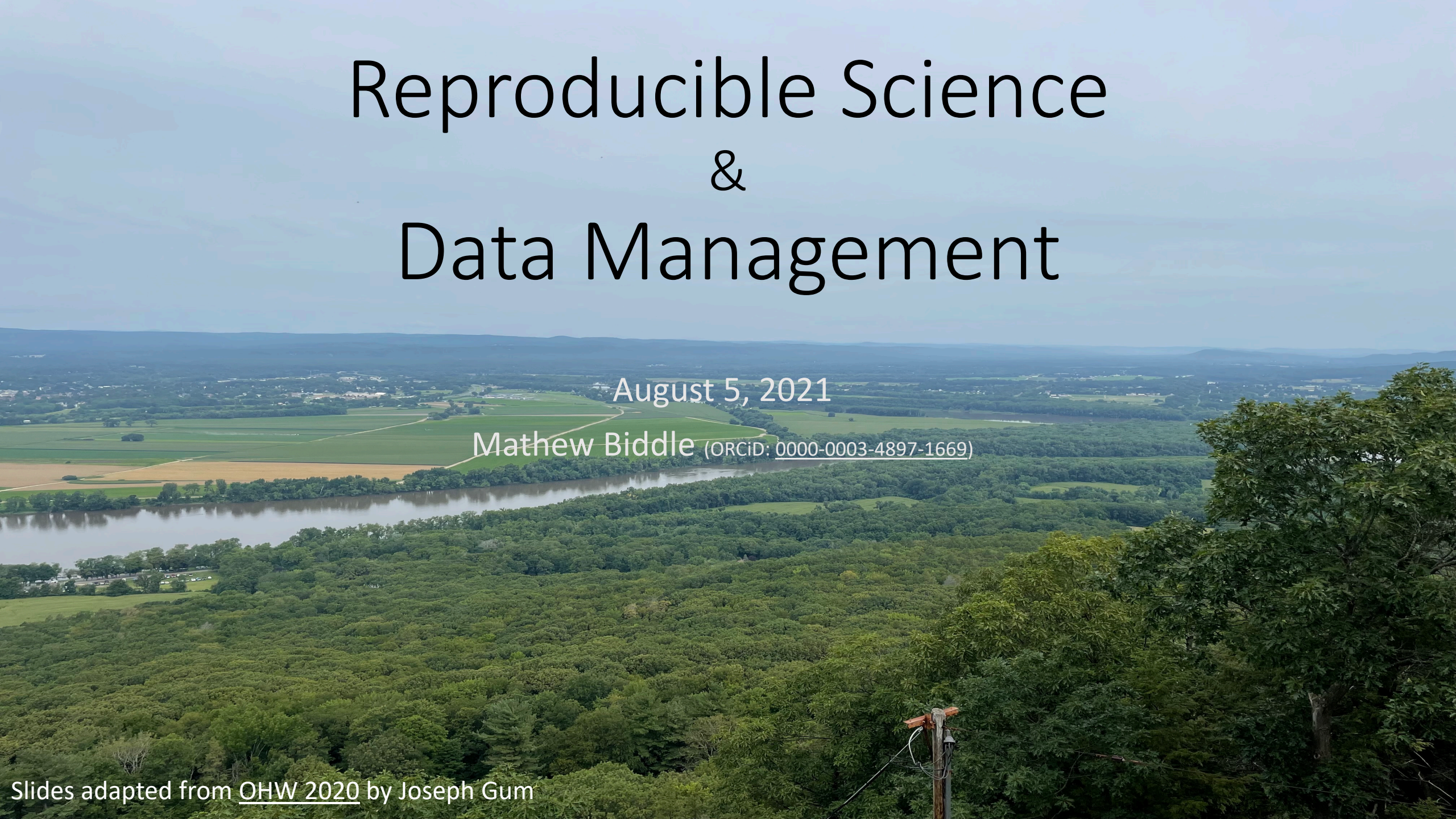
Reproducible Science
&
Data Management

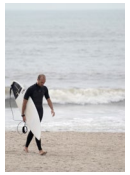
August 5, 2021

Mathew Biddle (ORCiD: 0000-0003-4897-1669)



Slides adapted from OHW 2020 by Joseph Gum

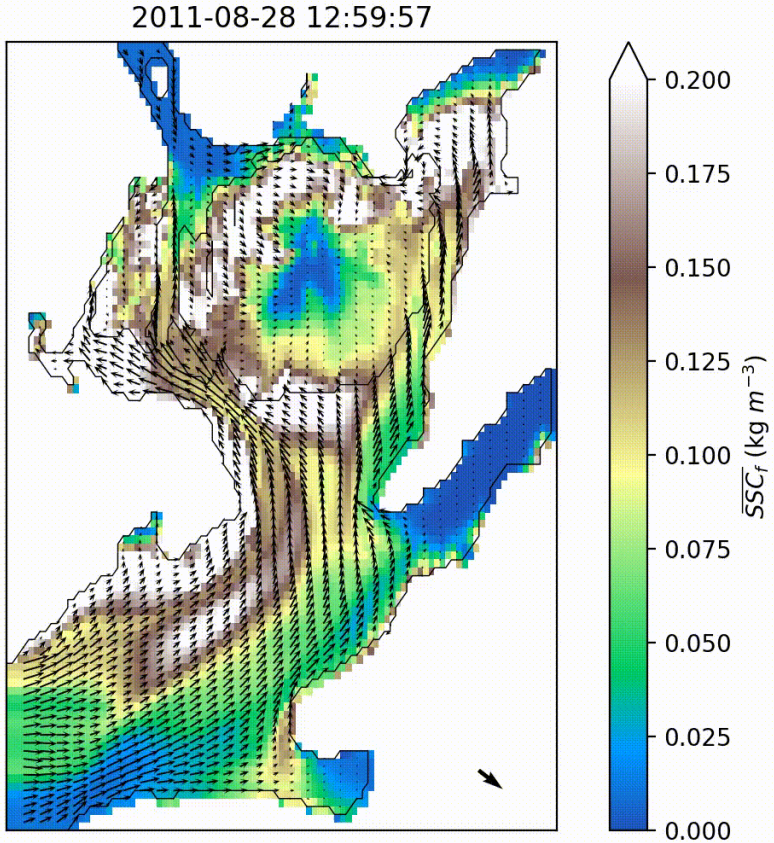
About me



B.S. Oceanography – Humboldt State University

M.Sc. Physical Oceanography – UMCES

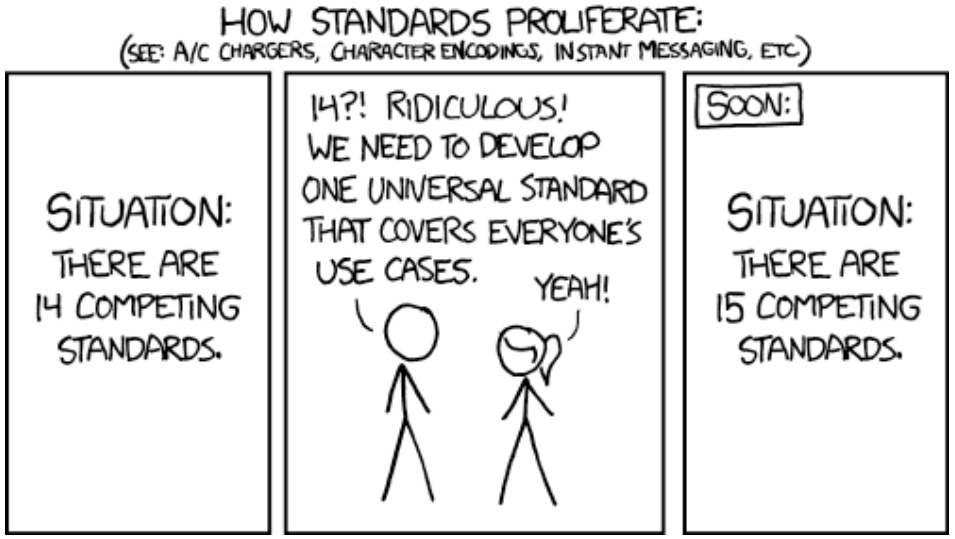
>10 years experience at various data centers/repositories



Biddle, M.M., Palinkas, C.M. & Sanford, L.P. Modeling Impacts of Submersed Aquatic Vegetation on Sediment Dynamics Under Storm Conditions in Upper Chesapeake Bay. *Estuaries and Coasts* (2021). https://doi.org/10.1007/s12237-021-00941-2

Quiz time!

- What is a repository?
  - Differences between code and data repositories?
- Who do you turn in your data/results to?
- Name a version control software (VCS)
- What is a branch in git?
- What is a package in Python?



- Have you had trouble running code from others, or sharing your code?
- Is interoperability important to you?
  - Do you feel there are too many standards to understand?

(Oceanography) Data Stewards Salad

- ESIP
- RDA
- EarthCube
- FAIR
- Ocean Best Practices
- DataONE
- IEDA
- Force11
- TDWG
- And so on…

Repeatability, Replicability, Reproducibility

- **Repeatability**: Same team, same experimental setup
- **Replicability:** Different team, same experimental setup
- **Reproducibility:** Different team, different experimental setup

- **Methods reproducibility**: provide sufficient detail about procedures and data so that the same procedures could be exactly repeated.
- **Results reproducibility**: obtain the same results from an independent study with procedures as closely matched to the original study as possible.
- **Inferential reproducibility**: draw the same conclusions from either an independent replication of a study or a reanalysis of the original study.

Plesser HE (2018) Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Front. Neuroinform*. 11:76. doi: 10.3389/fninf.2017.00076

FAIR Principles

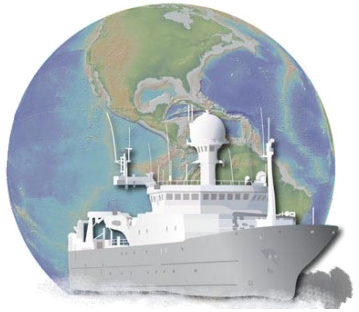
- Make data easier to use: “guidelines to improve the findability, accessibility, interoperability, and reuse of digital assets”
- **What does FAIR mean?**
  - **Findable**
  - **Accessible**
  - **Interoperable**
  - **Reusable**

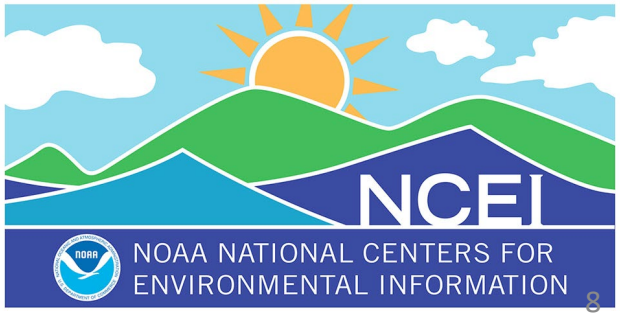
- “Is your data FAIR?” vs “Does your data follow FAIR virtues?”
  - Absolutes versus relatives – start with the relatives

https://www.go-fair.org/fair-principles/

Data Centers/Repositories

- Data centers are paid to make your data available – talk to us!
  - (It's also a requirement of your grant to submit your data)
- Use our libraries to read our data holdings, or ask us to make them
  - CCHDO/hydro, robis, gliderpy, etc
  - Feel free to contribute too!





Repositories of Methods

- Ocean Best Practices
  - DOIs for manuals and handbooks
    - How to do fieldwork, make models, etc.
  - UNESCO/IOC project
  - Internationally agreed upon methods…
    - …and publically submitted methods
- protocols.io
  - A secure platform for developing and sharing reproducible methods.

protocols.io

United Nations Educational, Scientific and Cultural Organization

Intergovernmental Oceanographic Commission

Ocean best practices

Reproducible Software: Opinionated Practices

Why spend effort for reproducibility?

- Faster deployment of code
- Faster understanding of code
- Easier extension of code
- Provenance
- It’s good for you and the community!
- See AGU guidance (https://doi.org/10.1029/2021EA001797)
- See AMS policy on software (link)

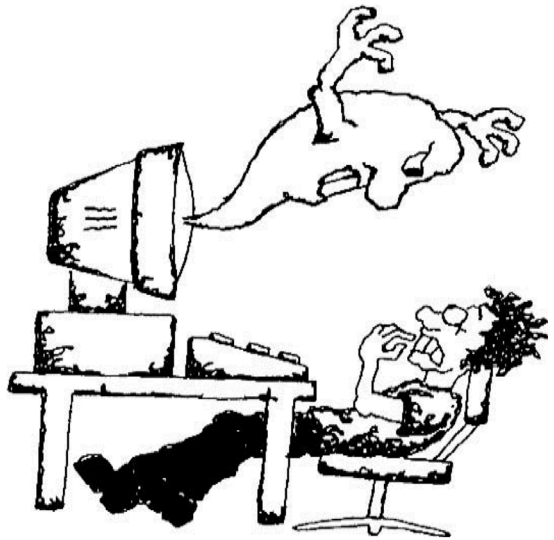
What is reproducibility?

- Read more than write – can you read it one month, year from now?
- New installs over current- can you install it one month, year from now?
- Code brittleness – can you extend it one month, year from now?

Lets see what some people have said about software, data before:

The ~dangers~ of big data

- Data may be hosted in several different places, which is a waste of resources
- Reproducibility of published analyses:
  - If code isn't readable it is difficult to check
  - Code is often written in proprietary languages
- Data may be used without understanding it or updating it (QC status)
- Where should this data be hosted? Who should maintain it?



PODS-DISCO 2018, Hawaii

How should we organize and share our datasets?

Access and reproducibility are key!
Projects like BCO-DMO and Earthcube are creating standards and improving access

Coding standards are important for sharing and reproducibility (IPython/Jupyter notebook/Github)

How can we discover the relationships in large multi-instrument data streams? How can we relate data collected from new technology to old data?

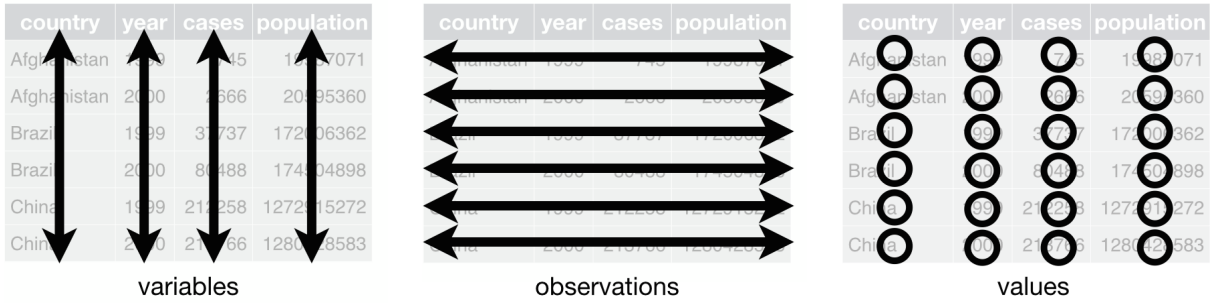
How to encourage scientists to participate?
Requirements by journals and funding bodies
Building community and tools to keep data sustainable
Are there non-science funding resources to facilitate this? Creating positions such as "Information Officer"

PODS-DISCO 2018, Hawaii

Community Decisions - Oceanography

- Formats:
  - netCDF (CF, ACDD, NCEI netCDF Templates, IOOS Metadata Profile, etc.)
  - DarwinCore – Occurrence/Environmental Observations
  - MiXS – genomics
  - BIOM – contingency tables
  - Descriptive metadata (ISO 19115-2, EML, etc.)
- Standard Vocabularies
  - Regulated vocabularies with strict definitions
  - Propose new names, pass committee approval
  - CF – Physical parameters
  - NERC Vocabulary Server – Collection of vocabularies
  - WoRMS / ITIS – Taxonomy
  - DarwinCore – Occurrence/Environmental Observations
- Cloud: Pangeo
  - Standardizing environments to make it easier to run code
- Tools
  - ioos_qc (QARTOD), compliance-checker, erddapy

The essentials for data



- Tidy data
- Follow ISO 8601 for dates
  - *YYYY-MM-DD*T*HH:mm:ss*Z (eg. 2021-08-05T12:38:22Z)
  - Include the time zone!
- Latitude and Longitude in decimal degrees WGS84.
- Identify units of measure.
- Don’t imbed data in the column headers.

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

How do we standardize our coding practices?

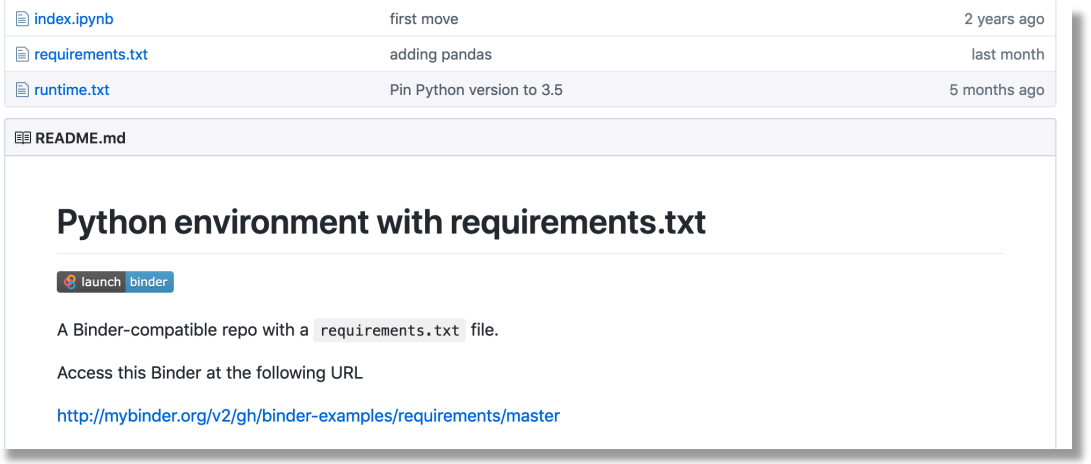
- “Opinionated” guidelines – PEP-20, ”This is the way we are doing things”
- More people following the same opinionated practices – easier to share code!
- PLoS: Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks (link)
- Software tools to help write opinionated Python
  - Conda/Conda-forge
  - Black
  - isort (non-notebooks, production code)
- Software tools to help facilitate opinionated Python
  - Git/Github
  - Binder
  - Sphinx (numpy docstrings)
- Science decisions made by communities
  - netCDF-CF
  - Pangeo

Working environments

- Simplify installing and setting up an environment to run code
- Compartmentalization makes it easier to prevent unforseen mixing that leads to problems
- Package manager:
  - Conda + conda-forge: easier installation of dependencies, environments
  - Pip: standard Python tool; if all of your packages are in there, fine! (probably not)
- Virtualization:
  - Conda provides virtual environments like virtualenv
  - Docker, other tools are good for deployment

Platforms for running notebooks online

- Binder - https://mybinder.org/
  - Plug-in to GitHub
- Google Colab - https://colab.research.google.com/
  - Straight from Google Drive
- Allow others to experiment with your code before downloading
- Make DOI notebooks runnable

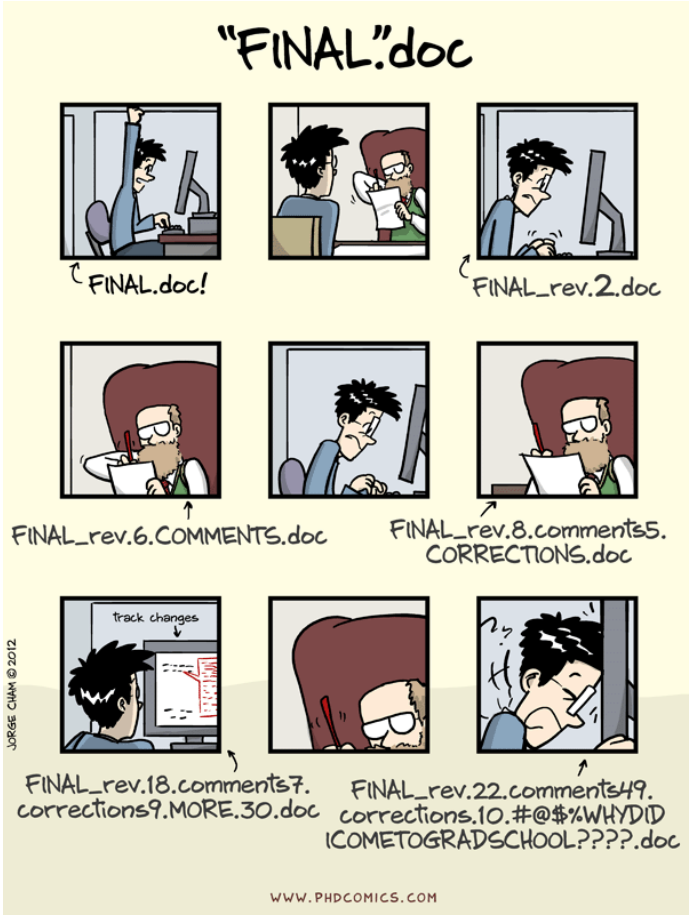


Version Control: git, Github

- Easier to keep track of yours/others changes
- Remote copies as failsafes in case of computer failure – but not a replacement for backups

- Commits should be “atomic”
  - Commits are fully formed, not half-done
  - Related edits are grouped into a single commit
  - Commits are small (one change) and often
- Use branches to try new ideas without polluting the master
- Want more git tutorials?

- Github – one of many places hosting code repos



Code Formatting

- Black: “The Uncompromising Code Formatter”
- isort: Sort and trim imports; for production code, not notebooks

```
# in:

def very_important_function(template: str, *variables, file: os.PathLike, engine: str, header: bool = True
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, 'w') as f:
        ...

# out:

def very_important_function(
    template: str,
    *variables,
    file: os.PathLike,
    engine: str,
    header: bool = True,
    debug: bool = False,
):
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, "w") as f:
        ...
```

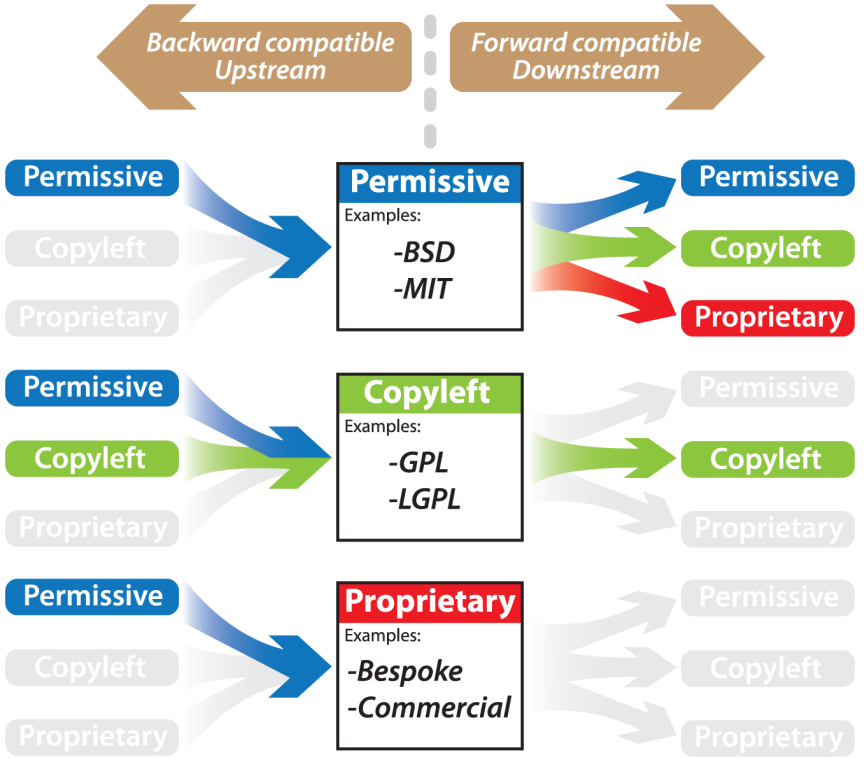
Code Readability part 2: comments and autodocs

- Docstrings: format for writing documentation in Python
- Flavors: Numpy, Google
  - Use **numpy** for your projects!
- Document/Autodocs: Sphinx
  - Automatically generate code documentation
  - Make html, latex, pdf, plain text, etc.
  - Use with Read the Docs for free hosting
- Jupyter{book}
  - Host with GitHub Pages and Actions

```
class Photo(ndarray):
    """
    Array with associated photographic information.

    ...

    Attributes
    ----------
    exposure : float
        Exposure in seconds.

    Methods
    -------
    colorspace(c='rgb')
        Represent the photo in the given colorspace.
    gamma(n=1.0)
        Change the photo's gamma exposure.

    """
```

Licensing Code and Data

- Tool to help you choose: https://choosealicense.com/
  - Software: MIT, BSD
  - Data: CC
- USA government data licenses: ODC, CC, GNU (Federal only)
- Publically funded data must be open access, data centers may license holdings
- If no license is stated, a proprietary license is assumed - "all rights reserved"



doi: https://doi.org/10.1371/journal.pcbi.1002598.g002

DOIs for Fun and Profit: Pt. 1

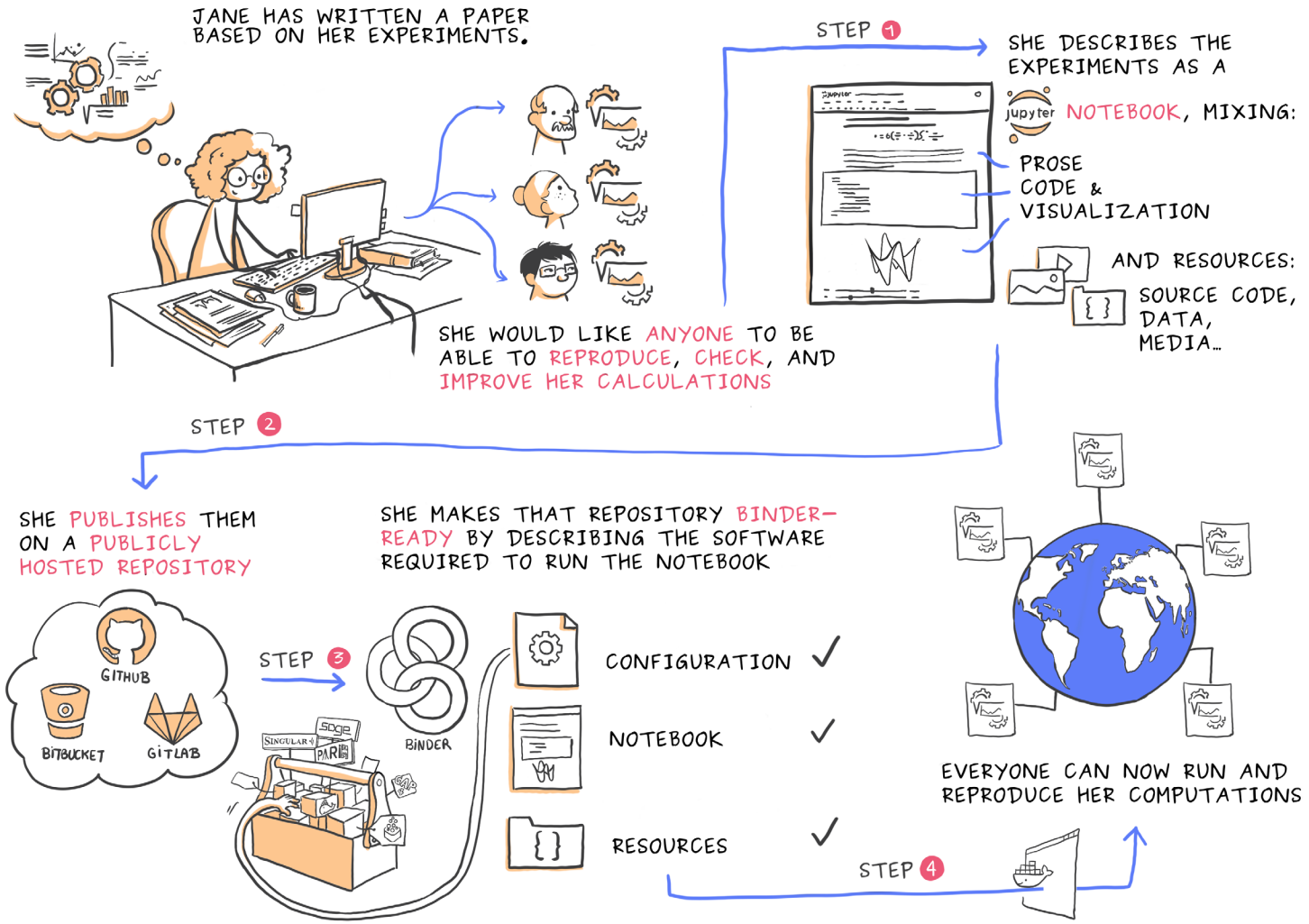
- DOI – Digital Object Identifier, can be made by many organizations
  - Your institution may do so, but you don't need to use them
- Making DOIs for code
  - https://guides.github.com/activities/citable-code/
    - You can now supply your own citation information!
  - https://www.zenodo.org/
  - CodeMeta (metadata about software)
- Making DOIs for data
  - Talk to your funder/repository first!
  - Your institutional repository may be able to make DOIs
  - Last ditch effort: zenodo

DataCite
FIND, ACCESS, AND REUSE DATA

DOIs for Fun and Profit: Pt. 2

- Citing DOIs:
  - Zenodo: lets look at **10.5281/zenodo.1256998**
    - Choose citation style
  - Online tool: https://citation.crosscite.org/
    - 10.7942/C2008W, “american-geophysical-union”, “en-US”
- These tools depend on the DOI fields being filled out properly, so you don’t have to manually form them
  - Otherwise, follow recommended style
  - ESIP software and service citation guidelines. (link)
    - **Author. (Publication Year). Title. Version No. Publisher. PID URL. Access Date.**

Future– Executable paper



Juliette Taka, & Nicolas M. Thiéry. (2018). Publishing reproducible logbooks explainer comic strip. Zenodo. https://doi.org/10.5281/zenodo.4421040

Thank you for attending OceanHackWeek 2021!

Mathew Biddle

Data Management Analyst

Integrated Ocean Observing System Office

Mathew.Biddle@noaa.gov

ORCiD: 0000-0003-4897-1669

Extra slides

Code readability – variable names

- Context is important
  - Internal functions vs public API vs data products?
  - Equations implemented in code can make sense to use shorter names
- Variable names
  - Temperature: temperature, temp, tmp, T, sea_water_temperature, sea_surface_skin_temperature, TEMPP681, TMESOT01
- Those names in blue are from standard vocabularies
  - Controlled, publically available, names added after proposal process
  - Group Examples: CF, SeaVox, NERC Vocabulary Server

Advanced Software Engineering

- Python language power features:
  - Logging, factories, list comprehensions, etc.
- APIs
  - Data: JSON, YAML, XML, REST, SOAP, etc.
- Packaging
  - Version schemes
- Continuous Integration (CI)
  - Special mention: Pre-commit
- GitHub Actions
- Tests
  - Unit, system/holistic
- Images/Containers
- “Better is the enemy of good”