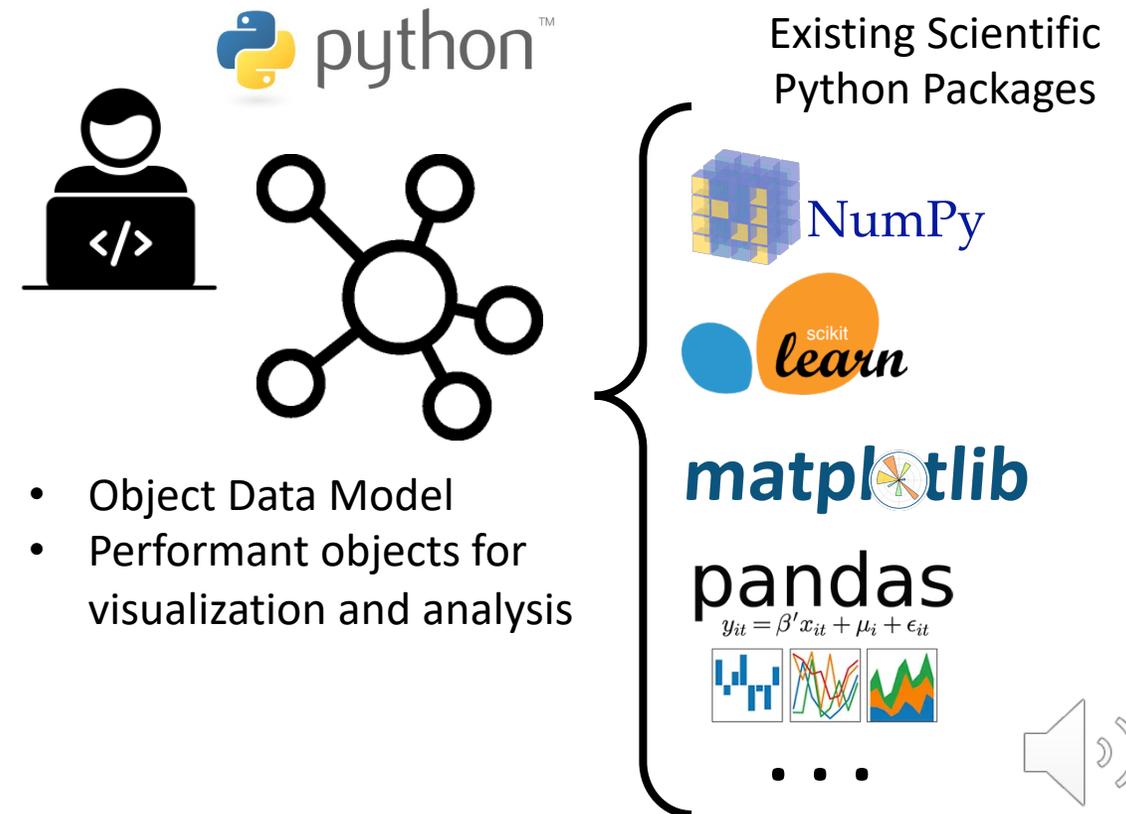
# HydroShare Python Client (hsclient)

- A set of Python functions for interacting with the HydroShare repository
- (Object) Structure of HydroShare resources is specified in the OAI-ORE RDF/XML resource map documents
- hsclient translates this structure to a Python object representation
  - Read the structure and metadata of a resource into Python objects
  - Manipulate it in your Python environment (local or Jupyter)
  - Save that structure back to HydroShare
  - Modify RDF/XML outside of HydroShare and send those files back to be ingested



# A flexible water-data science object data model (hsmodels)

- Extending the HydroShare Resource Data Model to Python analysis environments
- Maps HydroShare's resource metadata to a set of Python objects (classes) defined using pydantic models
- Maps common water-related data types (HydroShare content types) to performant data structures within Python
- Load and stage data for visualization/analysis using common Python tools (pandas, matplotlib, etc.)



# HydroShare Python Client 'hsclient' package

- A set of Python functions for interacting with HydroShare
  - Resource creation/editing
  - Interact with resources in an interactive, object-oriented way
  - Integrate HydroShare resources into data science workflows
  - Reduce the time required to get data for analysis and then save results
- Example Jupyter Notebooks:  
<https://www.hydroshare.org/resource/7561aa12fd824ebb8edbee05af19b910/>
- GitHub Repository:  
<https://github.com/hydroshare/hsclient>

The image displays two overlapping screenshots. The background screenshot shows the GitHub repository for 'hydroshare/hsclient'. The repository page includes a file browser with folders like '.github/workflows', 'binder', 'docs', 'fastapi', 'hsclient', 'tests', and files like '.gitignore', 'LICENSE.txt', 'Makefile', 'README.md', 'mkdocs.yml', 'requirements.txt', 'setup.cfg', and 'setup.py'. The repository is public and has 14 branches and 7 tags. The foreground screenshot shows the PyPI package page for 'hsclient 0.1.7'. The page features a search bar, a 'pip install hsclient' button, and a 'Latest version' badge. The project description states: 'A python client for interacting with HydroShare in an object oriented way.' The page also includes navigation links, project links, statistics (Stars: 0, Forks: 1), and instructions on how to install the package via PyPI.

# USGS dataretrieval Python package

- Python mirror of the R dataRetrieval tool
- Currently has most of the same functions
- Very similar results
- Collaborating with Timothy Hodson at USGS
- Example Jupyter Notebooks: <https://www.hydroshare.org/resource/c97c32ecf59b4dff90ef013030c54264/>

<https://github.com/USGS-python/dataretrieval>

The image shows two overlapping browser windows. The background window is the GitHub repository for 'USGS-python / dataretrieval'. It displays the repository structure, including folders like '.github/workflows', 'dataretrieval', 'demos', and 'tests', and files like '.gitignore', 'CONTRIBUTING.md', 'LICENSE.md', 'README.md', 'requirements.txt', and 'setup.py'. The README.md file is open, showing the title 'dataretrieval: Download hydrologic and climate data' and a section 'What is dataretrieval?' which describes it as a Python alternative to the R dataRetrieval package. Below the text, there is a code snippet for installing the package and specifying a site code.

```
# first import the functions for downloading data from the USGS
import dataretrieval.nwis as nwis

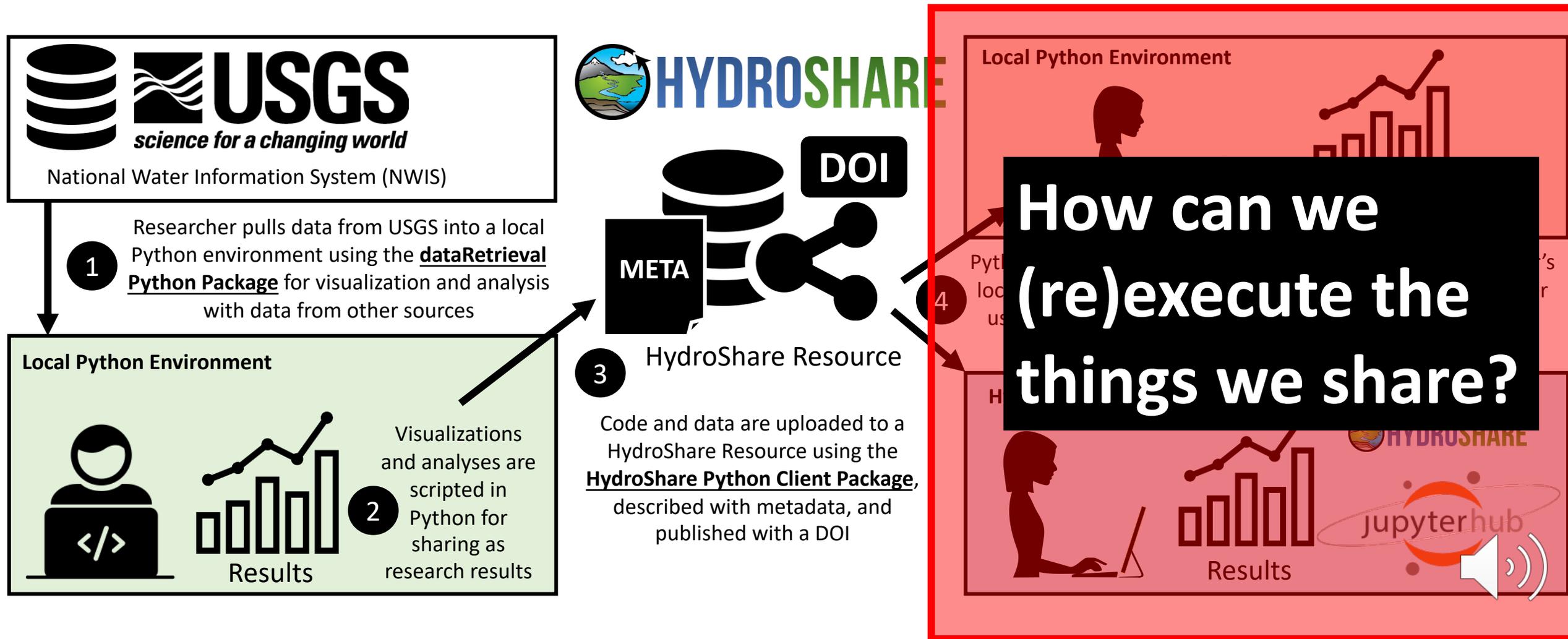
# specify the USGS site code for which we want data.
site = '03339000'
```

The foreground window is the PyPI package page for 'dataretrieval 0.5'. It features a search bar, navigation links (Help, Sponsor, Log in, Register), and a 'pip install dataretrieval' button. The 'Release history' section shows a vertical timeline of versions: 0.5 (Nov 12, 2020), 0.4 (May 14, 2019), 0.3 (Apr 29, 2019), 0.2 (Apr 18, 2019), and 0.1 (Sep 27, 2018). The page also includes statistics, meta information (License: MIT, Author: Timothy Hodson), and maintainers (thodson-usgs).

This is a detailed view of the PyPI package page for 'dataretrieval 0.5'. The page is blue and white. At the top, there is a search bar and navigation links. The main heading is 'dataretrieval 0.5' with a 'Latest version' button. Below this is a 'pip install dataretrieval' button. The page includes a 'Release history' section with a vertical timeline of versions: 0.5 (Nov 12, 2020), 0.4 (May 14, 2019), 0.3 (Apr 29, 2019), 0.2 (Apr 18, 2019), and 0.1 (Sep 27, 2018). The page also includes statistics, meta information (License: MIT, Author: Timothy Hodson), and maintainers (thodson-usgs). At the bottom, there are links for 'Help', 'About PyPI', 'Contributing to PyPI', and 'Using', along with a speaker icon.

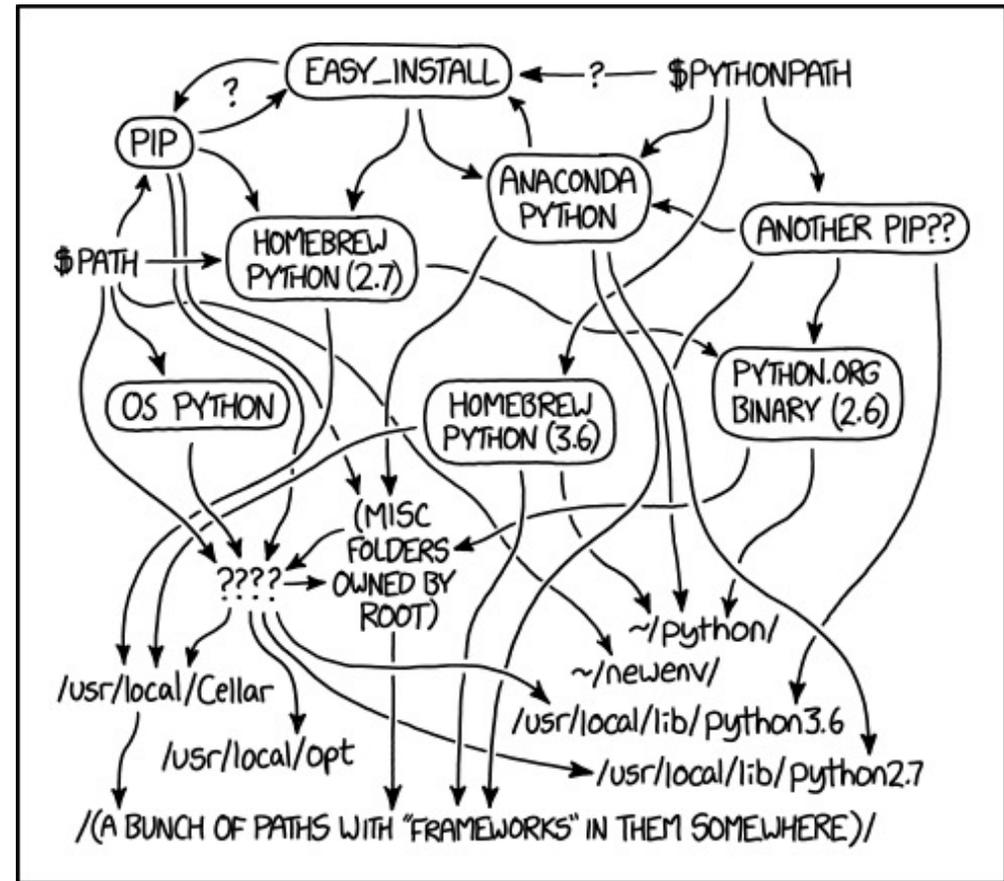
# Connecting Visualization and Analysis with an Online Repository

- Better enabling collaborative data science workflows and reproducibility



# One Option: Local Python Environment

- Set up a local environment
- Get the Python version right
- Install the right versions of all of the packages
- Cross your fingers and hope it will run . . .
- Virtual environments can help, but this can still be challenging



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

<https://xkcd.com/1987/>



# Collaborative and interactive computing for water-data scientists

## CUAHSI JupyterHub – Google Cloud

- Supports “unlimited” users (\$)
- Capable of creating classroom/workshop specific instances
- Completely customizable and uses the latest JH software

## CyberGIS-Jupyter for Water

- More available compute resources

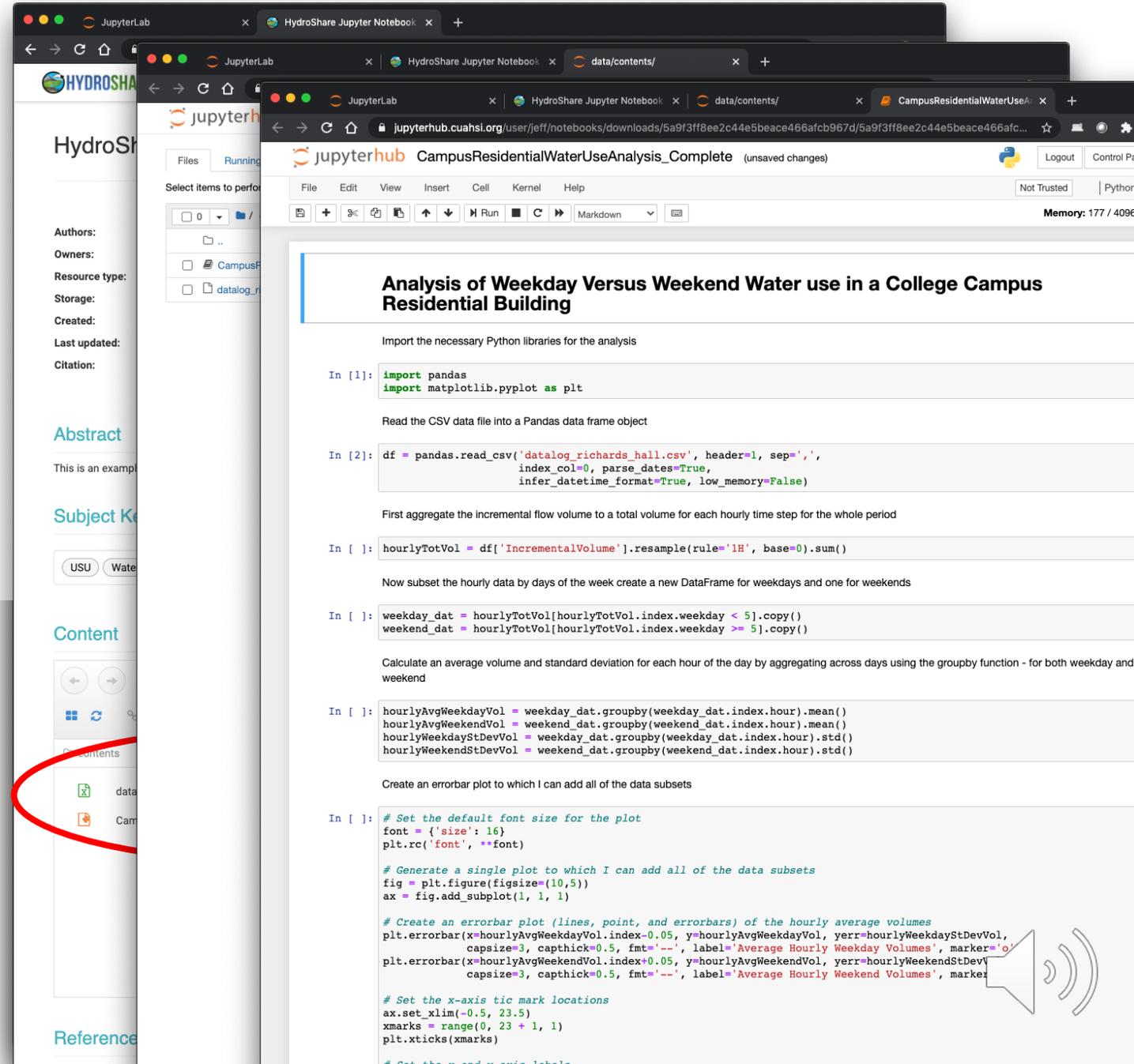
## MATLAB Online

- 50 concurrent users
- Livescript support and m-file
- 20+ toolboxes



# Creating and Sharing Reproducible Analyses

- Reproducible analyses: Sharing data and code together in a repository
- Linking repositories with computational environments
- Repositories as a gateway to high performance computing and cloud services



The screenshot displays a JupyterLab environment with a notebook titled "Analysis of Weekday Versus Weekend Water use in a College Campus Residential Building". The notebook content includes the following code blocks:

```
import pandas
import matplotlib.pyplot as plt

df = pandas.read_csv('datalog_richards_hall.csv', header=1, sep=',',
                    index_col=0, parse_dates=True,
                    infer_datetime_format=True, low_memory=False)

hourlyTotVol = df['IncrementalVolume'].resample(rule='1H', base=0).sum()

weekday_dat = hourlyTotVol[hourlyTotVol.index.weekday < 5].copy()
weekend_dat = hourlyTotVol[hourlyTotVol.index.weekday >= 5].copy()

hourlyAvgWeekdayVol = weekday_dat.groupby(weekday_dat.index.hour).mean()
hourlyAvgWeekendVol = weekend_dat.groupby(weekend_dat.index.hour).mean()
hourlyWeekdayStDevVol = weekday_dat.groupby(weekday_dat.index.hour).std()
hourlyWeekendStDevVol = weekend_dat.groupby(weekend_dat.index.hour).std()

font = {'size': 16}
plt.rc('font', **font)

fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1, 1, 1)

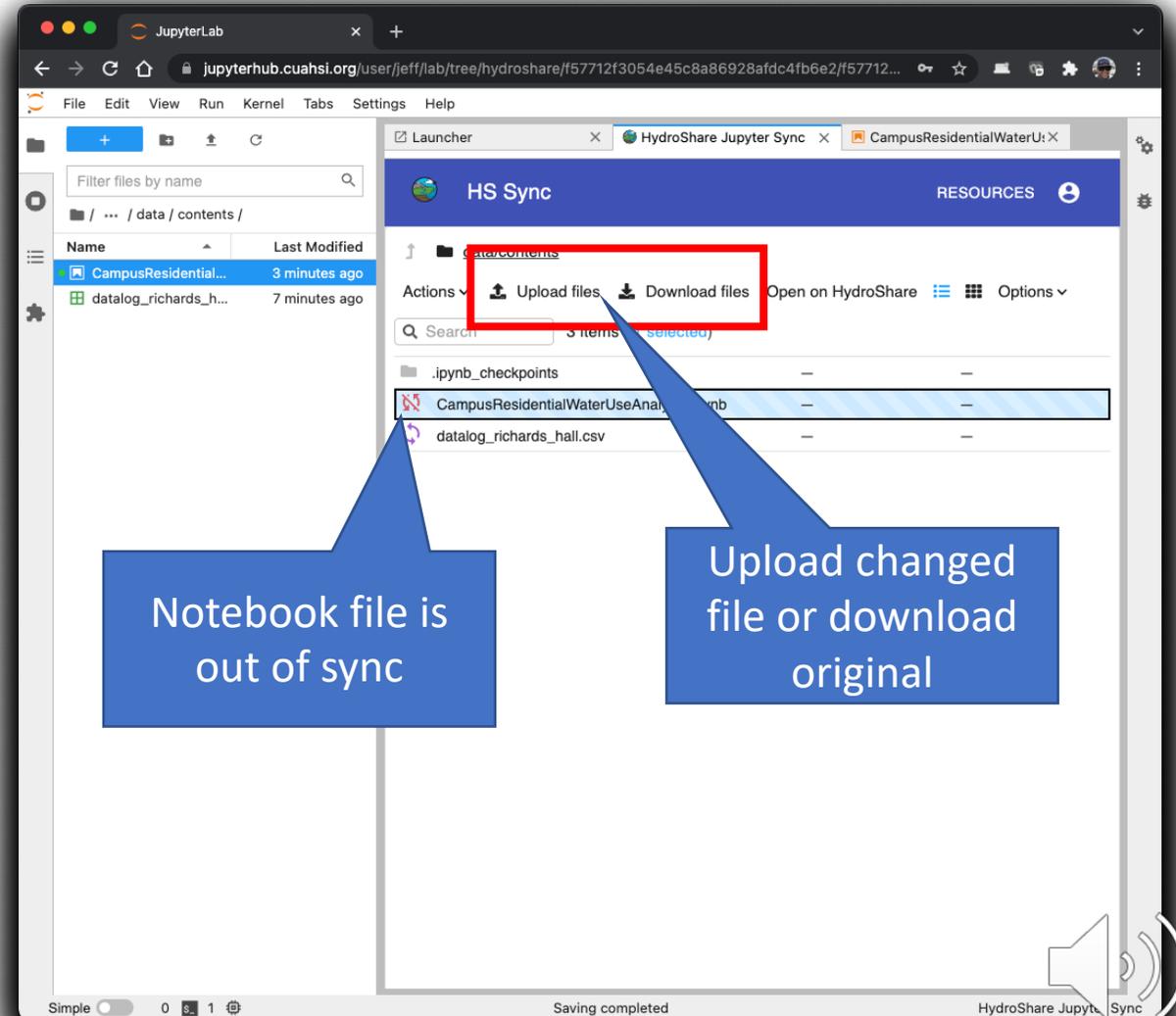
plt.errorbar(x=hourlyAvgWeekdayVol.index-0.05, y=hourlyAvgWeekdayVol, yerr=hourlyWeekdayStDevVol,
            capsized=3, capthick=0.5, fnt='--', label='Average Hourly Weekday Volumes', marker='o')
plt.errorbar(x=hourlyAvgWeekendVol.index+0.05, y=hourlyAvgWeekendVol, yerr=hourlyWeekendStDevVol,
            capsized=3, capthick=0.5, fnt='--', label='Average Hourly Weekend Volumes', marker='o')

ax.set_xlim(-0.5, 23.5)
xmarks = range(0, 23 + 1, 1)
plt.xticks(xmarks)
```

# CUAHSI JupyterSync App using hsclient

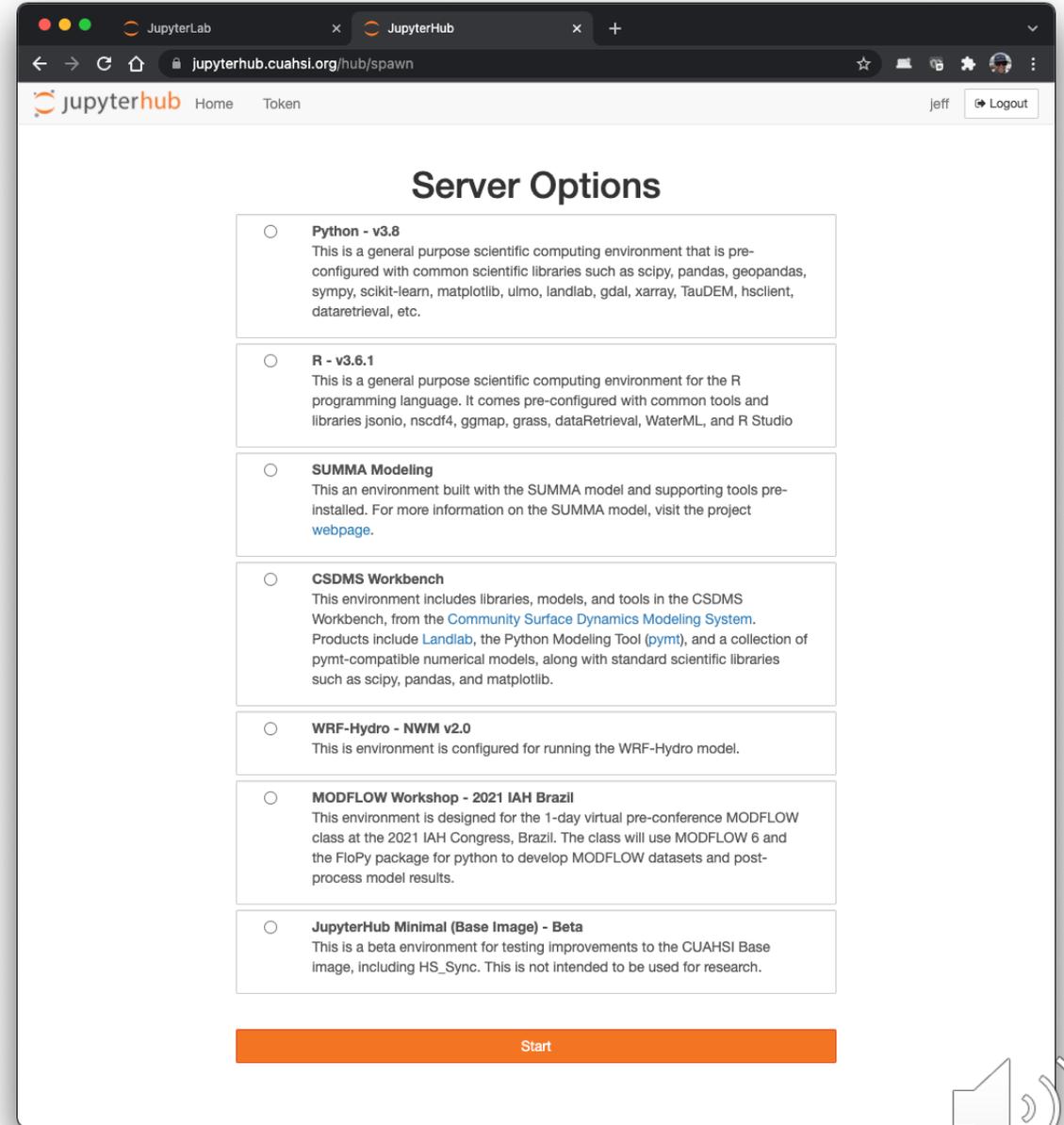
1. Launch CUAHSI Jupyterhub ([jupyterhub.cuahsi.org](http://jupyterhub.cuahsi.org))
2. Launch the Jupyter Sync App
3. Choose a HydroShare resource to work with
4. Select files to download to Jupyter environment
5. Open a file to edit or execute
6. Make changes to the file in the Jupyter Environment
7. Upload changed file to HydroShare or download original file to replace

Work by Tony Castronova and Austin Raney and students from Olin College of Engineering



# HydroShare's Linked JupyterHub Environments

- Better because I don't have to set up the environment
- Some nice tools for interacting with HydroShare
- But, I still need an environment
- Some potential limitations:
  - Software dependencies
  - Legacy code
  - Long run times
  - Complicated and large input/output files



The screenshot shows a web browser window with the URL `jupyterhub.cuahsi.org/hub/spawn`. The page title is "Server Options". It features a list of seven server environments, each with a radio button for selection and a brief description:

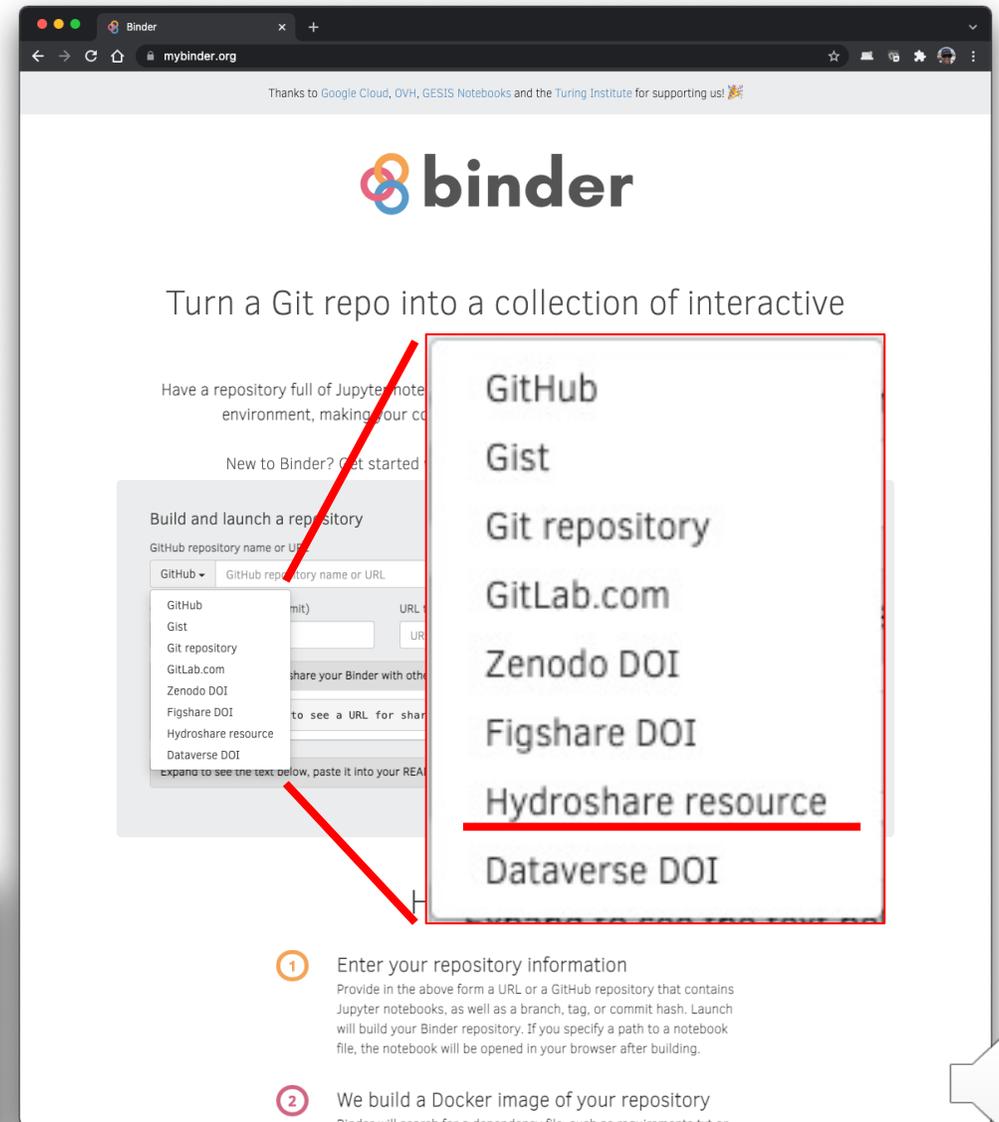
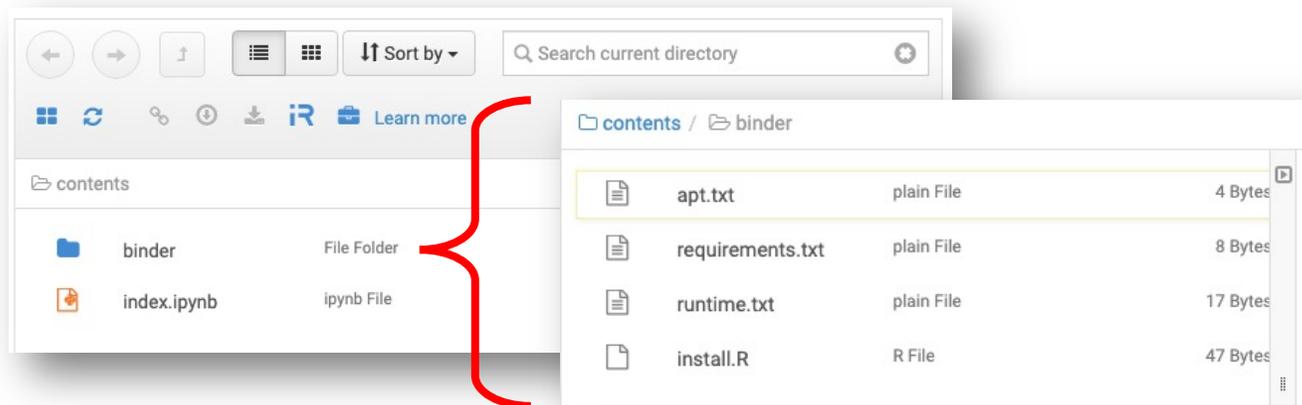
- Python - v3.8**: This is a general purpose scientific computing environment that is pre-configured with common scientific libraries such as scipy, pandas, geopandas, sympy, scikit-learn, matplotlib, ulmo, landlab, gdal, xarray, TauDEM, hscient, dataretrieval, etc.
- R - v3.6.1**: This is a general purpose scientific computing environment for the R programming language. It comes pre-configured with common tools and libraries jsonio, nscdf4, ggmap, grass, dataRetrieval, WaterML, and R Studio
- SUMMA Modeling**: This an environment built with the SUMMA model and supporting tools pre-installed. For more information on the SUMMA model, visit the project [webpage](#).
- CSDMS Workbench**: This environment includes libraries, models, and tools in the CSDMS Workbench, from the [Community Surface Dynamics Modeling System](#). Products include [Landlab](#), the Python Modeling Tool ([pymt](#)), and a collection of pymt-compatible numerical models, along with standard scientific libraries such as scipy, pandas, and matplotlib.
- WRF-Hydro - NWM v2.0**: This is environment is configured for running the WRF-Hydro model.
- MODFLOW Workshop - 2021 IAH Brazil**: This environment is designed for the 1-day virtual pre-conference MODFLOW class at the 2021 IAH Congress, Brazil. The class will use MODFLOW 6 and the FloPy package for python to develop MODFLOW datasets and post-process model results.
- JupyterHub Minimal (Base Image) - Beta**: This is a beta environment for testing improvements to the CUAHSI Base image, including HS\_Sync. This is not intended to be used for research.

At the bottom of the page, there is a large orange button labeled "Start".



# Improving Reproducibility with Binder

- Custom computing environments
- Free, but limited resources
- Can lower the barrier of entry for water scientists
- Integrated with HydroShare
- Users can start with a HydroShare base image





1664061  
1931297  
1931278

# Questions?

Jeffery S. Horsburgh

[jeff.horsburgh@usu.edu](mailto:jeff.horsburgh@usu.edu)



Utah Water Research Laboratory  
UtahStateUniversity

