Openscapes Champions Lesson Series

Open educational resources for Openscapes Champions

Openscapes team

2024-07-09

Table of contents

1 Welcome

Welcome

Hello! This is the lesson series for the **Openscapes Champions program**, an open data science mentorship program for science teams. This is a professional development and leadership opportunity for teams to reimagine data analysis and stewardship as a collaborative effort, develop modern skills that are of immediate value to them, and cultivate collaborative and inclusive research communities. Openscapes Champions is not a typical workshop — its cohort-based remote sessions for teams introduces concepts and workflows, facilitates teams to talk about problems, then go and solve them with accountability and support. It is a remote-by-design program, launched in 2019.

Figure 1: We think about open science like a landscape, with many paths forward. The Champions Program mentors researchers, meeting them at the trailhead to help them move from sad and lonely science to team science as they identify their common needs and start navigating the landscape together with a cohort of their peers. Artwork by Allison Horst.

All Champions Program resources are designed to also be a self-paced learning resource, and we know many people use the Champions Lesson Series for their own learning and for mentoring others, which is awesome. This Champions Lesson Series is improved openly and iteratively,

and the most recent version always available online for reuse and remix. Each chapter in our **Core Lessons** focus on building a mindset and skillset for collaborative, reproducible workflows and culture within teams, while developing sustained learning practices connected with broader communities.

See the Champions Program webpage for programmatic details (including frequently asked questions like what is a team). To learn what teams accomplish, explore stories from 20+ previous cohorts.

Purpose, Outcomes, Process (POP)

This is the Champions Program POP, a planning tool that we learned from the open science team at Mozilla.

Purpose: to strengthen habits for immediate benefit that will help create long-lasting resilience in teams and workflows. We'll help you reimagine data analysis and stewardship through exploring open tools and practices; develop modern skills and habits that are of immediate value to you, including confidence and agency as leaders; cultivate collaborative and inclusive research communities with a Future Us mindset, starting with your team.

Outcomes: different for everybody - this is about getting your own work done. Together we'll work on changing habits to improve your work and teamwork so you shape where you invest based on what you need. We'll develop a practice of reflecting, talking, and collaborating about data workflows with your team and community. You'll leave with a tangible Pathway (a planning document) that your team will use to help identify current practices and prioritize next steps. Your team will share your Pathway as work-in-progress during our last Cohort Call. Champions have found the Pathway valuable to communicate their needs and plans and to share with supervisors as a concrete outcome.

Process: through facilitated Cohort Calls, team-driven Seaside Chats & Coworking, and How We Work (how we describe technical and cultural norms); Read on to learn more about these below. Additionally, every Champions Cohort includes at least one Mentor from the partner organization or community. Often a Mentor has participated in a Champions Cohort and expressed their interest in helping empower their colleagues as they build their open science and leadership skills, collaborate on solutions to common problems, and modernize their workflows.

Cohort Calls

Research teams participate as a Champions Cohort with ~7 total teams over 2 months, meeting twice-monthly for five 1.5-hour sessions we call Cohort Calls. Teams focus on their own work. Each Cohort Call has two topics that introduce concepts, tooling, and examples from their peers, with time to reflect and discuss how the topics apply directly to our work.

Cohort Calls are highly-structured and designed to be engaging, requiring discussion and active participation through scaffolded Agendas with shared live notetaking and breakout-group discussions. We begin each Cohort Call with a reminder of our Code of Conduct and end each lesson with bite-sized Efficiency Tips and Inclusion Tips. Template agendas, as well as slides, tips, and other resources we use are openly available in this Google Folder, ready for reuse and remix.

Cohort Call Topics	Series Chapters	Seaside Chat Topics
1. Mindset, Better Science for Future Us	mindset, better science in less time	Pathways trailhead - where are you now
2. GitHub Clinic: Publishing, Project Management	publishing, project management	Shared organizing and onboarding
3. Team Culture, Data Strategies for Future Us	team culture, data strategies	Pathways next steps and documentation
4. Open Communities, Coding Strategies for Future Us	open communities, coding strategies	Pathways share preparation
5. Pathways Share, Next Steps	pathways	

You can find templates in this Champions Cohorts Resources public folder.

Seaside Chats & Coworking

Between Cohort Calls, teams have synchronous spaces via Seaside Chats and Coworking to meet, ask questions, and exchange knowledge. Seaside Chats and Coworking have been described by participants as one of the most valuable parts of the Champions Program because it helps strengthen habits and a culture of shared workflows and learning.

Seaside Chats (Lowndes et al. 2019) are when a team meets together independently for dedicated time for data/workflow discussions This is where teams talk and screenshare to begin identifying and addressing shared needs: Everyone has something to learn, ask, teach; don't need to be an expert in everything. Seaside Chats can include others that are not participating in the Champions Cohort. We'll provide suggestions for these meetings aligned with the Pathways Document to organize your thoughts. The main purpose is to focus on your work at hand and build the habit of talking about and strengthening shared workflows with your broader group. We suggest you schedule these Seaside Chats with your team as soon as possible.

Further, we facilitate optional **Coworking sessions**. Coworking comes in different flavors. For Champions Cohorts, coworking involves doing our own work at the same time together

with opportunities to check in, and sometimes screenshare to get feedback or solve a problem. Seeing how other people work is a big part of this transition to working more openly and collaboratively. Some teams join Coworking together, meeting for their Seaside Chat in a breakout room with the option of talking to others in the Cohort. We will share optional Coworking times where you are welcome to come do your own work in a social setting.

How We Work

"How We Work" refers to the technical and cultural norms we will establish during the Champions Cohort, which we developed through our Openscapes Flywheel (Robinson and Lowndes 2022). These approaches have become some of the most valuable things teams take with them through the Champions Program, and we love hearing how folks are taking these practices to new places as they grow as leaders.

Welcome

We are intentional to create a welcoming environment, through art, design, and clear norms. You are all welcome here. We are creating a positive learning space where everyone is welcome to ask questions and participate. We'll start off each call with a reminder of our Code of Conduct.

Space & Place

Together we will be creating space and place to learn, collaborate, and create shared workflows so that we aren't all responsible for this alone.

Our Cohort will have a single **shared folder** (via Google Drive/ Microsoft Teams) that we will share ahead of time and will be linked from the calendar invite. This will have all resources specific to our Cohort. Each Cohort Call will have a highly-structured **Agenda document** so you know what is planned — and so you can more easily catch up if your internet drops out. We use headers so that you can navigate via the Table of Contents view. (In Google Docs, you can enable Outline view by selecting View > Show outline.) These agendas are for live-note taking throughout the Cohort Call. It is our shared responsibility to contribute & help document for future us; this is the main place for nonverbal contribution, side conversations, and to reinforce ideas.

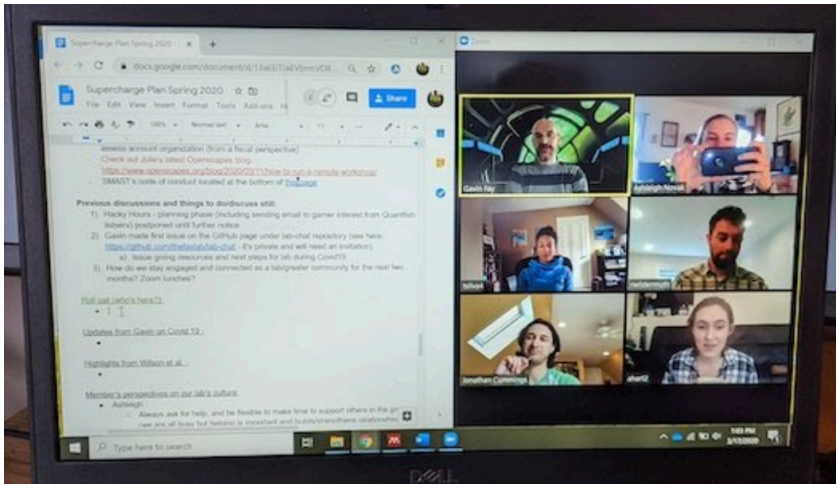


Figure 2: Example computer setup with Zoom & Google Docs side-by-side with windows narrowed to enable seeing faces and writing in the shared document. Credit: The Fay Lab

During Cohort Calls, we **encourage videos on** ("faces on") – but it is okay if this is not possible for any reason. In Zoom, you can enable Gallery View by clicking Settings > Video > Display up to 49 participants. We ask you to **mute liberally/quickly** to reduce background noise – but please unmute to speak up at any point. You'll interact in small groups via breakout groups, this will help you reflect on your work and get to know each other.

We have a **flipped approach to screensharing**. This shows up in two ways. First, we do not screenshare slides while presenting. Instead we provide links to slides and presenters indicate when everyone should advance to the next slide. This minimizes bandwith issues and enables everyone to zoom in, click on links, linger, and go back, as they prefer. Second, we do screenshare to demonstrate keyboard shortcuts, live examples of how we work, and troubleshoot. Screensharing to show current practices and ask for help is a big part of Coworking sessions and Seaside Chats.

At the start of each Cohort, we send **Google Calendar invites** to everyone for all Cohort Calls and optional Coworking sessions. These include the Zoom link (or Teams / Google Meet) to join, and a link to the Cohort's shared Google Folder so we don't all have to hunt for those each time.

We start and end on time.

Learning & Trust

Learning new things is uncomfortable and it can feel vulnerable to ask questions. Through creating a welcoming space we can invest in learning and trust together. We know there is

a range of technical experience – by design! We are all here because we want to learn and improve our work around data-intensive science. We're all imperfect and learning together – Openscapes team included. We're all accountable to each other.

Examples of how teams strengthen learning and trust within their groups are captured in our Pathways share in the final Cohort Call. Common themes are, meeting regularly, knowing who and where to ask for help, and building shared documentation and resources to reduce emails asking where things are.

Work Openly

We role model working openly throughout the Cohort by sharing ideas, resources, and screen-sharing. All our resources are created with the same tools we teach research teams (R, GitHub, Google Drive) and shared openly. During Cohort Calls, our Agenda and open facilitation style is part of working openly — we iterate openly as facilitators and mentors, for example through adjusting the timing live in the Agenda document.

We send **Digests** the week following each Cohort Call (early, on Monday or Tuesday). Each digest includes the Call's goals, links to the Call's agenda, slides, and recording, and some excerpts from the shared notes. These provide a touch point in the week between Calls. See examples: 2022-nasa-champions, 2022-noaa-afsc, 2021-fdd.

Many teams begin working more openly via shared documentation, project management, meta-data, file organization and naming, and code organization and interoperability, using tools like Google Drive/Microsoft Teams, GitHub, R, and Python.

Common Workflows

We reuse what works from other places in our work (we sometimes talk about this as "forking", borrowing a term from GitHub where you make a copy in your own workspace so you don't have to start from scratch). This often shows up in Champions Cohorts with teams reusing the Agendas structure and the concepts of Seaside Chats and Coworking with their broader research groups.

We discuss practices from different open communities, and teams contribute so that others can build from their work as well.

Inspire

Storytelling and sharing what works is so valuable for others to learn from and be inspired by. Many concepts and possibilities are new, so there is value in the idea of "show me" in a way that resonates.

Inspiring others happens in small moments, that can look like someone leaning forward in their chair saying "I want to do that" when someone else is screensharing during Coworking.

It can mean asking for 5 minutes at the start of a meeting to share something new you learned. These moments also build up to real culture change as folks give talks to their organizational leadership and broader open communities, and the Flywheel turns again.

Supercharge Your Research

If you have read this page, you are prepared for your Cohort! We're often asked what people can do to further prepare in advance. Additional suggested reading include:

Our path to better science in less time using open data science tools (Lowndes et al. 2017). This describes a marine science team's transition to open collaborative teamwork. It was the original inspiration for creating the Champions Program and heavily influences the Core Lessons.

Supercharge your research: a ten-week plan for open data science (Lowndes et al. 2019). This was co-authored with the inaugural Champions Cohort, capturing the most valuable take-aways for marine and environmental science early career faculty.

Shifting institutional culture to develop climate solutions with Open Science (Lowndes et al 2024, Ecology & Evolution). This was co-authored by Openscapes mentors across organizations – including NASA Earthdata, NOAA Fisheries, EPA, California Water Boards, Pathways to Open Science, Fred Hutch Cancer Center.

See also other publications and presentations at openscapes.org/media.

About

Openscapes is an approach and a movement that helps researchers and those supporting research find each other and feel empowered to conduct data-intensive science. Through a creative approach drawing inspiration and skills from many places, we provide structures for technical skill-building, collaborative teamwork, and inclusive community development. Our work builds from many others in the open movement.

Read more about How We Work:

- Openscapes Approach Guide
- The Openscapes Flywheel: A framework for managers to facilitate and scale inclusive Open science practices (Robinson & Lowndes 2022)
- Shifting institutional culture to develop climate solutions with Open Science (Lowndes et al 2024, Ecology & Evolution)
- 3 lessons from remote meetings we're taking back to the office (Lowndes, Cabunoc Mayes & Sansing 2020)
- How to run a remote workshop, Openscapes/Open Leaders-style (complementing blog post)

This Series Book

The Series is written (and always improving) to be used as a reference, to teach, as self-paced learning, and for reuse and remix. And also, awesomely, it's created with the same tools and practices we will be talking about: R/RStudio - originally bookdown and now quarto - and GitHub.

Citation

Please cite the Openscapes Champions Lesson Series through the project's Zenodo archive using DOI: 10.5281/zenodo.7407246. This DOI represents all versions, and will always resolve to the latest one.

The citation will look something like:

> The Openscapes Core Team, Julia Stewart Lowndes & Erin Robinson. (2022). Openscapes Champions Lesson Series (2022.12). Zenodo. https://doi.org/10.5281/zenodo.7407247

Please visit the Lesson Series DOI link to get the most recent version - the one above is not automatically generated and may be out of date if we release an updated version.

---

2 Core Lessons

Better Science for Future Us

Better science is science that is more open, reproducible, efficient, interoperable, and also more diverse, equitable, inclusive, and kind.

Future Us is the idea of considering ourselves, our teams, and communities - those who will be joining and continuing our work - whether that is in the next hour, week, or decades – and with a focus on onboarding ourselves and others to ongoing work.

Together better science for future us is an important mindset for science and society at large, and particularly important as we face global challenges in Earth & environmental science and society that are intensified with climate change.

Better science for future us is a real time investment, particularly for data-intensive research. And it is something we can do together.

This Series is going to be fun and empowering! We will talk about a lot of tools and practices to make your science more streamlined. This is really powerful, cool stuff, and not just for data: we made and published this book using the tools and workflows we'll talk about.

The first half of the Series focuses on efficiency and open culture within your team, and the second half is about sustained learning and bringing these practices to your broader communities.

Why we're here

We are passionate environmental scientists studying important, time-sensitive topics using data of all kinds. And we were never taught to work efficiently with data.

We are here because I know these files are on your computer — we all have them.

```
data_final_final.xls
data_final_usethis.xls
...
thesis_v16_new_ch1.docx
thesis_v16.docx
...
```

And we also send and receive emails with subject lines like:

```
Re:FWD:Fwd:Data question
Re:Sending again with the correct version
```

We are going to talk about how to make the data experience better, for you, your lab, your department, and beyond.

Data analysis can be inefficient and demoralizing when you're without the right tools/skills and you feel alone.

But! Open tools, practices, and communities exist that are powerful and empowering, and game-changing for science. And we can learn and use open practices for science.

They are like the Force from Star Wars:

- More powerful than you ever imagined
- Helps you solve your current question powerfully – but also broadens the scope of the questions you can ask
- Learn from jedis, pass on what you have learned, have a ton of awesome allies (and not all allies are jedis)

What to expect

This is going to be fun and empowering!

We are going to be discussing a wide range of topics and working to seed habits for you to engage and learn with them with our lab and others on campus.

Exposure to relevant tools & practices, confidence & agency to engage, community to learn with

The plan is to expose you to a lot of great tools and practices that you can have confidence using in your research. We will also spend time helping you plan how to actually incrementally weave them into your existing workflows.

The point is not to overwhelm you or make you feel like it's too late for you or that you would need to throw out and redo everything you've ever done in order to take the first step. No. By seeing what's possible and how shared practices can make your own life easier, and life easier and more streamlined and fun with your lab and beyond, you'll start experimenting with these practices and in a few years you will be working in a completely different way.

Create a shared culture – in your lab, on campus, and beyond

We're going to go through a lot and it's less important that you remember it all. More importantly, you'll know what is possible, have confidence that you can do it, and have allies so you're not alone. The main thing to take away is that there are good ways to approach your analyses; we will teach you to expect that so you can find what you need and use it! A theme throughout is that tools exist and are being developed by real, and extraordinarily nice, people to meet you where you are and help you do what you need to do. If you expect and appreciate that, you will be more efficient in doing your awesome science.

No skills required. We will strategize about general approaches, specific examples using R/RStudio and GitHub

There are no skills required to participate, and we will not be teaching hands-on how to code or set up databases. But we will be talking about how these are important and fit together in the big picture, and how to get started learning the skills you need. This is an opportunity to discuss existing tools and how to engage, meet other labs, discuss next steps, and stay accountable.

We'll talk about tools and practices broadly, but also with specific examples using R and GitHub. Won't that software eventually become outdated you say — is it worth learning them over something else? The answer is yes, software will change and become outdated; it always has. But seeing what is possible and becoming versed in embracing existing architecture and practices will set you up to make whatever transition comes, and you will make this transition

with the community, not alone. Your skills will be transferrable skills as the actual software changes. Analogy: if you learn one musical instrument, you will be able to learn another one more fluidly than if you have never learned one to begin with because maybe you can read music, understand something about timing and rhythm, etc.

Everyone is coming with different experiences & expectations

Everyone in this workshop is coming from a different place with different experiences and expectations. But everyone will learn something new here, because there is so much innovation in the data science world. You are encouraged to ask questions and answer those of others.

We are all learning together

These tools are new to all of us, and the best ideas come from questions from anyone. If you are already familiar with some of this material, think about how your experience was learning it, and how you might teach it to others. Use these workshop materials not only as a reference in the future but also for talking points so you can communicate the importance of these tools to your communities. A big part of this Series is not only for you to learn these skills, but for you to also teach others and increase the value and practice of open data science in science as a whole.

Vulnerability: yes! Shame: no.

Shame is not allowed here. No "I'm 34 and haven't learned GitHub, it's too late for me" or any of that. We have never had the opportunity to learn these things, there should be no shame on your part for that. It takes a lot of time and dedicated effort to learn and employ these practices, and they should be valued and taught. That's why you're here now, you should be proud that you are taking the initiative and your time to do this. No shame.

Vulnerability, however, will be involved in this Series. Vulnerability is a big part of learning and trying new things — this is a safe place for everyone to learn. Vulnerability is taking stock of where you are now and help you map out where you want to be. Being vulnerable is scary. But it shouldn't be lonely: we all have data confessions that would love to talk about and get help with, if only our scientific culture said that was OK; if only we knew how to articulate our questions and have someone to ask. This is a place to share our vulnerabilities to ignite real change. Ask questions. Whether it's a keyboard shortcut or philosophy of data workflows, ask and let's talk about it.

Everyone is welcome here

You are all welcome here, please be respectful of one another. We are setting a tone of mutual respect and a space place for learning where we assume good intentions and interact with kindness and empathy. Pass it on.

What's possible with open data science (demo)

- R for automation, visualizations
- github for collaborating (code, text)
- bookdown
- websites
- github for project management
    - organize by *project*, i.e., keep that code and those methods in same parent folder, rather than all the R code you've ever written being in a giant folder, spanning projects
    - public & private issues, tagging people on commits, kaban board

Live: fix a tpyo and republish the book/page

What we'll learn

Expect that there is a better way

Seeing what's possible opens up what you expect. There is a bit of a chicken and egg issue here: you need to be exposed to things so you know what's possible and what skills to develop, but you need to kind of know what to look for so you can absorb what you are exposed to.

Have agency to find it

Break down that "I teach you learn" model. We are all here to learn and improve. Learning horizontally.

This series is not about micro-managing your science but about providing guidance & structure so that everyone in the lab is not silently struggling to reinvent the wheel and coming up with weird homegrown data approaches.

What skills you should have and what you should be thinking of, along with some of the tools you can use. Will be building out the Resources page on the website for this purpose. And search the blogs.

Have community to learn with

No more silently struggling & reinventing the wheel & creating weird, homegrown workarounds.

Embrace emerging and established community best practices

Identify what skills and tools you need, map next steps & learn

We will introduce concepts, tools, and workflows and start creating a shared culture around them. That can mean knowing what skills or tools to learn or how to help someone else.

Deliverables

Also known as outcomes you can communicate with your supervisors.

- A more open culture in your group, in part via:
  - regular, dedicated group meetings to skill-share, discuss workflows, & build psychological safety (**"seaside chats"**, Lowndes et al. 2019, 2024)
  - beginnings of a group **pathway** (roadmap) of how you work and next steps for shared workflows
- Be agents of change for open science in your groups, departments, communities, and part of a networked global movement for kinder, open practices and culture change

What would you do in a Seaside Chat?

Example topics:

- Let's organize folders and have READMEs so we know what things
- Where to put data – here's how our server works
- Let's ask how each other are doing and help identify and address common challenges
- Set up Zotero with RMarkdown
- Filepath woes: R users use .Rprojects and `here` package
- Let's screenshare using Chat/Slack/etc and have a shared plan for how we communicate with each other and get help.
- Let's plan a group "hackathon" to move these .xls to .csv files we store on Github

Example lessons from the Ocean Health Index team's Seaside Chats (Lowndes et al. 2017):

- Introduction to GitHub
- Text analysis in R
- Spatial analysis in R
- Free websites with RMarkdown
- Animated plots in R with ggplot2 & gganimate

Coworking

Attending Cohort Calls is not enough, to apply what you learn to your work, build habits, and troubleshoot, you will need to dedicate additional time. At a minimum, we recommend 1.5 hours each week to review what you learned, develop your Pathway document, and try things on your own work. This time is designed to be done during team "Seaside chats", weekly meetings to discuss data workflows and establish shared practices, or Coworking, where Openscapes Mentors will be there to help answer questions and demo additional examples.

Reflection is important for skill-building; come prepared to reflect and share in the following Cohort Call!

Mindset

The Openscapes mindset is about better science for future us. It is about moving away from lonely and demoralizing individual science, and towards science that is more empowering, efficient, open, collaborative, inclusive and kind. It is about many reinforcing ideas that open data science tooling & communities exist that are powerful and empowering, and that we can harness this power broadly to create the culture we want to be a part of.

> Slides
>
> Openscapes mindset
> Openscapes Mindset - NASA Champions

Reframe data analysis & stewardship

Reframe data analysis & stewardship as a collaborative effort rather than an individual burden or personal craft

This is the mindset where you're not alone, and it's not too late. It's not that everyone else learned this and you somehow missed out. You don't need to feel shame; you were likely not supported to learn. This is about releasing those feelings and becoming a confident learner and contributor, no matter where you are starting from.

Redefine collaborators & community

Redefine collaborators & community – beyond your own discipline/institution

Redefine collaborators to include folks that don't work on the same projects or deliverables than you - this starts with other folks in your research group. And it extends to strangers on the internet doing wildly different things than you are but united by the need to manage, analyze, and communicate around data.

Redefining collaborators and community opens opportunities for new perspectives, tools, methods, ideas, colleagues, allies, friends, jobs.

"Future Us" builds from the idea of "Future You/Future Self" being your most important collaborator (see Wilson et al 2017) but with a team science mindset promoting diversity, equity, inclusion, and belonging. How can you work today so that someone other than yourself could reproduce/contribute tomorrow? We describe this more in Lowndes et al. 2019. This mindset is really critical for progress: how you can work now so that you can succeed later (whether that's this afternoon or 4 years from now, whether it's for you or someone you don't know who will pick up where you left off).

Reimagine challenges

Reimagine challenges – expect there is a better way

Reimagining challenges is based on the idea that there is a better way that already exists — and that you can find and make us of it. It is the idea of reusing rather than reinventing. It is a release from lonely struggles that also reduces the cognitive load associated with starting from scratch. This means no more silently struggling, reducing the times you reinvent the wheel, and reducing the times you create weird, homegrown workarounds.

Expecting there's a better way will help you focus on asking for help sooner, to find what you need faster, iterate with confidence & agency.

Open as a daily practice

Open as a daily practice - efficient, collaborative, inclusive, and kind

Open science is not only about products (papers, data) - it is about how you show up and work every day.

This means sharing imperfect work, having empathy, learning & teaching together. It requires trust and time: it means slowing down now to speed up together.

Open data science

We define open data science as **the tooling and people enabling reproducible, transparent, and inclusive practices for data-intensive science.**

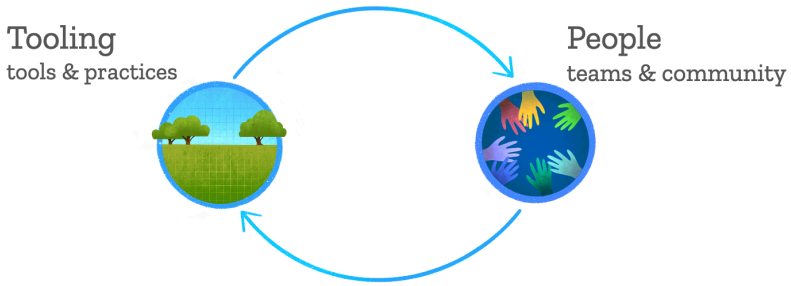


Figure 3: Open data science as a beautiful feedback loop.

This works as a beautiful feedback loop: Using similar tooling promotes and streamlines teamwork. And teamwork better equips you to learn new tooling. With this shared mindset, the idea of team becomes ever-broadening, along with your abilities to adapt to an evolving softwarescape.

Open data science feedback loop in action

There are innumerable examples of this feedback loop in action.

One great example is GitHub.

Tooling: As a tool, GitHub can be used for collaborative version control that handles archiving, bookkeeping, and searching. It also has tools for project management, as well as publishing with Markdown.

People: Community norms with GitHub include the way files are organized, how documentation looks, and the idea that READMEs displayed on the homepage are used to help onboard new folks to using or contributing to the project. Additionally, folks use the GitHub Issues and Projects feature to discuss and coordinate. Used these ways, GitHub is enabling a new era of #SciComm (Science Communication).

GitHub enables collaboration by reducing both `analysis_final_final_final.R` and `re: fwd: FWD: analysis_final_final_final.R`

Open data science key concepts

Data science as a discipline

Data science is "the discipline of turning raw data into understanding" - R for Data Science (Wickham, Çetinkaya-Rundel & Grolemund 2023).

There are concepts, theory, and tools for thinking about and working with data. Just like a field chemistry has concepts for things, theory for how they work, and tools for studying them, so does data science — for data.

No matter what your study system or your question, there are discernible steps involved that when you embrace will make you more efficient. These steps are illustrated in this figure from Wickham & Grolemund: you will need to import/get your data into analytical software, wrangle it (tidy and transform), and THEN you can start understanding your data and asking your science questions by making sense of it visually and with models.

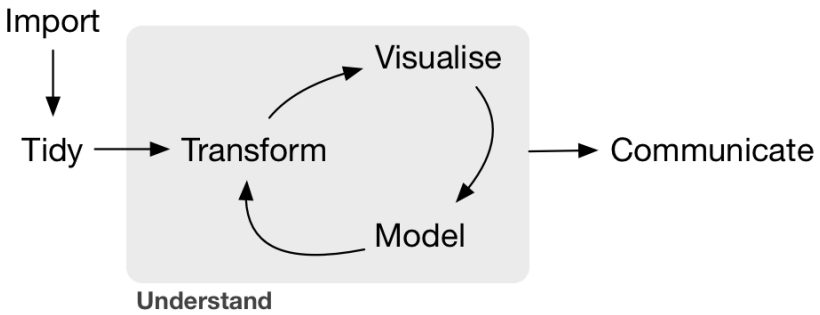


Figure 4: Image from R for Data Science by Wickham & Grolemund

Tidying your data before asking your science questions is a critical point here. Don't build your whole analysis around whatever weird format your data may have come in. That will make it very difficult to later decide you want to examine only a small subset of years, locations, etc. Remember that Data tidying ("data wrangling", "data prep") can take up to 50–80% of a data scientist's time (Lohr 2014, New York Times).

> **Mindset: Decouple your research questions from data questions, get help sooner**

Data science also has an emphasis on communication. It is incredible what is possible on the communication front. This this one-minute video called What is RMarkdown? shows some of the possibilities.

Data science is not just for "big data" or AI or machine learning. You can use data science theory and tools no matter the size or context of your data.

Another important reason to think about data science as a discipline is that it helps dispell the idea that your study system is somehow unique - this is not the case when it comes to data. Distinguish data questions from research questions, learn how to ask for help. Don't confound them or it will be really hard to iterate (for example, when you want to explore a subset like a specific year or place).

Expect there is a way to do what you want to do. This will help you find commonalities and unite you with other members of your research group, and beyond.

> "When I was doing my PhD, I confounded my science questions with my data questions. I was studying squid, asking questions that no one had ever asked before. But I needed to reformat some dates in MATLAB. I treated this date-formatting task also as something that no one had ever asked before. I was looking in the cephalopod peer-reviewed literature for guidance; I couldn't step back and decouple this data question from my squid question, and I struggled for much, much longer because of it." - Julie Lowndes

Open data science tools exist

Open data science tools exist - tools to match data science theory.

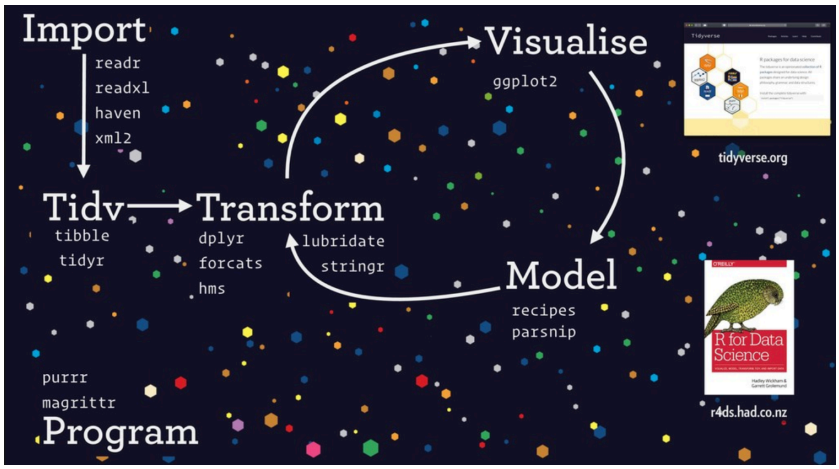


Figure 5: From Wickham 2017, Welcome to the tidyverse

As Hadley Wickham described in his slides Welcome to the tidyverse, tooling exists for every step of this conceptual diagram. They exist to streamline working with data.

These tools are developed by actual people – nice people! We'll explore them more in the Open Communities lesson.

Open as a way to work

Open science is a daily benefit: it is a way to be efficient & streamlined. It is not just an added ask to share data at publication. It's not only about sharing data. It's about how you work, who you include, and the tools that you use. It starts with reuse: when you leverage existing resources, you learn far more than you came for about good practices and approaches,

and spend less time reinventing. It starts off with yourself: be open with yourself and your team first.

Open means easier onboarding – personal and collective. A focus on inclusive documentation and operations means that it is easier to return to a project, whether it's picking up where you left off last month or when you're handing it off to another person. It's like an external memory (personal and collective).

Open requires trust and safety – starting with your team. Psychological safety is a big part of open.

Open is a spectrum – what you share, who you share it with, or how you share it. It's not all-or-nothing.

- What: slides, tweets, blogs, forums, wikis… then also code, data, protocols
- Who: your self, research group, project team, institution…then also public
- How: internal servers, DropBox…then also Google Drive, GitHub, data repos

> Mindset: Increasingly weave open into your work so it's easier to share as you need

Open helps you find solutions faster as you learn to talk about your data, express your needs, and recognize solutions.

Open helps you build confidence – skills are transferable beyond your science. Working openly requires being empathic and inclusive – this will grow a network of allies.

Research group as a team

Focus on common parts that unite you, not what sets you apart.

Build horizontal leadership & resilience so that there is less reinventing, less knowledge lost, less loneliness. Think of the lab horizontally as skillsets & needs instead of vertically as science bins.

It's about diversity, equity, inclusion, and belonging: Instead of the skills you have when you come to the group determining how you will be able to Do Science, co-create shared practices and deliberate paths to onboard new people, and skill-share to continue learning together.

> **Mindset: What do our workflows have in common and how can we learn from each other**

Science is collaborative. Let's make it more head up, heart open and less heads down, elbows out.

Collaborators & community (redefined) as a way to learn

Learn from, with, and for others (Lowndes 2019). This helps overcome isolation, self-taught bad practices, apprehension (Stevens et al. 2018). It helps you transform from "self-taught" to "community-taught".

> "Everyone in the community, no matter how accomplished they are in their own specialty, comes with a **mind open to learning from, with, and for each other.**
>
> We learn from others by using their code and documentation, their blogs and tutorials, their talks and webinars. We learn with others online through Twitter, discussion forums and Slack channels, and in person through coding clubs, hacky hours, meetups, workshops and conferences. And we learn for others by writing tutorials and blogs about our own learning processes, or adding or fulfilling feature requests, which we can then contribute back to the community." - Lowndes 2019, Open Software Means Kinder Science

Use the Internet more for science (it is such an underleveraged tool!). Learn and connect through social media, blogs, forums, new communications outlets…(It is such a cool time to be a scientist).

> **Mindset: Engage as feels right but know you're not alone**

Open science as part of the climate movement

> **" 'We' speaks to the collective, to collaboration, to community, to the relational work at hand. Addressing the climate crisis...will take everyone.**
>
> **'We' speaks to justice, to how we do the work that needs doing and whose contributions are valued. We cannot, we must not, go it alone"**
>
> - Ayana Elizabeth Johnson & Katharine Wilkinson, All We Can Save

Working towards solutions to some of our world's greatest challenges means teamwork and progress at unprecedented levels in ways that have never been seen before. We are part of shaping this. We can use this motivation to ask difficult questions, have difficult conversations, and be leaders, no matter our job titles.

Recap

Open data science tools, practices, & communities exist and are powerful and empowering, and game-changing for science. They enable us to do better science in less time. They are like the Force from Star Wars:

- More powerful than you ever imagined
- Enable you to broaden the scope of the questions you can ask
- You can be a Jedi to others: pass forward what you've learned
- You can join & build diverse communities of allies

We can harness this power for science more broadly. We can create the culture that we want to be a part of – towards kinder science. We can do this with the Openscapes mindset:

- Be efficient, open, collaborative, inclusive, and kind
- Reframe data analysis as a collaborative effort rather than an individual burden or personal craft
- Redefine collaborators & community: Future You, Future Us
  - Think like a team: share imperfect work and learn together
  - Beyond your own discipline, and online
- Reimagine challenges: Expect there is a better way
  - Iterate with confidence, agency, & community
  - You're not alone, it's not too late

Our approach to help develop this mindset is by focusing on the following:

- Data science as a discipline
- Open as a way to work
- Group members as a team
- Collaborators and community (redefined) as a way to learn
- The Internet as an underleveraged tool for science

Notes and resources

- Biased by default: exploring discrimination in research code - Abby Cabunoc Mayes Bioinformatics Community Conference Keynote 2020
- Carpe Talk — Bryan & Averick. Consider for your next talk!

Better Science

Better science is less time is science that is more efficient, reproducible, open, inclusive, and kind. There are growing examples of better science in less time in environmental and Earth science, and beyond, including from the Ocean Health Index team: Our path to better science in less time using open data science tools (Lowndes et al. 2017).

Slides

Better science for future us

Video stories

Better science for future us (22 min). Julie Lowndes shares part of her path to open science with the Ocean Health Index and Openscapes Champions success stories from 3 NOAA Fisheries teams 1) at the end of their cohort; 2) 2 months after their cohort; 3) several years after their cohort.
First Forays into the Cloud (19 min), by Aronne Merrelli to the 2024 Cohort of NASA Openscapes Champions. Aronne is Assistant Research Scientist at the University of Michigan College of Engineering (Climate and Space Sciences and Engineering) and a 2023 NASA Openscapes Champion.
My path to open science (7 min), by Stefanie Butland, Openscapes Team Member

Additional slides

These slides were contributed by guest instructors.
Empowering transformational science - Dr. Chelle Gentemann

Here we also introduce the **Pathways concept** that teams will develop throughout the Champions program. The Pathway is based on Table 1 in Lowndes et al. 2017, and helps teams deliberately identify data workflow practices and next steps to facilitate efficiency and open practices in terms of reproduciblity, collaboration, communication, and culture.

Pathways to better science in less time

Figure 1 of Lowndes et al. 2017 shows that open data science tools increased the ease of reproducibility and the ease of collaboration for the Ocean Health Index (OHI) team. But it was not the tools alone - it was the process the team co-created and prioritized.

Create space

The OHI team created space for synchronous collaboration: convenings to learn and teach each other together.

A critical first part of this was prioritizing time (which included getting buy-in, lobbying, showcasing). Then, this meant that the team could focus time on:

- Getting comfortable talking about data/workflows
- Building trust (to share imperfect work)
- Recognizing that what we invest incrementally will have large dividends in the future

The OHI team started having "Seaside Chats": 1x/week where they discussed filenaming, code review, standard operating procedures and documentation, and much more.

Create place

Creating place is critical for asynchronous collaboration. This includes GitHub Organizations, Repositories, Issues; Google Drive Folders, Docs, Spreadsheets, and Slides; Slack Organizations and Channels; JupyterHubs, etc.

It is a place for code, shared practices, resources, conversations. Critically, this involves making sure that everyone on the team is comfortable contributing through these channels. This means both with the technology, and the culture of the team.

Find the common

Through creating space and place, teams will find the common workflows, tools, skills that they already have and need to do their work.

For the OHI team, we asked how to make sure everyone can participate as they need to? We introduced new software sparingly, and helped each other learn. This included initial setup as well as follow up and practice. We leveraged existing habits & resources - within and beyond our team. Open communities were a big part of this learning. Through this we were able to distinguish data preparation (tidying) as distinct from our science, and make this actionable by shifting to smaller modular code to combine for different reports/audiences.

Documentation was a key part of this. And, writing documentation "for nobody" is very hard, and it's a huge task. We prioritized documentation based on Onboarding and Offboarding: for our future selves first, and then future us.

What was it really like? Transition to GitHub

Changing behavior/habits takes time, and is messy. Here is part of the story:

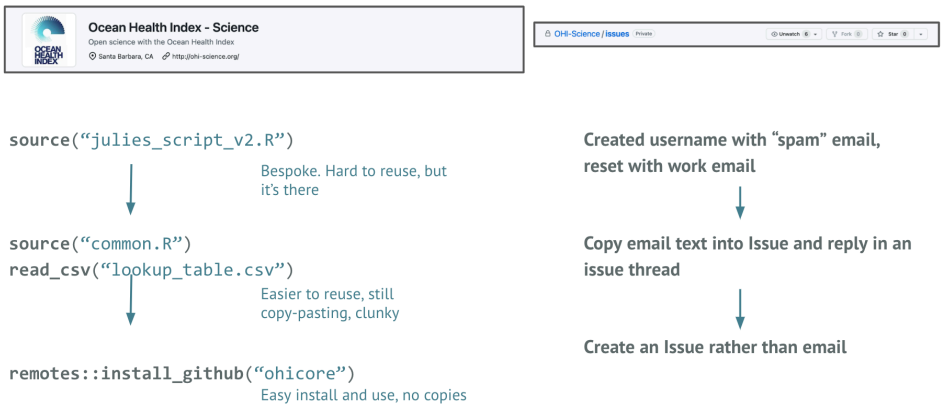


Figure 6: Ocean Health Index (OHI) team's transition of code and shifting communication to GitHub

Ocean Health Index: behind the scenes

Some key points to discuss from Our path to better science in less time using open data science tools (Lowndes et al. 2017):

- Reproducibility & communication enabled by open tooling
- Shared practices are useful beyond shared projects

If you're interested in more overview of the OHI setup, see this 2017 talk (25 mins): OHI Better science in less time or this 2021 Plenary at Better Science for Future Us at the inaugural Society for Open, Reliable, and Transparent Ecology and Evolutionary Biology SORTEE conference (30 minutes) (video).

OHI pathway

- Motivated by necessity
- Reimagined by possibility and community
- Done incrementally!
- Yes: it's an investment.
- Also yes: huge, enduring payoff for (your) science

Reproducibility & communication enabled by open tooling

RMarkdown/Quarto to reimagine data analysis and communication. RMarkdown/Quarto combines analyses & figures together, rendered to your reporting output of choice.

An example: http://ohi-science.org/bettersciencein lesstime/

- Website built with R/RMarkdown & Github
- You can get started too: 1-hour RMarkdown tutorial; Quarto website tutorial

Shared workflows not only useful for shared projects

- OHI team: we identified as a team & prioritized helping each other
  - We work on many different projects
  - Use same workflows, share feedback, can think together across projects
- Shared conventions reduce friction & cognitive load
  - Common ground, easier to talk about, easier to ask for help
  - You don't need to design everything from scratch

And, critically:

- It's about increasing efficiency and reproducibility and open science.
- But it is also about increasing participation and inclusion.
- Consider diversity, equity, and inclusion in your daily practices.
- How you work and onboard others to your projects is a DEI issue.

Examples: environmental science

Here are a few examples to showcase what is possible and being done by the community.

- Regime Shifts in R & Data Science within the BC Public Service Observations from the field - Stephanie Hazlitt, Government of British Columbia, slides from CascadiaRconf keynote
- NMFSReports: Easily write NOAA reports and tech memos in R Markdown! - Emily Markowitz, NOAA Alaska Fisheries Science Center, slides from CascadiaRconf talk
- Automated reporting in Tampa Bay with open science - Marcus Beck, Tampa Bay Estuary Program, Openscapes blog
  - TBEP's Data Management Workflow and open science cake

Further resources

Not so standard deviation podcast

Parker & Peng, http://nssdeviations.com. Great discussions about data concepts and "in the wild". Start with Episode 9: Spreadsheet drama

Practical computing for biologists

Haddock & Dunn, http://practicalcomputing.org. Software & computing concepts already on your computer. Start with Chapter 2: Regular expressions

GitHub Strategies

What is GitHub[1], and what are GitHub strategies for Future Us?

GitHub is a powerful tool for collaborative coding with version control, but here and in our GitHub Clinic we are going to focus on some of its lesser-celebrated awesomeness. We'll talk about GitHub for supporting, reusing, contributing, and failing safely, as well as GitHub for publication and project management. We will focus on how to use GitHub for collaboration and communication for science, and spend time with hands-on practice.

GitHub Clinic

Openscapes GitHub Clinic introduces GitHub with the motivation of collaboration and open science. It is designed for both new and seasoned learners, focused on how to develop workflows with diverse teams.

Slides

GitHub Clinic

Video recordings

Part 1: Publishing, recorded as 19 minutes of lesson, cut for participants hands-on in breakout rooms, and 5 minutes of followup lesson.
Part 2: Project Management, recorded as 9 minutes of lesson, cut for participants hands-on in breakout rooms, and 5 minutes of followup lesson.
Part 3: 2i2c JupyterHub, introduces a beginning GitHub workflow from our 2i2c JupyterHub (part of NASA-Openscapes). It is recorded as 9 minutes of lesson.

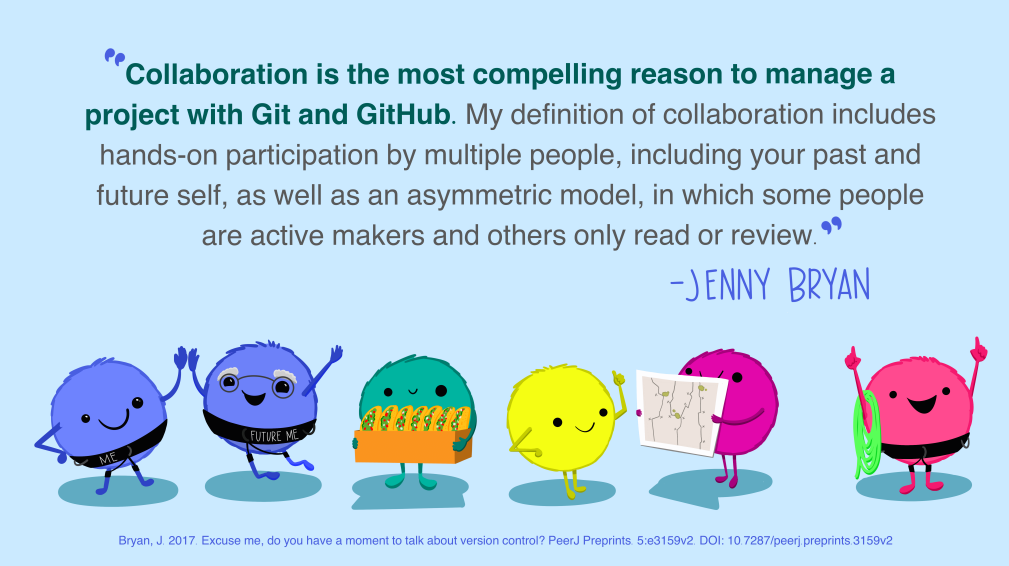
[1] From Bryan 2017: "we're targeting GitHub - not Bitbucket or GitLab - for the sake of specificity. However, all the big-picture principles and even some mechanics will carry over to these alternative hosting platforms. We are advocating for the use of hosted version control as a general concept, with GitHub being the best and most common provider today."

GitHub for supporting, reusing, contributing, and failing safely

This introduction was developed with Allison Horst in the **GitHub Illustrated Series** (Horst & Lowndes 2022):

When we talk about managing projects to help us contribute, reuse, collaborate, and fail safely, we are talking about GitHub. We mean using modern collaborative software designed to organize otherwise unwieldy interrelated files and to track changes by potentially different people through time – something that is for everyone, whether or not we identify as "coders". People do use GitHub for code- and data-intensive projects, but we are not limited to that use. GitHub is a powerful approach for any project that we want to organize and communicate about; it's a way we can be open with ourselves and our teams, and also work more publicly as we feel comfortable.

So let's talk about why GitHub is so empowering for our project-oriented and collaborative lives.



See the **GitHub Illustrated Series** (Horst & Lowndes 2022) for the full series!

Growth Mindset

When we teach GitHub, we find it's useful to say explicitly that we should go in with a Growth Mindset - the idea that we don't know something ***yet*** but we are able to learn. Growth Mindset is work by Dr. Carol Dweck - see her book, TED Talk (10 min); illustrated video (2 min).

As part of this, we like how Jenny Bryan (Code Smells and Feels) thinks about this - that your tastes develops faster than your abilities.

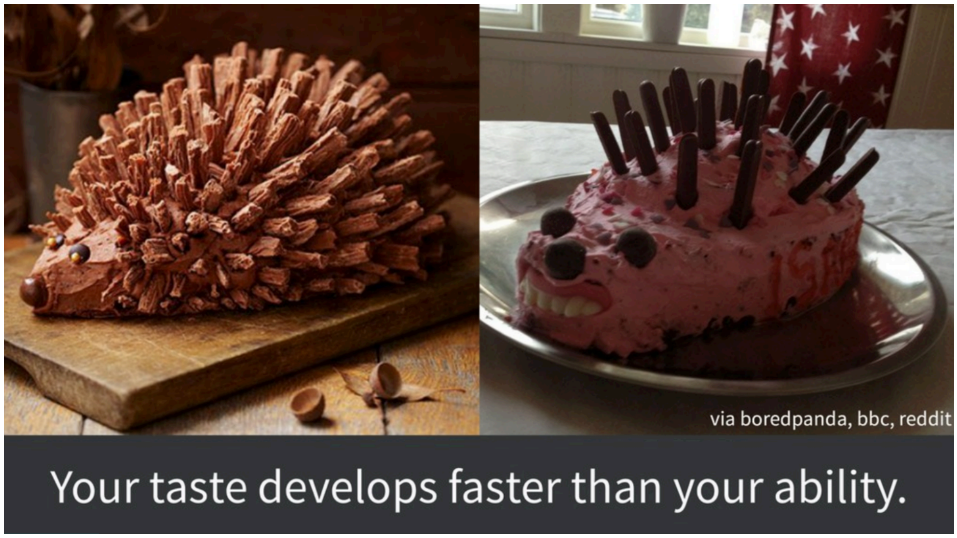


Figure 7: Slide from Jenny Bryan (Code Smells and Feels)

GitHub for research groups

As you get a better hands-on sense of GitHub's capabilities, you'll be thinking about how to get organized and use this for your own research group.

Organizations

Here are examples of GitHub Organizations created as part of Openscapes Champions Cohorts where the content is now developed and maintained to support a broader range of researchers.

- NASA Openscapes https://github.com/NASA-Openscapes
- NMFS Openscapes for NOAA Fisheries https://github.com/nmfs-openscapes
- California Water Board https://github.com/CAWaterBoardDataCenter

Many Champions teams create GitHub Organizations for their research group. This is a way for all the work that happens in the research group to be organized in one place, but also clearly attributed and credited by each user who contributes.

You can explore other Champions teams' GitHub organizations they've created. They're listed under the Cohorts in which they participated:

NOAA NEFSC

- Gavin Fay Lab https://github.com/thefaylab

2021 NOAA NMFS Cohort

- NWFSC Fisheries Engineering and Acoustic Technologies (FEAT) team https://github.com/NOAA-FEAT
- NWFSC Protected Salmonids Team https://github.com/nwfsc-math-bio
- AFSC GAP Survey Data Products https://github.com/afsc-gap-products

CS&S Cohort blog post; Cohort repo

- Kenai Watershed Forum https://github.com/Kenai-Watershed-Forum
- WildCo Lab https://github.com/WildCoLab

CSU-COAST Cohort blog post; Cohort repo

- Logan Lab https://github.com/loganlabcsumb
- Ecological Oceanography Lab at CSUN https://github.com/ecoocelab-csun
- Claisse Lab @ Cal Poly Pomona https://github.com/ClaisseLab
- Coastal Ecosystems Lab https://github.com/coastal-ecosystems-lab

Should my students create repos in our group organization?

While there are many different approaches that could make sense for your group, our starting recommendation would be that any project happening in the research group gets its own repository, and that repository is in the research group's Organization. This is then clearly part of the research group's work and more findable by past/current/future members. And it can be forked/moved to other accounts at any time.

Issues

Issues and Projects are a great way to keep organized. See a few examples in the NOAA Fisheries Resource Book and Openscapes How We Work issues and Planning project.

How much should I write in a single Issue?

The "size" of each Issue is something you'll get a feel for as you use them more, so don't worry too much about it as you get started. The most common "size" of Issue is something you can complete, and close. This might include smaller checklists and conversations, but it is more along the lines of "change axes labels in my plot" rather than "write thesis chapter".

Some Issues are also used more as reference and shared resources, and might not be closed so they're easier to discover. Depending on your workflow and norms you establish with your group, it might be best to keep those reference issues in a different repository so you don't have lingering open issues in a repo that you like to close issues and move on.

When you're logged into GitHub, go to https://github.com/issues to see all of the Issues assigned/created/etc for your account!

Branches

Branches are a really powerful feature for software development, but aren't necessarily the right place for new GitHub users to get started. If there are a small number of people in a single repository and you're all first learning, we'd suggest that you start using GitHub without branches, and focus on checking in with each other as you contribute (see R for Excel Users example collaborating with GitHub from RStudio).

> Aside: Julie Lowndes only worked in the main branch with her small group of Ocean Health Index colleagues for the first 8 years as a GitHub user; it was only when working with the NASA Openscapes Mentors community that she learned how to use branches as software developers do.

Branches are powerful as your teams grow and/or as you become savvier with GitHub and code, so it is good to discuss strategies with them.

Should we do everything in a `dev` branch?

Full question: "Right now any collaborative code in my group has a main branch and a dev branch. All collaborators clone the dev branch and push back to the dev branch on GitHub before anything ever goes to main. Is this best practice?"

Advice: This system with the dev branch works so long as the dev doesn't diverge too far from the main branch - the Ocean Health Index project does it that way but it's someone's job for a few weeks at the end of each year making sure the dev branch can merge into main.

Another approach is to treat each branch as very temporary: you make a branch from main, add your contribution, make a pull request, merge and delete the branch. This is what we've been doing with the NASA Mentors and it's more continuous little updates rather than larger big updates, and I think it works better when working with Jupyter Notebooks, which can cause a lot of merge conflicts.

GitHub for publishing

GitHub is known as a place to store code, but it's also a powerful publishing system. It is a way to help you share about your project on the open web, which lets you share about your science earlier.

Preamble

We are going to work with GitHub from the browser only, because it makes the best use of our short time together. It is also a powerful way for folks to contribute and collaborate even if they are not involved in day-to-day hands-on analysis. So this might be good for new lab members or students to contribute to your lab as soon as possible.

GitHub can reduce friction for open science: it gives us avenues for communicating and publishing methods, blogs, interactive graphics and more, without a lot of heavy lifting!

Prerequisite

You will need to create **GitHub** account at http://github.com, if you don't already have one. Optional advice from Jenny Bryan about choosing a username.

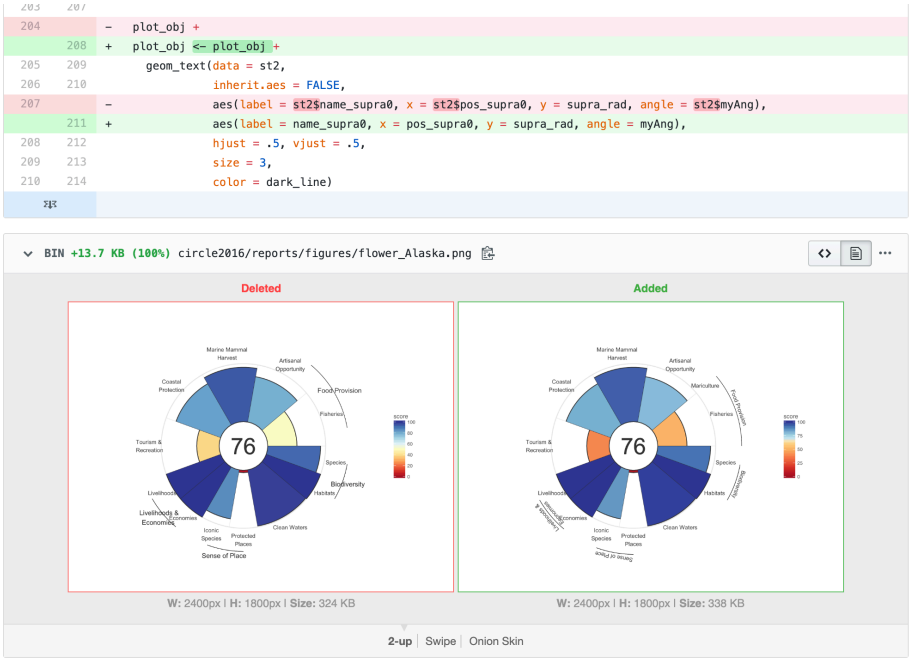
What is GitHub? — Traditional answer

GitHub means GitHub.com; it's a company that is an online collaborative platform, with some features familiar to social media users.

GitHub centers around git, which is powerful version control software for your local computer. This has been around for years, taking care of bookkeeping for you locally on your computer.

GitHub makes git's local bookkeeping collaborative through its powerful online platform. It will weave together all the version control from your local computer with other collaborators you work with.

It is used for code and files: organize, archive, bookkeeping, searchable, changes visualized, etc. In the figure below, notice the familiar red and green to denote deletions and additions line-by-line, with darker shading to identify specific text within a line. Also notice the differencing in the image's color bar!



Disclaimer

We aren't going to teach traditional git/GitHub today, but here are some recommendations if you'd like to learn. First, read Jenny Bryan's "Excuse Me, Do You Have a Moment to Talk About Version Control?" (open-access pre-print from PeerJ, published in The American Statistican). This provides an excellent overview. One quote I like in particular is

> Collaboration is the most compelling reason to manage a project with Git and GitHub. My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review. - *Jenny Bryan, "Excuse Me, Do You Have a Moment to Talk About Version Control?"*

When you're ready to learn GitHub with R, the absolute best resource is Jenny Bryan's Happy Git With R. This is a comprehensive, friendly step-by-step process of how to do so, and is an awesome reference for seasoned git/GitHub users as well. If you want a shorter-form resource, I'd recommend 2 tutorials from R for Excel Users. This also teaches you how to set up GitHub to sync directly through RStudio, without any other software (including the command line) required to do so: see chapters on GitHub and Collaborating with GitHub

> On my local computer, I interact with GitHub through RStudio 99.9% of the time (use command line .1% of time). - Julie Lowndes

What is GitHub? — Non-traditional answer

Publishing platform

We can use GitHub to create a URL to share individual or groups of files, or for books like this one (openscapes.github.io/series), websites like openscapes.org, which was built with Quarto (previously RMarkdown), and interactive dashboards.

Project management system

GitHub is also a project management system for short and long-term tasks. It is really powerful to have collaborative "todo" 's in the same software (and user accounts) as all the analysis and all the people that you're already working with.

We will talk about "Issues" & "Projects" in the next chapter.

> The overall effect is that a directory that is a GitHub-synced Git repo can simultaneously be the code-heavy back end of a project and an outward-facing front end. - *Jenny Bryan, "Excuse Me, Do You Have a Moment to Talk About Version Control?"*

GitHub framework in a nutshell

Users vs. organizations

Example: jules32 is a user account, openscapes is an organization group.

You can think of them like other social media accounts: I can be an individual or part of a group, and there are permissions associated with both.

Repositories ("repos")

Repos are GitHub's main unit. They are essentially a folder, and you'll put files and folders in them. They are contained, with permissions specific to each one.

It makes it easier to navigate through and find stuff — so you are "not sifting through a zoo of files" as one Openscapes Champion has said.

"Commits" & "commit messages"

Unlike Dropbox or Google Drive that constantly and automatically sync to the cloud, you have to deliberately tell git/GitHub when you have an amount of work that you want to be versioned and synced. You have to commit to telling them. GitHub takes care of the backend bookkeeping involved, but you have to write a human-readable message to your future self and others. That is the commit message.

There is no absolute guidance for how often to commit, but I think of it as leaving breadcrumbs for yourself. How much work and on what things/in what combination would you like to be able to reverse? What kind of information will make it easier for Future You to work with?

Public vs private

You can have both public (the free default) and private repositories, and change these permissions later on. I mostly work in public repos, but if I work in private ones, I often have the expectation that they will be made public some day. So I practice good habits with commits and documentation, and keep conversations on-topic.

The search feature is awesome

You are able to search within a GitHub repository, across repositories in an organization, or across all GitHub public repositories. I find this helps me find things quickly if I'm looking for how I've used a function in the past, or if I remember a word that would stand out that I included in a commit message as a breadcrumb to myself. It will also search Issues within the repositories, so you can look for specific words in conversations as well.

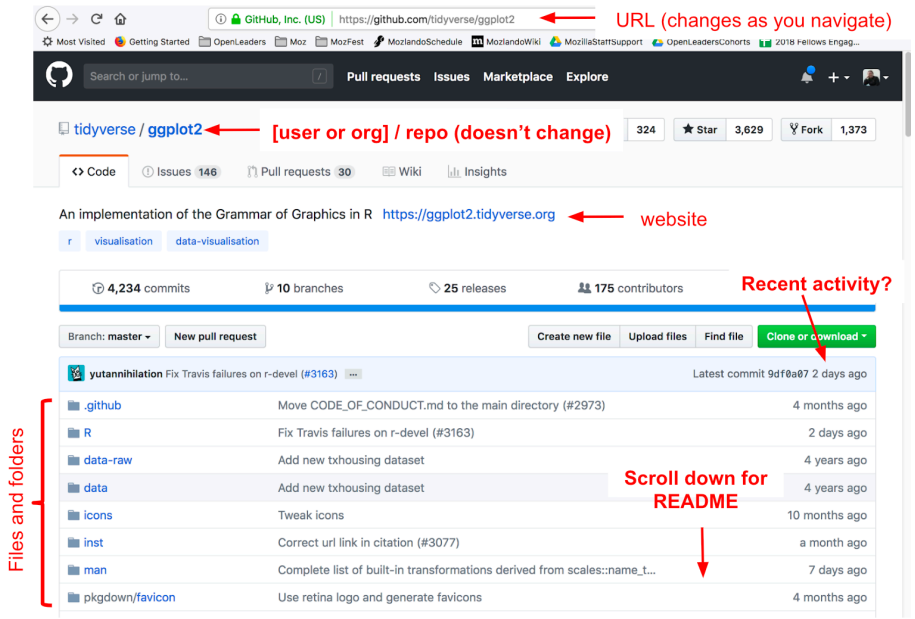
Branches & Forks

We're not going to talk about branches and forks. These are a core feature of what makes GitHub super powerful for software development, and one of the first things you'll see in GitHub tutorials geared towards software engineers. But I do not think that is the most

relevant or smoothest entryway for those of us who are scientists fairly new to collaborative coding and version control.

GitHub Orientation

For a demo of the following, see the GitHub Clinic recordings listed on the GitHub Strategies page.



Editing files from GitHub

First a Disclaimer: you don't want to edit from the browser for most things – you would want to "clone" the repo to your local computer and leverage more goodies & power. However, you will sometimes edit in the browser, and it's a good entry point for us today, and maybe for onboarding folks in your lab in the future.

Why not edit in the browser? You don't want to overwrite each other or forget yourself. Good for quick md editing, not script editing.

In the demo, the example .md was a deliberate example of sharing slides from a talk :)

What to do: (you all have permissions)

- Go to your cohort's repository and find yourname.md (an example repository: https://github.com/openscapes/demo)
- Click on the pencil to edit your file

- Make many edits & commits with commit messages
  - github.com has a default message, but get into the habit of writing an actual message to yourself/others (breadcrumbs)
- This is different from saving (cancel if you save!)

Further resources

- Git for Humans - Alice Bartlett, 2016

GitHub for project management

GitHub is best known as a collaborative coding platform. But of course productive collaboration requires communication, and GitHub has powerful features to support communication and project management through GitHub Issues.

Preamble

We are focusing on GitHub Issues here because they are a powerful way for team members to have active discussions about data and code, and therefore ways to participate in analyses even for those that are not involved in the day-to-day coding.

> I find Issues not only useful to discuss topics as a team, but I also treat it as my external memory: I write notes to myself, link to files and websites; I leave breadcrumbs for myself so that I am more easily able to remember my past thought processes and pick up projects where I left off. - Julie Lowndes

In this way, GitHub Issues help actualize the mindset of **Future You** and **Future Us**. This means being deliberate now about communicating decisions and progress so that you or others can work in the future a little more smoothly.

> Using project management software is a strategy used by every software developer or people working on projects with many moving parts. It streamlines technical discussions with people who are coming/joining a group. It also helps organize and track projects that single or multiple & overlapping users can be a part of. - Jenny Bryan

While there are many options for project management software out there, we use and teach GitHub because it's already managing our code and our work, and linked to our collaborators so it offers a streamlined way to communicate. (It's also one less account to manage, which is a huge bonus in Julie Lowndes' mind).

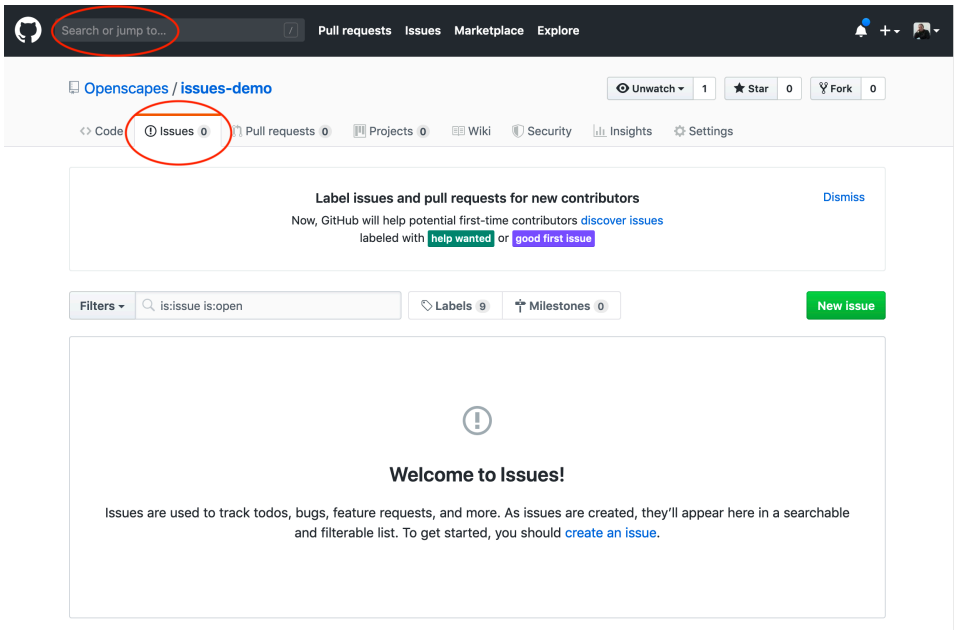
One of the reason we talk about Issues in Openscapes is because they are an excellent way to develop habits for using GitHub for your analytical project more broadly.

What are Issues?

For a demo of the following, see the GitHub Clinic recordings listed on the GitHub Strategies page.

Every GitHub repository (shortened to "repo") has a feature called Issues. Issues is GitHub's project management and task-tracking feature.

> Issues "track ideas, enhancements, tasks, or bugs for work on GitHub." - GitHub



Jenny Bryan has an excellent summary of Issues in her article "Excuse Me, Do You Have a Moment to Talk About Version Control?" (open-access pre-print from PeerJ, published in The American Statistican):

> GitHub issues are another powerful feature of the platform. Recall that we are repurposing Git, a tool that facilitates software development. Think of the issues for a project as its bug tracker. For projects that are not pure software development, we co-opt this machinery to organize our to-do list more generally. The basic unit is an issue and you can interact with one in two ways.
>
> First, issues are integrated into the project's web interface on GitHub, with a rich set of options for linking to project files and incremental changes. Second, issues and their associated comment threads appear in your email, just like regular messages (this can, of course, be configured). The result is that all correspondence about a project comes through your normal channels, but is also tracked inside

the project itself, with excellent navigability and search capabilities. For software, issues are used to track bugs and feature requests. In a data analysis project, you might open an issue to flesh out a specific sub-analysis or to develop a complicated figure. In a course, we use them to manage homework submission, marking, and peer review.

Issues can be assigned to specific people and they can be labelled, e.g. "bug", "simulation- study", or "final-exam". Coupled with the ability to cross-link issues and the project files or file changes, you have extraordinary power to document why things have happened in the past and to organize what needs to happen in the future.

You create an Issue for a topic, and use it track progress or ask questions. You can provide links, describe updates, link to other Issues, and you can close the Issue when it is completed. You can also re-open previously-closed Issues.

Every GitHub repository has this Issues feature. This means that sometimes Issues are public and sometimes they are private.

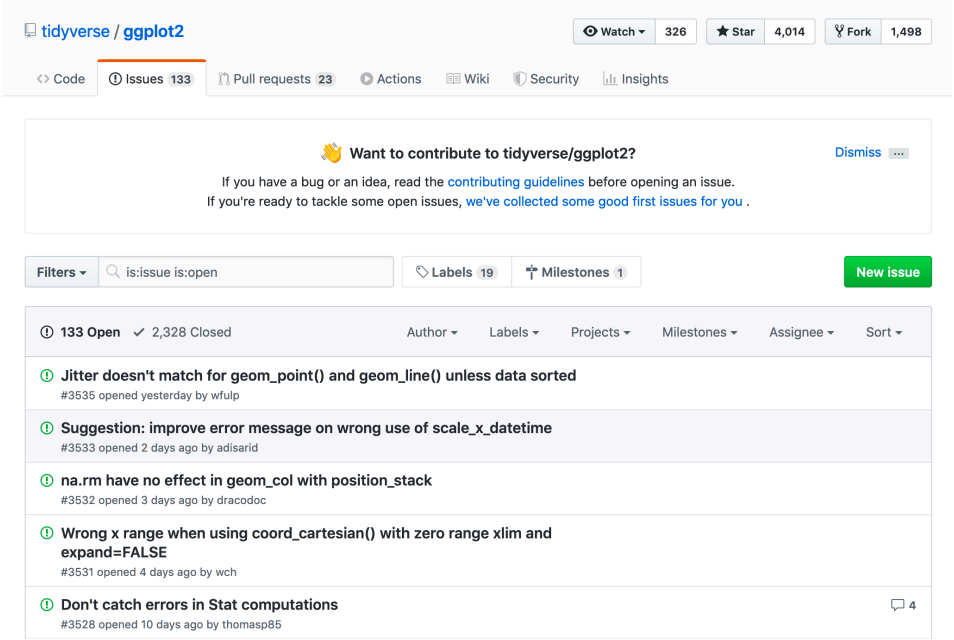
- In a public repo, anyone with a GitHub username can create and comment on issues.
- In a private repo, only users with permission can create and comment on issues, or see them at all

GitHub search is awesome – it will search all of your files and Issues!

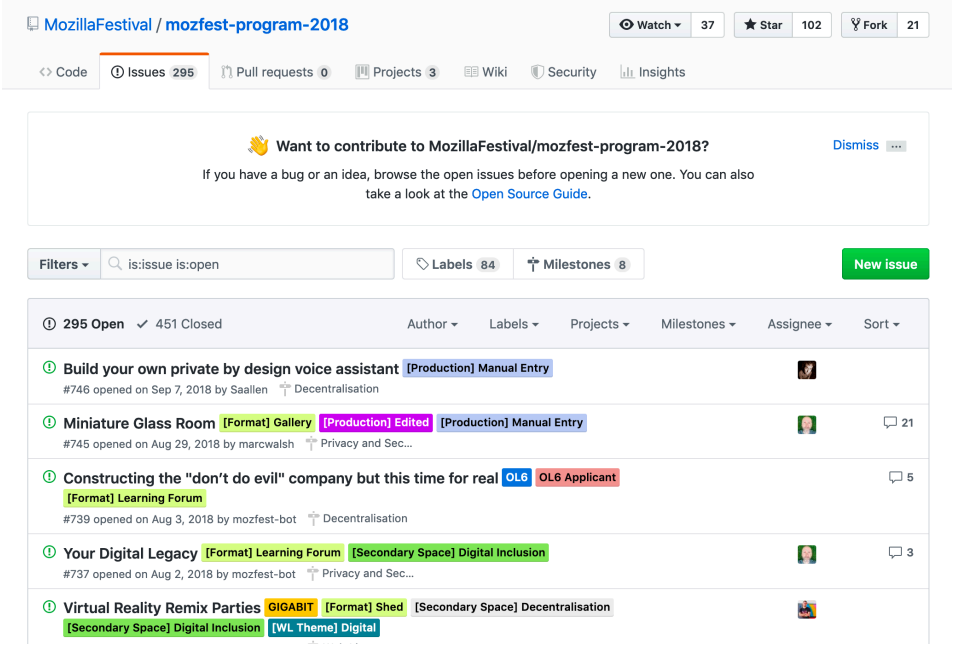
Issues in the Wild

Here are some examples of "traditional" and "non-traditional" use of issues.

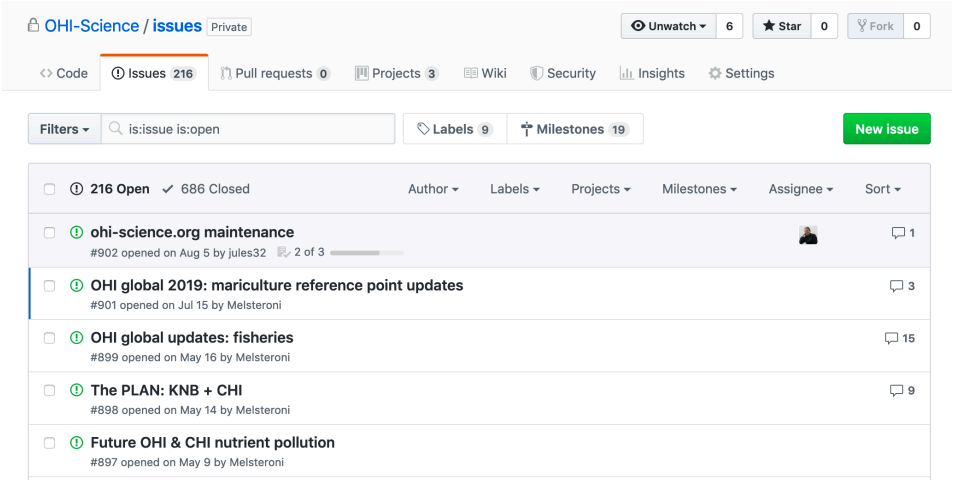
ggplot2's Issues is an example of what I think is the "traditional" use of Issues, which is in a pretty pure software development context. This is a public repository, and all topics are directly related to ggplot2. Issues are largely used to report bugs, troubleshoot and sometimes to request features. Note the "Filters" feature on the top-left: this by default will search through the Issues that are still open, but you can also change this if you wanted to search also for closed Issues (just below "Filters" you can see that there are over 2000 closed Issues, documenting the innovation that's been ongoing in ggplot2).



MozillaFestival's Issues are an example of a less "traditional", but increasingly common use of Issues: for project submissions, coordination, and community engagement. It is also an example of the use of labels: those colorful tags that help group and categorize the Issues. To the right of "Filters", you'll see a "Labels" button: clicking on this will give you a list of all the labels and how many Issues are tagged with each label. There are also options to create new labels.



OHI-Science's Issues: are also an example of less "traditional" use of Issues but perhaps also somewhat common. This is a private repository, which is why there is no link to these Issues. Here, Issues are used for private conversations and archiving ideas and discussions: the OHI-Science team uses issues instead of email to have private, archived, searchable conversations about scientific methods as described in [Lowndes et al. 2017](https://www.nature.com/articles/s41559-017-0160). The team is diligent about having important science conversations in these Issues, rather than those conversations being lost in emails or Slack. This is more organized and also makes onboarding team members much smoother since the team does not need to forward emails to new team members and tease through attachements.

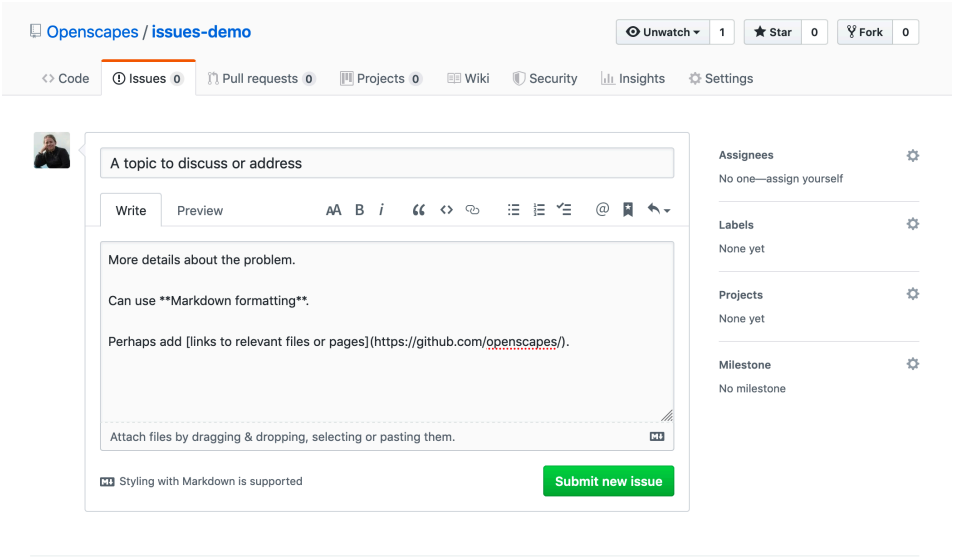


How to use Issues

Creating a new Issue

When you click on the green "New Issue" button", you're asked to give a Title and Leave a Comment. You can also attach files or images by dragging them into the Issue.

Then you'll be able to Submit the Issue. On the right side, you'll see options to Assign someone to the Issue, add a Label, add it to a Project, or add it to a Milestone. We'll explore these features a bit more in a moment.



When you click Submit, your Issue gets a number, which is now written next to the title. This number is also part of the URL as well.

On the right of the Title, note that there is an "Edit" button if you ever need to change the title of the Issue as the conversation evolves. The Issue number will stay the same.

What happens if you want to edit the text of your Comment after you've Submitted it? No problem. See that once you've submitted an Issue, there is a blue bar at the top of the Comment, attributing your username to this comment along with the date. At the very right of this blue bar, there are 3 dots. Clicking here will give you the option to edit.

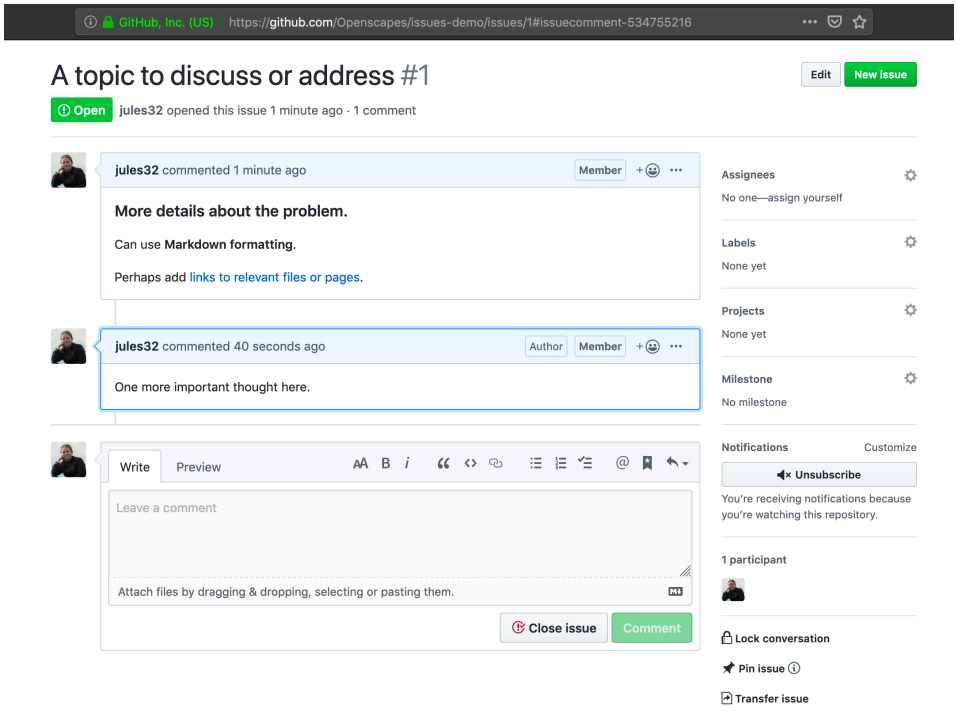
Commenting

The great thing about Issues is that they are for conversations with yourself and others. So once you've submitted an Issue, you can string together additional Comments within this comment. Maybe you asked a question, and someone else will respond with a solution or idea. They might link to an Issue with a related topic, or an external link that might be helpful.

You can also tag people in Issue Comments with the "@" symbol. As noted above, anyone who is part of the repository will automatically get email notifications when comments are submitted. But tagging specific users will also send them an email, and is a good way to bring folks into the conversation who might not already be "watching" the whole repository. In a public repository you can tag any GitHub user, and in a private repository they have to have permission.

Each time there is a comment in the Issue thread, there will be a new date marked in the blue bar at the top of the Issue. This is a nice way to see how current conversations are.

And something really great is that you can click on the date — and watch the URL change. This allows you to anchor to a specific comment within the Issue thread. This is really useful if, for example, you wanted to share a specific comment with someone else instead of having them scroll down themselves. (Note: you can also click on the three dots at the right of the blue Comment header to copy the anchored link).

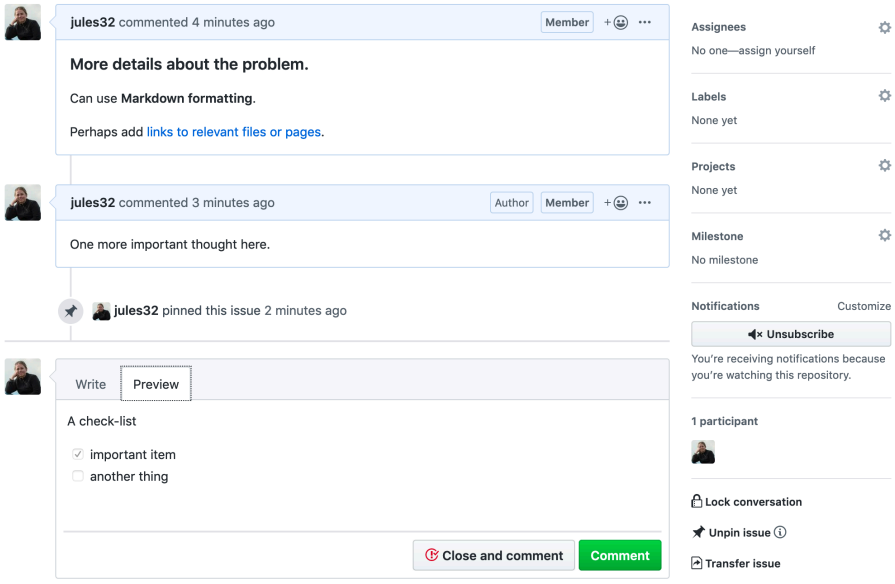


Markdown

Issues support Markdown. This means that you can add simple formatting to your text, such as headers, bold and italics, lists, images, links, and formatted code. To help you use Markdown formatting as you learn, GitHub Issues have built-in help: there are icons between the Title and the Comment of the Issue that will do the Markdown formatting for you, and help you learn along the way.

There is also a "Preview" tab between the Title and Comment (next to the "Write" tab, where you are by default) where you can preview what your Markdown formatting looks like before you Submit the Issue. Submitting the Issue will also render the Markdown formatting.

GitHub enables you to also create Markdown check-lists by typing `- [ ]`. Once this is rendered, you can click it to check this box. Alternatively, in Markdown you check a box by typing `- [x]`. The number of checked and unchecked items will be visible in the Issue as well.

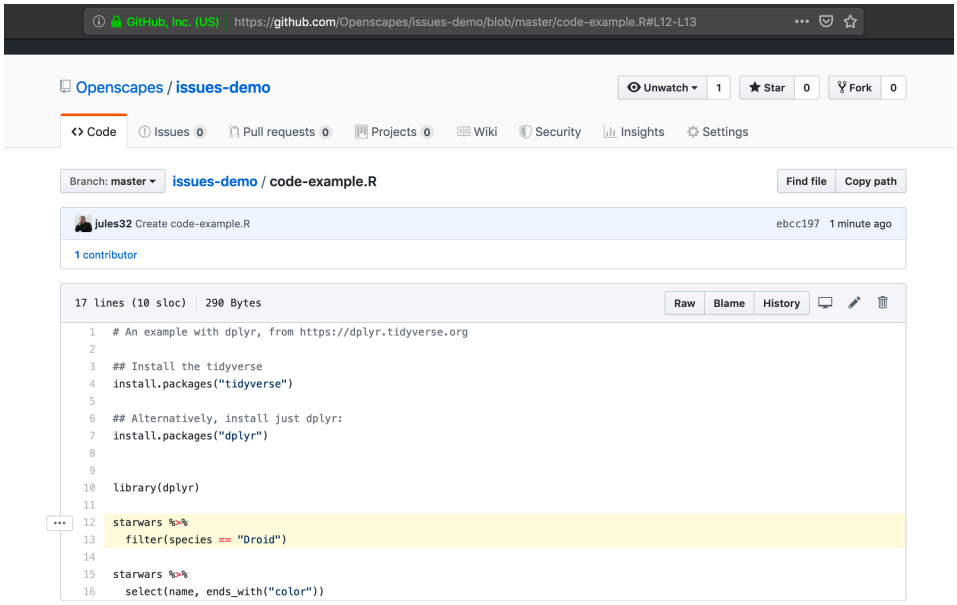


Linking to files

Linking to specific files or versions of files is good practice when you are discussing it in an Issue: reduce the work for the person reading the Issue (which might be Future You!). You can link to the file by opening it in the browser and copying its URL and placing it in Markdown formatting for hyperlinks: `[text to hyperlink](URL)`.

You can also navigate to a specific version of that file, or a specific commit message, if you want to capture that file at a specific point in time.

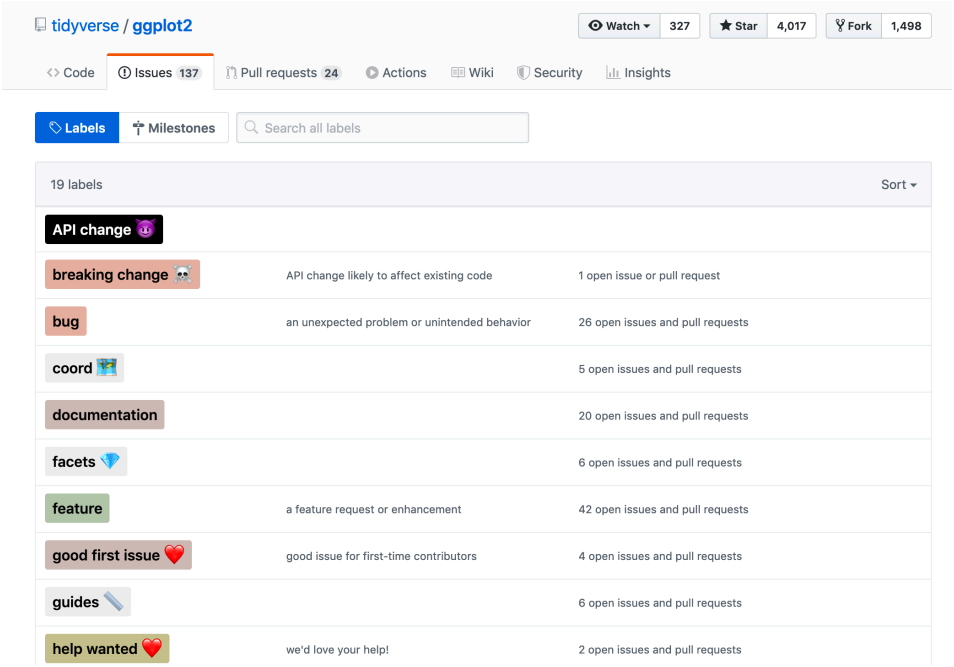
You can also anchor to specific lines within a file, which is useful if you are requesting feedback on a specific part of an analysis or asking for help troubleshooting. I can send someone to a specific place within a file with the appropriate lines highlighted. For example `[important code](https://github.com/Openscapes/issues-demo/blob/master/code-example.R#L12-L13)` will render as important code.



Assigning, Labels

On the right side of the Issue thread, there is the "metadata" for the Issue. You can assign the Issue to a specific user, or label it with a suite of labels that you can customize (when you click on labels, see all the way at the bottom the option to edit labels. And there are other ways to navigate there as well).

If you navigate back to the full list of Issue topics (which will have the URL github.com/username-or-organization/repo-name/issues), you'll see these metadata categories listed at the top as well, which lets you filter or view based on these categories.



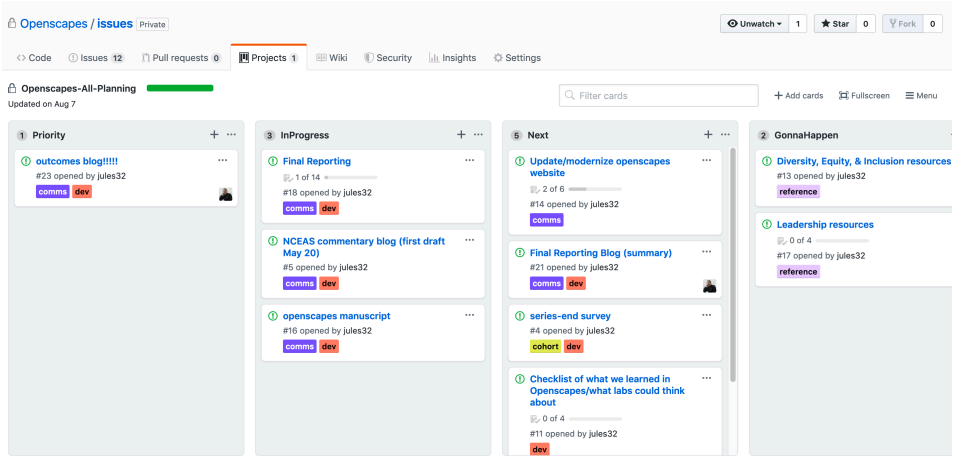
Projects, Milestones

Projects and Milestones are further ways to organize and track progress with your Issues.

Projects are a way to organize and prioritize your issues. It uses the idea of a Kanban board, which Wikipedia says "visually depict work at various stages of a process using cards to represent work items and columns to represent each stage of the process. Cards are moved from left to right to show progress and to help coordinate teams performing the work." The simplest have 3 columns labeled "to do", "doing" and "done".

You can use Projects for both Organization or personal projects. In fact, you can have multiple projects within the same repository, so different people can have different Projects organized within their shared repository, for example.

You have a lot of control over how you will manage your Projects; at this point I do not use all the features but have been playing around with using them for Openscapes planning:



For examples of Projects, see https://nmfs-opensci.github.io/ResourceBook/content/project_management/github_project_boards.html as well as a 2024 Openscapes Community Call: GitHub for NASA Openscapes community calendaring & project management.

Milestones are a way to attach deadlines to your Project (although you do not need to identify a date if you don't want to). Once you create a Milestone, you can add Issues to that Milestone to help track progress. For example, maybe you have a presentation coming up and there are several Issues that need to be addressed before then.

Strategies for Issues

Every repo has Issues, but do you want to use Issues in every repo? It helps to consider the purpose for the Issues.

Using Issues for "traditional" bug/features for code, it makes sense to keep the repository public and have all Issues pertaining to that repo there within that repo.

If you're using Issues for "non-traditional" laboratory research group and science conversations, there are other considerations. Maybe you do want a private repository, but even so you'll want to think ahead. Will you eventually make that repo public when you publish your study? Changing a repo from private to public (or vice versa, both are possible in the repository's Settings) will make not only the code and files of that repo public, but also all the Issues. Which is fine, but it might add considerations in terms of what is discussed in those Issues.

OHI example

Here is an example of the Ocean Health Index team's thought process & strategy.

Our team works within a GitHub Organization called "OHI-Science". Within that Organization, we work in many repositories, with different combinations of people working primarily within different repositories. Sanity-wise, we didn't want to have conversations in Issues within

each of those repositories because it would make finding those conversations more difficult (although now GitHub can search all Issues/code across an Organization!).

We also wanted our repos to be public, but to have private conversations using the Issues feature.

These two needs led us to create a single private repository named "issues", and we only use it for Issues. This works really well for us, especially since our team lead can engage in these discussions by receiving emails about the Issues in his inbox, and can respond without having to go to GitHub.com.

Your Turn: Create & comment on issues

We will break into groups and you can explore some of these features in Issues. Here is what to do:

- Go to github.com/openscapes/demo/issues
- Create an issue, tag people in your breakout group (ask for their username)
- Browse issues, comment in other issues
- Try:
  - Linking to the .md document you created in the previous chapter
  - Creating a label and applying it, assigning people
  - Adding Issues to a Project (create one if need be)
  - Closing an Issue

Have fun! And throughout the process, talk to your breakout group, and share what you learn.

Here's what your inbox will look like afterwards:

[Openscapes/demo] Breakout group 3 is best -- which isn't an issue, really (#18) - @	**11:55 AM**
[Openscapes/demo] how to specify a certain script (#9) - @nt246 @joyceongjl — Y…	**11:50 AM**
[Openscapes/demo] testing out the labels (#16) - — You are receiving this because …	**11:49 AM**
[Openscapes/demo] what does this do? edits (#20) - now its something we need to…	**11:48 AM**
[Openscapes/demo] i'm thinking i might eat my arm instead (#19) - it'll be fine i hav…	**11:47 AM**
[Openscapes/demo] Practicing issues! (#14) - Making a new issue in Openscapes. …	**11:47 AM**
[Openscapes/demo] breakout room 3's issue (#15) - @agdedrick @hefroehl Uh oh, …	**11:46 AM**
[Openscapes/demo] How to create issues without drama! (#17) - These issues are …	**11:46 AM**
[Openscapes/demo] My first issue! (#11) - @chelsealwood @scrislee testing an iss…	**11:45 AM**
[Openscapes/demo] What is a Kaban board (#5) - @chelsealwood , @allisonhorst —…	**11:45 AM**
[Openscapes/demo] we're never going to get out of here alive! (#13) - this is a test …	**11:45 AM**
[Openscapes/demo] playing around with issues (#12) - how do i do this help @nt24…	**11:44 AM**
[Openscapes/demo] another issue (#6) - here is another issue — You are receiving t…	**11:44 AM**
[Openscapes/demo] list on readme (#10) - Can you add a list to my readme? @mst…	**11:43 AM**
[Openscapes/demo] we have issues (#8) - I could really use some help from @jaho…	**11:42 AM**
[Openscapes/demo] Important issue! (#7) - This is an important issue please help! …	**11:42 AM**
[Openscapes/demo] Need more people (#4) - have a look, @nicolas931010 and @j…	**11:41 AM**
[Openscapes/demo] i have no lunch today (#3) - @lilyzzhao tell my family i love the…	**11:41 AM**
[Openscapes/demo] all the things need to be done (#2) - do all the things! @scrisle…	**11:40 AM**

This is pretty rare to receive so many emails all at once. But you can always switch your setting to "Not Watch" this repository so that you only receive emails about Issues that you are tagged in.

Team Culture

We discuss team culture because while we know that diverse teams are more innovative, creating spaces where everyone can do their best work and feel safe to contribute takes intention; it does not happen by default. There is a lot of work to do to improve research culture, and we can lead by example in our own research groups and communities.

Slides

Psychological safety, contributed by Tara Robertson

Additional slides and recordings

Team culture. These slides were developed in 2019, before we updated to the above Psychological safety slides in 2021.

> Psychological safety (slides | recording). This presentation was contributed by Matt Fisher, National Snow and Ice Data Center (NSIDC) for the 2024 NASA Openscapes Champions Cohort.

See also the chapter on Codes of Conduct.

Why talk about team culture?

Role modeling sets a lot of team culture, and there is a lot we can learn and do to deliberately create a scientific culture that we want to be a part of.

Psychological safety

Amy C. Edmondson is the Novartis Professor of Leadership and Management at Harvard Business School.

a shared belief that team members will not be rejected or embarrassed for speaking up with their ideas, questions, or concerns

https://hbr.org/2022/03/research-to-excel-diverse-teams-need-psychological-safety

They recommend three ways to build psychologically safe environments: framing, inquiry, and bridging boundaries.

Frame meetings as opportunities for information-sharing. Frame differences as a source of value.

Open questions. Questions that build shared ownership and causality.

Listening -

```
Hopes and goals. What do you want to accomplish?
Resources and skills. What do you bring to the table?
Concerns and obstacles. What are you up against? What are you worried about?
```

Science benefits from diversity

And we need to be deliberate about welcoming and including people from diverse backgrounds.

A few articles from Nature with many more links within:

- Science benefits from diversity
- What does it take to make an institution more diverse?

Sexual harassment is rife in the sciences

Sexual harassment is rife in the sciences, finds landmark US study. *Existing policies to address the issue are ineffective, concludes a long-awaited report from the National Academies of Sciences, Engineering, and Medicine.*

Most common form is gender harassment: it's the "put-downs as opposed to come-ons".

We need to unlearn racism and build antiracist culture in science

- Ten simple rules for building an antiracist lab - Chaudhary & Berhe, 2020. PLOS
- Unlearning Racism in Geoscience — URGE

Put your values forward

Model the behavior you want to see in your research group & beyond (lab, dept, campus, organization, online)

Building trust

Have to build trust and be intentional, don't hope for organic.

- How to build (and rebuild) trust - Frances Frei

Sustain the culture

Overtly showing kindness & a Code of Conduct can filter out people who don't want to be subject to its enforcement – rOpenSci

Labs have people coming and going all the time; how do you set the tone and have it be sustainable?

Deliberately setting the tone

Opening remarks at RStudio::conf 2019, in front of an audience of 1700 at a global software conference, Chief Scientist Hadley Wickham announces the Code of Conduct, how to identify RStudio staff if you need help, and how to mingle with welcoming body posture to invite others to join. This set the tone of the whole conference to be the most positive I have ever attended.

Collegiality

We must deliberately set the tone for collegiality to create a positive, inclusive research group environment.

Safety and accessibility as parts of inclusion and empowerment. Does everyone feel safe to speak up? Does everyone have channels to contribute? This is especially true as the tech we use evolves. Who can participate?

This builds resilience to your research group. If someone needs to leave for a family emergency, maternity/paternity leave, vacation, set yourselves up so your team continue smoothly/ — Angela Bassa RStudio talk

Opportunity cost of not doing this. Burnout, people leaving science.

Team efficiency

We must deliberately set the tone to create a positive, inclusive research group environment that fuels team efficiency.

This means create a team mindset, and focusing on similarities rather differences. We all work on different projects and have different research questions, but we all have to wrangle data, organize version files, have things we don't know…let's create a space where we can talk about all this and find common ground to tackle together so we don't reinvent.

There can be an advantage to having team conventions. This can both reduce friction and reinventing the wheel. But there also needs to be room for different skills people come in with. For example, if they're more efficient in Python, don't want to force R. But want to create space where folks can interoperate and work together. The tech/software side helps with this, but it's our mindsets too. We need to be open to it.

Open software can facilitate open/shared culture

A lot to say here, for now, see:

- https://openscapes.github.io/slides/betterscience/environment-canada
- https://blogs.scientificamerican.com/observations/open-software-means-kinder-science/

Enabling & participating

Here are some ideas that you can support and participate in to learn and create a kinder team culture:

Seaside chats – discuss share data workflows

From Michelle Stuart's blog about the Pinsky Lab's first Fishbowl chat:

> This open communication has leaked into the general discussion going on in our open work space. Lab members seem more comfortable with asking teammates for help, and it is exciting to see all of us getting on the same page with our data science.

Hackathons or documentation parties – co-create

Social events

Get to know each other outside of work. Do some during work hours can include more people who can't participate after work

Onboarding – how to welcome new people to your research group

Asking for help

Create a welcoming environment where they know where to ask for help – They won't know what questions to ask. Provide structure.

Code of Conduct

See Additional Lessons > Code of Conduct

Further Resources

- A Practical Guide to Mentoring Across Intersections - Harriot 2020 *VanguardSTEM Blog*
- Get it wrong for me: What I need from allies - Carpenter 2020.
  - "Now, when someone asks, 'what do you need from me', I say, 'I need you to learn, I need you to care'. Somehow, we've all evolved to underestimate the power of learning and the power of seeking to understand. Knowing what things harm me is a sign that you value me. …Then I want an ally who works to change their individual behavior and change the system around us for the better. Not just one or the other. I want a bunch of people who are interested in becoming allies to me to get it wrong. Because I promise, you will get it wrong, likely more than once. But please get it wrong, for me. Be wrong on my behalf. Try stuff, learn stuff, make attempts, and fail. Embrace the discomfort of not knowing, of not being certain, of not understanding and then be motivated enough to learn and get better. I will

give you grace if you give me effort. We are risking our lives; you can risk getting things wrong."

- Inclusivity in STEM: Interview with Dr Mica Estrada (video, 17 mins). "Dr. Mica Estrada is a social psychologist and faculty member at University of San Francisco. Her research explores the role of identity and values in influencing the persistence of historically underserved students in STEM."
  - PEERS
  - micro affirmations
- Recreating Wakanda by promoting Black excellence in ecology and evolution — Schell et al (2020)
- For Our White Friends Desiring to Be Allies — Ariel (2017)
- Dr. Dori Tunstall on Respectful Design: Models for Diversity, Inclusion, & Decolonization — Tunstall (2020)
- Sexual harassment is rife in the sciences, finds landmark US study — Witze (2018)
- All We Can Save — Johnson & Wilkerson (2020)
- Braiding Sweetgrass — Kimmerer (2013)

Data Strategies

Data strategies are part of a shared workflow strategy: How do we structure data, where do we store and back up data, how do we create metadata, and the mantra of "keep the raw data raw" and distinct from your analysis. Here we will discuss personal and team habits for data management and sharing: data strategies for future us.

Slides

Data strategies for Future Us
Data Strategies in the Cloud. Tips for managing your data and adjusting for the cloud, by Alexis Hunzinger, NASA GES DISC (2024)
Data strategies for Future Us for Cloud, by Andy Barrett, NASA National Snow and Ice Data Center (NSIDC) (2023)
Data strategies for Future Us for Metadata, contributed by Jessica Couture (2021)

Video recordings

Data strategies for Future Us, contributed by Ileana Fenwick, Openscapes Pathways to Open Science.

Additional slides

These slides were contributed by researchers from previous Champions cohorts who shared examples and stories from their work.
Data strategies; examples from NOAA Fisheries Alaska (Marine Mammal Lab Stock Assessment Report), contributed by Rod Towell, Nancy Young, Tony Orr, Brian Fadely, Erin Richmond
Data to Product Workflows, contributed by Emily Markowitz (video)

Data organization in spreadsheets

We'll start off discussing the Data organization in spreadsheets publication by Broman & Woo, 2018, which appears in the "Practical Data Science for Stats" collection in PeerJ & American Statistician.

It is a delightful read, from the first opening sentences:

> "Spreadsheets, for all of their mundane rectangularness, have been the subject of angst and controversy for decades…. Amid this debate, spreadsheets have continued to play a significant role in researchers' workflows. The dangers of spreadsheets are real, however – so much so that the European Spreadsheet Risks Interest Group keeps a public archive of spreadsheet 'horror stories'…"

Broman & Woo share practical tips to make spreadsheets less error-prone, easier for computers to process, and easier to share. And something incredibly cool, it's the 3rd most downloaded stats paper in the American Statistician, behind 2 papers about p-values (twitter thread).

Read their paper for strategies behind their basic principles:

1. Be consistent
2. Write dates like YYYY-MM-DD
3. Don't leave any cells empty
4. Put just one thing in a cell
5. Organize data as a rectangle ("Tidy data")
6. Create a data dictionary
7. Don't include calculations in the raw data files
8. Don't use font color or highlighting as data
9. Choose good names for things
10. Make backups
11. Use data validation to avoid data entry errors
12. Save the data in plain text files

Good enough practices in scientific computing

Next we'll explore this Good enough practices in scientific computing publication by Wilson et al. 2017 in PLoS Computational Biology. It follows a previous publication by Wilson et al. 2014: Best practices for scientific computing.

In terms of data management recommendation, they have 2 main themes:

1. work towards ready-to-analyze data incrementally, documenting both the intermediate data and the process
2. embrace the idea of "tidy data", which can be a powerful accelerator for analysis

Read their paper for strategies behind their basic principles (Box 1):

1. Save the raw data.
2. Ensure that raw data are backed up in more than one location.
3. Create the data you wish to see in the world.
4. Create analysis-friendly data.
5. Record all the steps used to process data.
6. Anticipate the need to use multiple tables, & use a unique identifier for every record.
7. Submit data to a reputable DOI-issuing repository so that others can access & cite.

The publication also covers:

- Software: write, organize, and share scripts and programs used in an analysis.
- Collaboration: make it easy for existing and new collaborators to understand & contribute to a project.
- Project organization: organize the digital artifacts of a project to ease discovery & understanding.
- Tracking changes: record how various components of your project change over time.
- Manuscripts: write manuscripts in a way that leaves an audit trail & minimizes manual merging of conflicts.

Tidy data for efficiency, reproducibility, & collaboration

We'll explore the concept of **tidy data** through the illustrated series Tidy data for efficiency, reproducibility, & collaboration by Lowndes & Horst 2020, posted on the Openscapes blog.

When we talk about organizing data to help us work in an efficient, reproducible, and collaborative way, we are talking about TIDY DATA. We mean deliberately thinking about the shape and structure of data – something that might not seem super exciting but is truly game-changing.

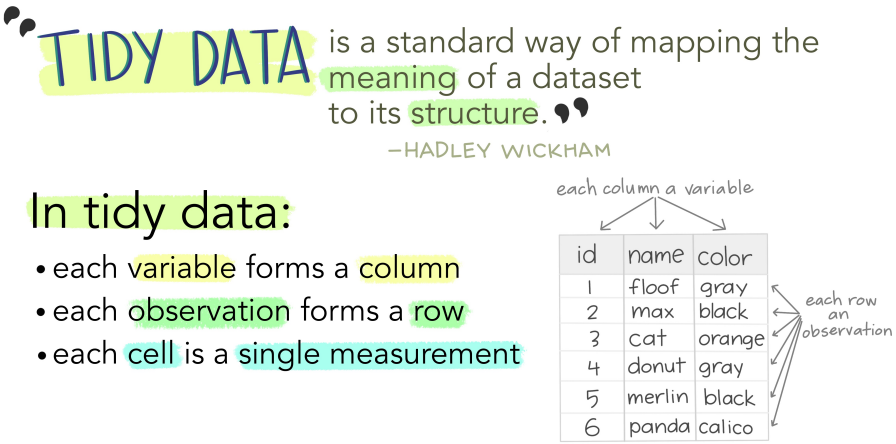
Tidy data has been mentioned in each of the above, as a way to organize data in spreadsheets, to prepare ready-to-analyze data, and for sharing with the FAIR principles. And remember

that "tidying data ("data wrangling") – up to 50–80% of a data scientist's time" Lohr 2014, New York Times, so it's important to leverage these existing philosophies and tools.

So let's talk about what tidy data is and why it is so empowering for your analytical life.

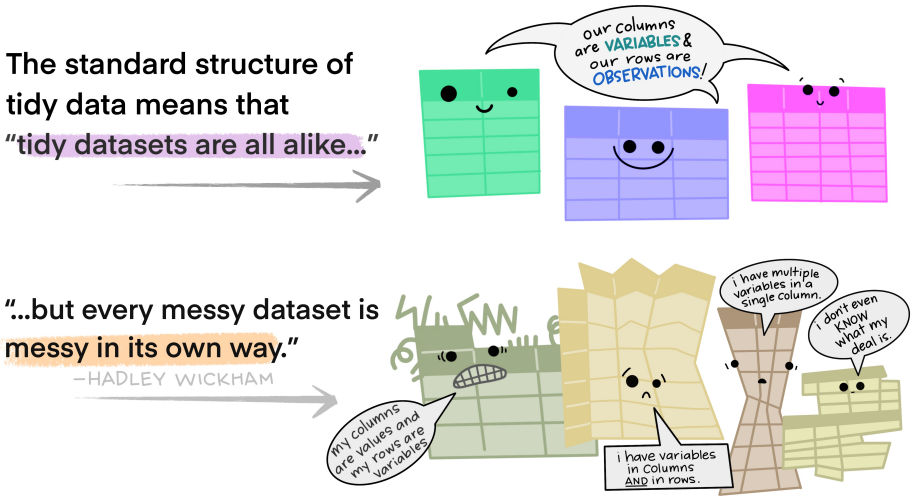
What is tidy data?

Tidy data is a way to describe data that's organized with a particular structure – a rectangular structure, where each variable has its own column, and each observation has its own row (Wickham 2014).



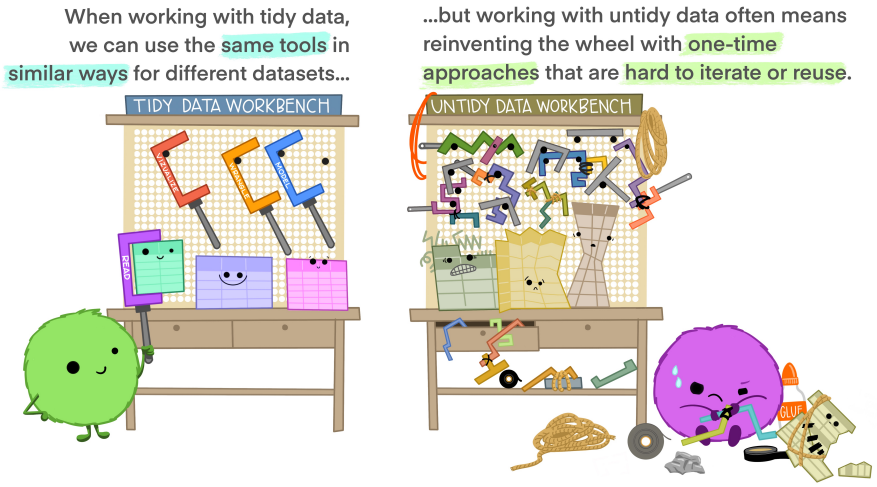
Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

This standard structure of tidy data led Hadley Wickham to describe it the way Leo Tolstoy describes families. Leo says "Happy families are all alike; every unhappy family is unhappy in its own way". Similarly, Hadley says "tidy datasets are all alike, but every messy dataset is messy in its own way".



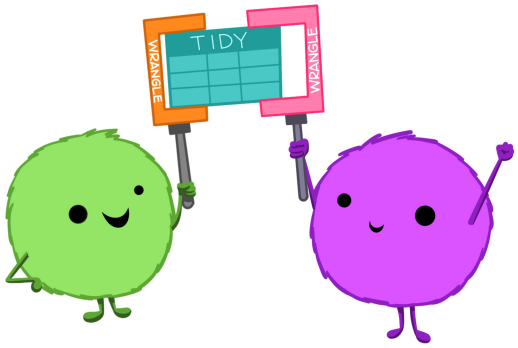
Tidy data for more efficient data science

Tidy data allows you to be more efficient by using existing tools deliberately built to do the things you need to do, from subsetting portions of your data to plotting maps of your study area. Using existing tools saves you from building from scratch each time you work with a new dataset (which can be time-consuming and demoralizing). And luckily, there are a lot of tools specifically built to wrangle untidy data into tidy data (for example, in the tidyr package). By being more equipped to wrangle your data into a tidy format, you can get to your analyses faster to start answering the questions you're asking.



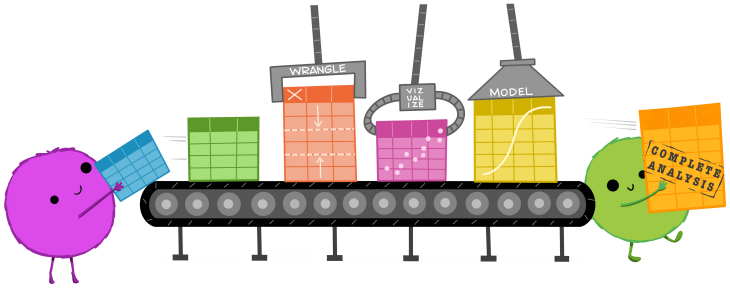
Tidy data for easier collaboration

Tidy data makes it easier to collaborate because our friends can use the same tools in a familiar way. Whether thinking about collaborators as current teammates, your future self, or future teammates, organizing and sharing data in a consistent and predictable way means less adjustment, time, and effort for all.



Tidy data for reproducibility and reuse

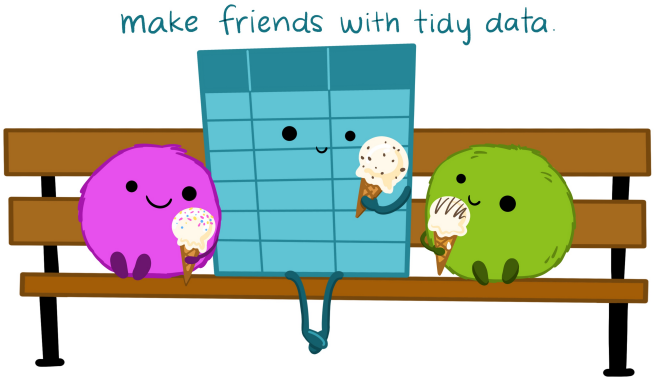
Tidy data also makes it easier to reproduce analyses because they are easier to understand, update, and reuse. By using tools together that all expect tidy data as inputs, you can build and iterate really powerful workflows. And, when you have additional data entries, it's no problem to re-run your code!



Tidy data for the win!

Once you are empowered with tools to work with tidy data generally, it opens up a whole new world of datasets that feel more approachable because you can work using familiar tools. This transferrable confidence and ability to collaborate might be the best thing about tidy data.

So for more efficient, reproducible, and collaborative analyses, make friends with tidy data!



Learning Resources

- Wickham, Çetinkaya-Rundel, & Grolemund (2023). R for Data Science: https://r4ds.hadley.nz/
  - See Ch 12: Tidy Data
- Wickham, H (2014). *Tidy Data*. Journal of Statistical Software 58 (10). jstatsoft.org/v59/i10/
- Broman, KW and KH Woo (2018). *Data Organization in Spreadsheets*. The American Statistician 72 (1). Available open access as a PeerJ preprint.
- Leek, J (2016). How to share data with a statistician

Coding Strategies

Coding strategies can blend with workflow strategies, and the idea is working in a way that is not just for you in this moment. Here we will discuss good coding practices for beginning and seasoned coders alike that make it easier to work with other people, times, and computers.

Slides

Coding strategies for future us
Data strategies for parallelizing code. Diving into parallel processing from a scientist perspective, by Mahsa Jami, NASA LP DAAC

Additional slides

Previous iterations of Coding strategies for future us: Filepaths and Project-oriented workflows
Expanding ouR community!, contributed by Dr. Chanté Davis
State of the Ecosystem Product Development Workflow, contributed by Kim Bastille

---

WTF: What they forgot to teach you about R

Most of this advice comes directly from Jenny Bryan & Jim Hester's awesome course What they Forgot to Teach You About R. We highly recommend reading Chapters 1-4 that go into much better detail than we cover here. The advice here is solid coding practices for any language, with examples from R.

Workflow versus product

Distinction between things you do because of personal taste & habits ("workflow") versus the logic and output that is the essence of your project ("product").

Workflow:

- Editor you use to write code.
- Name of your home directory.
- R code you ran before lunch.

Clearly product: - Raw data. - R code someone needs to run on your raw data to get your results, including the explicit library() calls to load necessary packages. (script, notebook)

Ideally, you don't hardwire anything about your workflow into your product.

Source files

What are they and why?

Code that creates objects is "source code". Source code is essentially text files you edit in a text editor that is then executed in the console.

Examples:

- .R, .Rmd
- .py
- .m

Save the source, not the workspace

Save the source code; do not save the R object itself.

Save your commands as "scripts" (`.R`, `.py`) or "notebooks" (`.Rmd`, `ipynb`). It doesn't have to be polished. Just save it!

Everything that really matters should be achieved through code that you save – including objects and figures The contrast is storing them implicitly or explicitly, as part of an entire workspace, or clicking via the mouse.

Load libraries/packages at the top. Just like a recipe: tell us the ingredients need before we get going!

Always start R with a blank slate

Saving code is an absolute requirement for reproducibility.

When you quit, do not save the workspace to an .Rdata file. When you launch, do not reload the workspace from an .Rdata file.

In RStudio, set this via *Tools > Global Options.*

Restart R often during development

> "Have you tried turning it off and then on again?" – timeless troubleshooting wisdom, applies to everything

If you use RStudio, use the menu item Session > Restart R

Additional ways to restart development where you left off, i.e. "re-run all the code up to HERE"

Avoid `rm(list = ls())`

It's common to see scripts begin with this object-nuking command: `rm(list = ls())`

This is highly suggestive of a non-reproducible workflow.

The problem with `rm(list = ls())` is that, given the intent, it does not go far enough.

It only deletes user-created objects from the global workspace.

Instead, Restart R with a clean slate OFTEN (e.g. many times/day), and write every script assuming it will be run in a fresh R process

Filepaths

Every saved thing gets a unique path.

Your code needs to run from somewhere specific. And when it interacts with other things (data or other code), you need to tell your code where things are.

The more deliberate you are about where things live,

- The easier it will be for you and future you
- The easier it will be for other people
- The easier it will be on another computer

setwd("path/that/only/works/on/my/machine")

The chance of `setwd()` having the desired effect – making the file paths work – for anyone besides its author is 0%.

It's also unlikely to work for the author one or two years or computers from now.

Hard-wired, absolute paths, especially when sprinkled throughout the code, make a project brittle. Such code does not travel well across time or space.

setwd()

BUT, if you still decide to use `setwd()` in your scripts, you should at least be very disciplined about it:

Only use `setwd()` at the very start of a file, i.e. in an obvious and predictable place.

Always set working directory to the same thing, namely to the top-level of the project. Always build subsequent paths relative to that.

R users: use the `here` package

`here()` identifies your project's files, based on the current working directory at the time when the package is loaded. https://here.r-lib.org/.

```
library(here)
here()
```

Project oriented workflows

Dilemma and Solution

Problem statement:

We want to work on project A with the working directory set to path/to/projectA (my data analysis) and on project B with the working directory set to path/to/projectB (my teaching material).

But we also want to keep code like setwd("path/to/projectA") out of our scripts.

Solution:

Solution: use an IDE that supports a project-based workflow.

An integrated development environment (IDE) offers:

- a powerful, R-aware code editor

- many ways to send your code to a running R process
- other modern conveniences

And it eliminates:

- temptation to develop code directly in the Console. (instead:.R!)
- tension between development convenience and portability of the code.

Organize your work into projects

Here's what I mean by "work in a project":

- File system discipline: put all files related to a project in a designated folder.
  - This applies to data, code, figures, notes, etc.
  - Depending on project complexity, you might enforce further organization into subfolders.
- Working directory intentionality: when working on project A, make sure working directory is set to project A's folder.
  - Ideally, this is achieved via the development workflow and tooling, not by baking absolute paths into the code.
- File path discipline: all paths are relative — relative to the project's folder.

Synergistic habits: you'll get the biggest payoff if you practice all of them together.

Portability: the project can be moved around on your computer or onto other computers and will still "just work". is the only practical convention that creates reliable, polite behavior across different computers/users/time. This convention is neither new, nor unique to R.

It's like agreeing that we will all drive on the left or the right. A hallmark of civilization is following conventions that constrain your behavior a little, in the name of public safety.

RStudio Projects

The RStudio IDE has a notion of a (capital "P") Project, which is a very effective implementation of (small "p") projects.

Project have an.Rproj file in the folder, which is used to store settings specific to that project. Use File > New Project … to get started.

Allows for multiple projects

no danger of crosstalk: each has own R process, global workspace & working directory

Same "unit" as a GitHub repo!

Tips for RStudio Projects

One suggestion for organizing:

Have a dedicated folder for your Projects. - If you have One Main Place for Projects, then go there in Finder/File Explorer to launch any specific project with .Rproj. - Mine is called "~/github/".

Switching Projects: RStudio knows about recent Projects.

Name files deliberately

Jenny Bryan's 3 rules for Naming Things:

- machine readable
- human readable
- plays well with default ordering

Slides available from Speakerdeck. See also Jenny Bryan's "Naming things" video (5 mins) from NormConf · Dec 4, 2022

Software considerations for coding

The following advice is from Tiffany Timbers, UBC Data Science, Intro to Reticulate:

You will need these software tools:

- Programming language (R, python)
- Code editor (RStudio IDE, Jupyter)
- Version control software (git, GitHub/bitbucket)

How to choose the "best" tool for the job:

- Reproducible and auditable
- Accurate
- Collaborative (and portable)

If you're choosing between R, Python, and other modern languages, they will aready be reproducible, auditable, and accurate. That leaves collaboration – what do your collaborators use? What do folks in your lab, or field use? What is mentioned in the papers you read? There is increasing interoperability between languages (e.g. see reticulate to run python code from R) so when you have some idea it's best to get started!

See also: Opinionated analysis development (Parker 2017). Tools like RStudio are already doing this to help you. Reserve your mental energy for the fun part of the analysis!

Further reading

- Tidyverse Skills for Data Science - Wright, Ellis, Hicks, & Peng
- Principles for data analysis workflows - Stoudt, Vasquez, and Martinez 2021, PLOS Computational Biology
- R Cookbook — JD Long
- Big Book of R - Oscar Baruffa
- R for Excel Users - Lowndes & Horst
  - R/RStudio workflows with tidyverse, RMarkdown, and GitHub, using ecological data from LTER (update from OHI's intro to data science)
- An introduction to Earth and Environmental Data Science - Abernathy
  - Intro to Python, JupyterLab, Unix, Git, some packages & workflows
- Data analysis and visualization in Python for ecologists - Carpentries
  - Setup recommends using Anaconda and Jupyter Notebooks
- Python for Data Analysis - Thompson
  - Assumes no previous experience.Also intro to the command line.

Open Communities

Open communities play a big role in advancing research, helping research feel less lonely and reducing the amount of time we all spend being stuck reinventing the wheel. This is a brief (incomplete) introduction to the idea of open communities. We will explore: What are open communities? Why engage with them? How to engage with them? And also how social media is a legit tool for coding and science. Originally focusing on R communities, we are adding more examples beyond including python and Earth science data communities.

Slides

Open Communities

Additional slides

Discovering Community Tools

In your work, you're not limited to the people & resources in your physical environment! Open communities are waiting for people like you to come participate in whatever way feels

comfortable right now. Awesome people are developing and sharing code and learning resources. They connect through open communities and software (GitHub, forums, Mastodon, Bluesky, TikTok, Slack)

We can all learn, reuse, remix, share, and contribute and this means less reinventing, more doing, more fun!

What are open communities?

Figure 8: Dr. Bri Lind, Land Processing Data Center (LP DAAC) Presenting at NASA's Hyperwall @ Ecological Society of America Conference, August 2023.

Open communities are groups of **people** openly creating, sharing, teaching, collaborating.

They are united around a **shared interest**: coding language, topic, discipline, etc.

They have a **culture** of shared & continued learning, prioritize diversity, equity, inclusion.

They **connect** online and / or in-person.

Examples:

Figure 9: Some open communities with in-person and online engagement opportunities. The slides linked above have links and full details.

Read more in Collaborating & getting help, in R for Excel Users (Lowndes & Horst, 2021).

Open communities have a vibe

They signal whether a community is for you. Check out their Contributing guides, READMEs & CONTRIBUTING.md.

- rOpenSci Community Contributing Guide
- CONTRIBUTING.md template for R packages (Desmet)
- FIMS Developer Guide, NOAA Fisheries Integrated Modeling System has a whole chapter on Contributor Guidelines.

Are they inclusive? Some are more welcoming of newer users e.g. community.rstudio.com vs Stackoverflow #rstats

Why engage with open communities?

- Skillshare (teach & learn)
  - Coding, software, workflows

  - Collaboration, leadership, DEIJ

- Meet allies, grow friendships, career opportunities
- Help improve science & scientific culture
  - Increase visibility & value of coding, data, collaboration skills
  - Drive change: modern ways to contribute to science. Formally incentivize and teach! Include in promotion/tenure (we shouldn't really have to teach ourselves on our own time and not be credited), create jobs
- Carry on & forward your experience from Openscapes

How to get started?

Anywhere you're starting from is great. Know what you need. You don't have to be an expert to learn together & teach what you know.

Talk to your colleagues: where do they learn? Seaside Chats are a great way to start - tell others you're meeting!

Google what you need, see who is working on those topics

- Look for tweets, toots, blogs posts, tutorials, slides, websites, forums
- When you Google, include what you want to do PLUS r, rstats; python.
- Ideas: Big Book of R (compiles 250+ R tutorials), Pangeo Showcase

Engage within your limits: 5 min reading Mastodon toots; 30 min sharing your use case in a public forum; 1 hr attending a webinar then 15 min to share with your team what you learned.

In the past, Twitter has been a legit tool for science, a good way to learn what you need to learn and broadening your horizons while building community. We've learned from `#rstats, #rstatsES, #rspatial, #pydata, #BlackAndSTEM, #MeTooSTEM`. Many of us are still seeking a new thriving place for this kind of discourse. When you find that place, start by listening and learning. Be deliberate; curate who you follow. Like and share when you're comfortable. Then, contribute – your ideas, your blogs, your papers, your code…

Impacts of open communities

ggplot & knitr were created by students (Wickham & Xie) (McBain 2019)

- Increase visibility & value of coding, data, collaboration skills
- Drive change: modern ways to contribute to science, and not based on hierarchy

R-Ladies changed the landscape of the R project (de Queiroz 2019, opens a pdf)

- Mission: to promote gender diversity in the R community.

- Influenced who is a contributor
- Role-modeling collaboration, leadership, diversity, equity, inclusion, belonging, justice

Culture and belonging

> "Open science saved me. I found community in RStudio and its users. Openscapes changed my workflow. I discovered my passion for data science education and engagement" (Fenwick 2022)

Open community examples

Government peer-learning groups

- NMFS Open Sci (NOAA Fisheries)
  - see also https://openscapes.github.io/series/forkeds-lessons/#nmfs-openscapes
- Interagency Ecological Program Data Science Project Work Team
- R Govys
- EPA CEMM Data Science Technical Exchange, R & Python user groups, Data Science Community of Practice

Campus coding clubs

- BioData Club at OHSU
- The Bio-Data Club at Moffitt

Pathways

This page has guidance and examples around the Pathways document; to learn more about what teams accomplish, explore stories from previous cohorts.

The concept

Perhaps the most important part of Openscapes is helping teams identify their trailhead together, as illustrated in the Champions landscape illustration.

We support individuals to "find their teams" by discussing approaches and software, so that they find they have common parts of how they work, whether they are working on the same or different projects. This builds from the ideas of creating space and place in order to find the common, as introduced in the better science chapter.

Slides

The Pathway Intro slides in the latter part of this deck were presented in a Champions Cohort.
Documentation & the Pathway

The Pathways Spreadsheet provides a structured way for your team to think how you work and find common approaches and needs (template).

Openscapes Pathway	Topic	Now	Next steps
Reproducibility	Data storage	icloud and google docs, dropbox	Github and Rmarkdown
	Metadata	A sheet in each excel doc	A metadata sheet for Github?
	Data prep	Excel files, Rstudio	Data cleaned up on Github, protocol
	Data analysis	Rstudio, minitab, JMP	Uniform coding practices in R
	Version control	Some people work in google docs, microsoft suite	Github and Rmarkdown
	Organization	Solo	Github
	Lab notebooks and protocols	Some of them in our google folder, computer	Github, read me file!
Collaboration	Coding	Email, evernote	Github, use projects!
	Storing data	Google drive, one drive, icloud, computers, hardrives, cluster	Github, large files -> google drive/dropbox/ github?
	Internal discussions	Lab meeting	Lab Slack channel
	File sharing	Email and Google Drive	Github
	Project management / to-do lists / milestone tracking	Google calendar, sticky notes, blank pieces of paper	Lab notebook function on Github
	Onboarding	Google documents, EH&S trainings	Github for specific projects, lab notebooks
Communication	Sharing data	Google spreadsheets	Github
	Sharing methods	Lab Google Drive folder	Github, slack
	Talks	icloud, google drives	None right now
	Teaching	Canvas, zoom, google docs, kahoot	Github, slack
Culture	Team agency (ability to make deci	Lab meeting	Data science meetings, slack chat
	Supervisor sponsorship	Lab meeting every week to check in mentor and students	Data science meetings!
	Promoting DEI	DEI discussions + folder with resources	New discussion leads
	Seaside Chats	Zoom	In person!

Figure 10: Example Pathway Spreadsheet documenting how the team is working now and ideas for next steps

This format has helped many groups think through their trailhead and identifying where they are now. Some groups have found it more useful to draw things out as a workflow on a whiteboard or google slide. The spreadsheet format might not work for your group, and that is okay; use whatever format makes sense for you.

Planning guidance

You'll develop your Pathway by talking with others on your team, screensharing ("show me"), and asking questions.

You'll use the document by creating a copy of the template and then discussing with your team. Start with the "Now" column. How do you work now? Add rows as best reflects your work,

but take a moment before deleting them in case it's something you haven't thought about in this way before rather than if it's not relevant to you. Then, move on to the "Next Steps" column as you think through priorities and learn from/with/for your cohort. This will be a work in progress throughout the cohort that you'll present a snapshot of in the final Cohort Call (see next) and that you can revisit following the cohort as well.

Presenting guidance

On the final Cohort Call, each team presents their Pathways. Each team has 3 minutes to share followed by 2 minutes for questions. We encourage leads not to present, and it's great when we hear multiple voices from the teams.

This is informal sharing of unpolished work-in-progress. Everyone makes progress throughout the Cohort: in mindsets, planning and actions. This is an opportunity to reflect and talk about it, building on the reflection breakout rooms that began most Cohort Calls. We've been reflective each week and we've all made progress.

Presentations do not need be line-by-line of the Pathways Spreadsheet and there's no wrong way to talk about your progress. Whatever your group wants to create and share is great - could be a photo of a whiteboard, a slide, the Pathways spreadsheet, or a "screenshare and tell" live walk-through of your files/code/ideas.

A few reflection prompts that can help frame the presentations:

- What are the practices we've covered during the Cohort that most compel you?
- What are the practices we've covered during the Cohort that most confuse you?
- Are there practices that are new to you that you didn't originally think would work with your own research interests? If so, which?
- What are your two biggest take-aways from the Cohort?
- What future revisions will you propose making for your pathway?
- Any final questions you want to workshop with the Cohort?

Pathways stories

Here are a few one-slide pictures of Pathways presented by Champions teams in their final sessions:

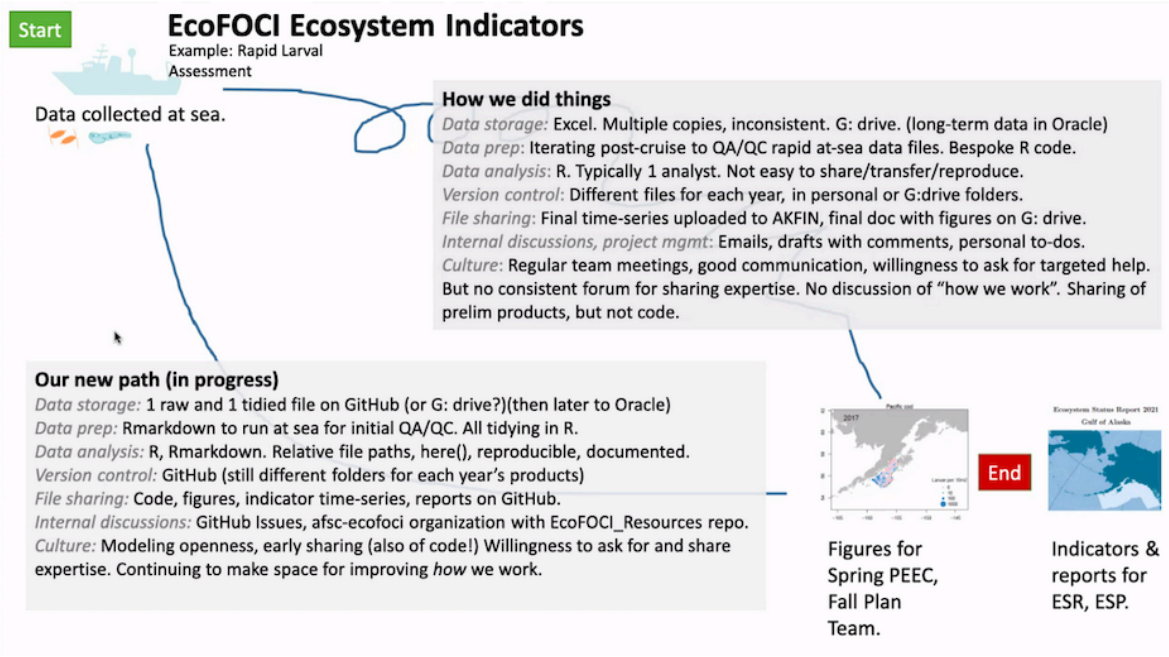


Figure 11: An example of an Openscapes pathway written by the NOAA Alaska Fisheries Science Center EcoFOCI team. The diagram summarizes how they used to share code, data, and files, and what their ongoing plans are for streamlining these processes and creating a team culture of openness and psychological safety. Source: A supportive forum for continued learning and collaboration at NOAA Fisheries Alaska

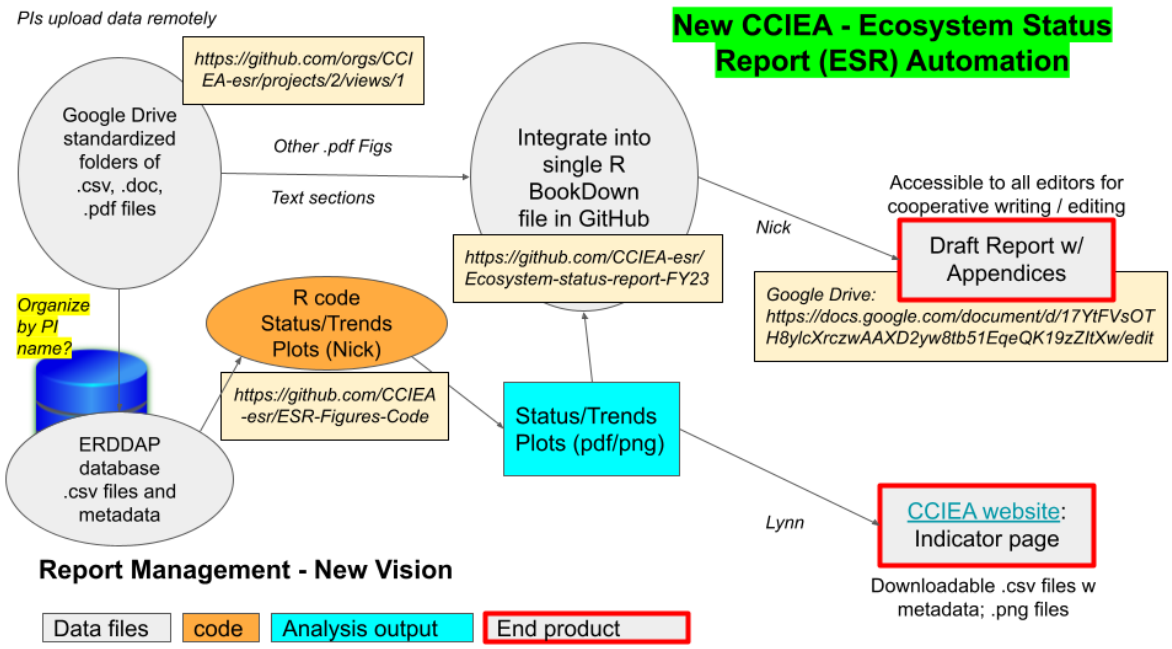


Figure 12: NOAA Northwest Fisheries Science Center Ecosystem Status Report Automation team was united by working on reports with many contributors of maps, time series data that must be compiled into 20-page reports. Source: Nationwide Openscapes Training at NOAA Fisheries Science Centers: Facilitating Collaboration, Skill-sharing, and Open Science

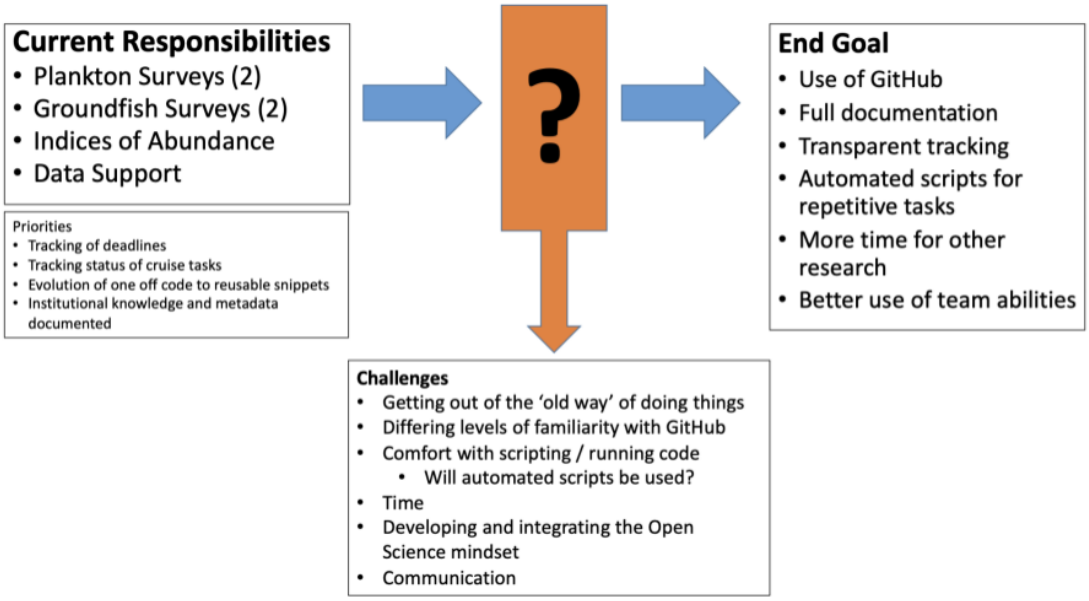


Figure 13: The NOAA Southeast Fisheries Science Center Trawl and Plankton Branch team focused on "Where are we going and how do we get there", and plans to do better documentation, use GitHub, which in the near-term involves setting up discussions to include more team members and set up collaborative processes. Source: Nationwide Openscapes Training at NOAA Fisheries Science Centers: Facilitating Collaboration, Skill-sharing, and Open Science

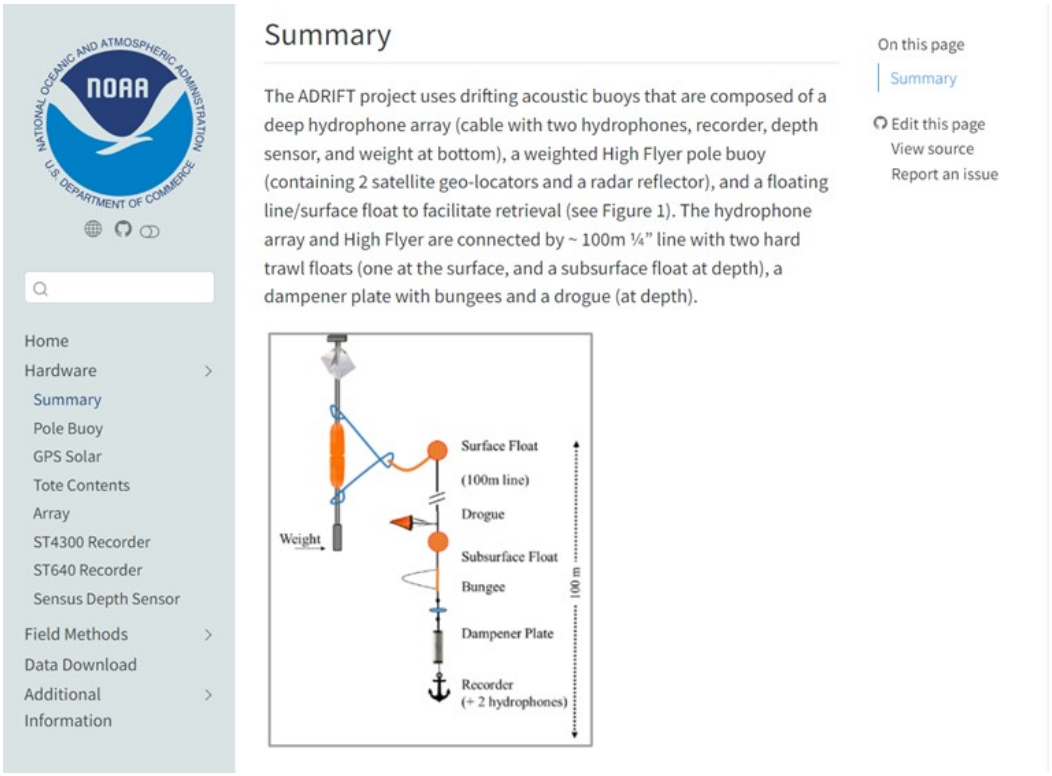


Figure 14: Screenshot from the ADRIFT Field Methods website created through the Openscapes Program. This is a living document where we can outline our methodology and update/archive specific components of our methods and hardware as changes are made. Credit: NOAA Fisheries / Kourtney Burger. Source: Sound Bytes: Championing Open Science

Read more

- Champions Program blog posts; you'll find more Pathways shared as slides and stories following the 2-month cohort
- A TL;DR Collection of Stories from Openscapes Champions and Mentors; these stories are largely from 3 months+ following Pathways presentations

3 Additional Lessons

Code of Conduct

Please see accompanying slides until this chapter is built out more.

Please also refer to Openscapes' Code of Conduct: https://openscapes.org/code-of-conduct

Important for:

- Conferences & workshops
- Online communities
- Labs & departments

Code of Conduct also known as

Community Participation Guidelines
Code of Practice

Similar ideas:
Lab philosophy, mission statement, participation agreements

Requirements

- Clear explicit statements
- Seen and heard – that all participants know about
  - Accessible and discoverable online
  - Mentioned aloud in meetings/interviews/onboarding
- Avenues for action, identified committees, recusals

Case study: rOpenSci

CoC itself
Blog post about the CoC
Blog post about creating the CoC — following a community call on the topic. Interesting points:

- Drafting – make it good and revisit; but not a living doc
- Adopting and sharing – so people know it exists & where to find it
- Reporting and enforcing – standardized form can help

Examples to build from

- rOpenSci CoC

- Buffer's CoC & Why It's Important For Diversity And Inclusion
- Mozilla Community Participation Guidelines
- Template CoC for labs

Further Reading

- How Codes of Conduct Are Combatting Open Source's Diversity Problem — Margaret Rhodes, AIGA Eye on Design

Reporting & SciComm

We can use Quarto and RMarkdown as powerful tools for reporting and science communication. So much potential for public communication, lab protocols and documents, etc. with R and Python workflows.

We've talked and taught about this in different places:

- Documenting things: openly for future us - September 19, 2023, at posit::conf(2023) shared practical tips and repeatable strategies (from impact of NASA Openscapes & beyond)
- Data science as an entryway to open publishing - May 27, 2020: Fireside Chat co-presented with Dr. Nick Tierney at Open Publishing Fest.
- These 2019 Champions slides show some early descriptions for teaching

Further resources

Strategies with RMarkdown also apply to Quarto

- RMarkdown - RStudio (2021)
- How I teach RMarkdown - Hill (2020)
- R Markdown for scientists - Tierney (2020)
- R for Excel Users - Lowndes & Horst (2020)
- RMarkdown website tutorial - Lowndes (2016)
- Academic Publications with R Markdown- Ovando 2020

Documentation

Work culture exists whether or not you talk about it or write it down.

Corey Clatterbuck and Anna Holder, co-leads of Openscapes Champions Cohorts at the California Water Boards / CalEPA created this lesson on Documentation. Corey brought a new perspective to documentation starting with "why document and what to document" and Anna Holder introduced thinking about your Pathway.

> Slides
>
> Documentation & the Pathway

Openscapes at the Water Boards is the home for the California Water Boards "fork" of the Openscapes Champions Program.

4 Forked Lessons

"Forking" is a concept from GitHub where you make a copy of someone's files (traditionally code and documentation) into your own workspace so you can reuse, tailor, and extend to your needs. Importantly, your forked work is still networked back to the original with credit, and it is visible to others so they can learn from you, and find the original source.

Open science ideas and Openscapes approaches can be forked - this is a contributing way we can see culture change in action. There are growing resources developed by the Openscapes community as they make real change within their research groups, across their organizations, and beyond.

This is the beginning of a growing list; please share others to add as you see them You can suggest via GitHub on the right of this page - click "Edit this page" or "Report an Issue", or email hello @ openscapes.org.

NASA Openscapes Cloud Tutorials

The NASA Earthdata-Openscapes Mentors team and other contributors are creating open educational resources to help researchers migrate workflows to the Cloud - all available for reuse and remix.

NASA Earthdata Cloud Cookbook

NOAA Fisheries Openscapes

The overarching vision of the NOAA Fisheries Openscapes is to support scientific researchers within the National Marine Fisheries Service (NMFS) by providing training in reproducible scientific workflows and platforms, facilitating collaboration across divisions in common scientific data science tasks, and providing shared "best practices" resources. Check out the NMFS R UG (R Users Group) for upcoming events and Wiki for links to resources and examples of open science at NOAA Fisheries.

NMFS-OpenSci (github)

NMFS-OpenSci Resource Book

NMFS-Openscapes (github)

Highlighting a few lessons by Eli Holmes and Em Markowitz:

- Advanced GitHub Clinic
- googledrive R package
- RVerse Tutorials, particularly Practical R+Git Workflow for Scientists May-July 2022 by Eli Holmes
    - RVerse Tutorials is a collection of teaching materials on R and R Workflow by the Mathematical Biology Program and the Northwest Fisheries Science Center.

Fay Lab Manual

This is the lab manual for the Fay Lab at the University of Massachusetts Dartmouth School for Marine Science and Technology.

https://thefaylab.github.io/lab-manual/

Check out how they've organized this onboarding document that prospective students and postdocs have said stood out to them. Consider reusing/remixing for your own group!

Slides from Gavin Fay's 2022 talk at NOAA Alaska Fisheries Science Center

Getting to Know Open Science

Getting to Know Open Science: How to Engage and Flourish in the Growing Open Science Community

This is a slide deck presented on August 9, 2023 by Bri Lind at the NASA Hyperwall at the Ecological Society of America 2023 Annual Meeting, Portland, Oregon.

It is a super introduction to why open science.

California Water Board Data Center

https://github.com/CAWaterBoardDataCenter

R for Excel Users

This course is for Excel users who want to add or integrate R and RStudio into their existing data analysis toolkit. It is a friendly intro to becoming a modern R user, full of tidyverse, RMarkdown, GitHub, collaboration & reproducibility.

https://rstudio-conf-2020.github.io/r-for-excel/

Check out the resources throughout as well - it's not only for Excel users, it's for everyone.

Allison Horst - open educational resources

Allison Horst shares many open educational resources, including workshops and slides, at https://www.allisonhorst.com/:

> "I actively work to contribute to open educational resources, including software for data science education (e.g. the palmerpenguins R dataset package), and artwork for data science & stats teachers and learners. From 2019 - 2020 I was RStudio's first Artist-in-Residence. You can read more about my motivation in that role in this blog post, and you can find some of my R- and stats-related artwork in this repo."

Allison developed the environmental data science program at UCSB https://ucsb-meds.github.io/.

Tidy Tuesday with lab groups

Reusing open approaches and ideas learned in Openscapes is awesome! Tidy Tuesday is a great example of this.

Allison Horst, 2019 Champion describes how she hosted in-person events with students and broader community:

> https://openscapes.org/blog/2019/05/02/tidy-tuesday-coding-club/

Nyssa Silbiger, 2021 Champion describes how she hosted sessions with her research group:

> For our group we had 2 hour sessions (1 hour is a bit too short for more novices and 2 hours gives the more advanced students time to try really new cool things). And we have a facilitator (usually me) that just checks in with the crew every few minutes and helps when students get stuck. But, it is very informal in that everyone just shows up and codes and chats. I have done this both in a conference room and at a bar. I also usually have people put some of their works in progress or final plots on the screen to show everyone else and share their "coding wins" or ask for help from the audience. And, yes, we use the Tidy Tuesday data because then they can also share their plots and code on twitter with the community and see the awesome creativity of the broader R community as well.

5 How Do I...

This is a collection of resources that complement what we cover in the Champions Lesson Series - things that often come up as questions and that folks can screenshare and do together during Seaside Chats and Coworking sessions. We point to existing resources as much as possible.

```
Warning in readLines(file): incomplete final line found on
'how-do-i/setup-rstudio-github.qmd'
```

Setup RStudio & GitHub

Installation

To setup RStudio and GitHub, you will need four things:

1. R
2. RStudio
3. Git
4. GitHub

Follow the UCSB MEDS Installation Guide for detailed instructions on how to create accounts, download, install, and configure on Mac and Windows. (For an even more detailed walk-through, see Allison Horst's ESM 206 Google Doc). Thanks for sharing Allison!

Practice

Learn R interactively

swirl lets you learn R, in R.

swirl teaches you R programming and data science interactively, at your own pace, and right in the R console!

It's developed by Sean Kross, who also co-created the Tidy Data Tutor

R and GitHub together

Now you'll want to practice how to use RStudio and GitHub. Follow R for Excel Users' GitHub Chapter to create a GitHub repository, clone it to RStudio, make edits, and practice syncing.

Do & fix git things

These are our go-to resources about GitHub; they include philosophy, step-by-step guidance, and troubleshooting. Most examples are with R and RStudio, and are applicable more broadly.

Excuse Me, Do You Have a Moment to Talk About Version Control?

"Excuse Me, Do You Have a Moment to Talk About Version Control?" Bryan J, 2017, PeerJ

Happy Git With R

> "Happy Git provides opinionated instructions on how to Install Git and get it working smoothly with GitHub, in the shell and in the RStudio IDE; Develop a few key workflows that cover your most common tasks; Integrate Git and GitHub into your daily work with R and R Markdown."

https://happygitwithr.com/

R for Excel Users

R for Excel Users. Lowndes J and Horst A, 2020. Chapter 4 and 8 focus on starting setup and collaboration with GitHub and RStudio

`usethis`

usethis. Wickham H and Bryan J.

> usethis is a workflow R package: it automates repetitive tasks that arise during project setup and development, both for R packages and non-package projects.

Git Flight Rules

"A guide for astronauts (now, programmers using Git) about what to do when things go wrong."

https://github.com/k88hudson/git-flight-rules

This is quite a comprehensive and maintained list of git commands, using the command line.

GitHub: A beginner's guide to going back in time (aka fixing mistakes)

https://ohi-science.org/news/github-going-back-in-time

This has screenshots and a workflow with RStudio and the command line.

Practical R Workflow for Scientists

By Dr. Eli Holmes

https://rverse-tutorials.github.io/RWorkflow-NWFSC-2022/week1.html

Oh Sh*t, Git!?!

Oh Sh*t, Git!?!. Swear-free version: Dangit, Git!?!

> Git is hard: screwing up is easy, and figuring out how to fix your mistakes is [nearly] impossible. Git documentation has this chicken and egg problem where you can't search for how to get yourself out of a mess, unless you already know the name of the thing you need to know about in order to fix your problem.
>
> So here are some bad situations I've gotten myself into, and how I eventually got myself out of them in plain english.

Not just for programmers

Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution Braga et al 2023 provides 12 ways researchers in ecology and evolution use GitHub, as well as resources and examples of researchers using and learning GitHub.

```
Warning in readLines(file): incomplete final line found on
'how-do-i/refactor-modularize-code.qmd'
```

Refactor code

Shifting to collaborative and portable code often means rewriting code, not to change what it does, but how it does it. This is often called "refactoring", "modularlizing" or "mapping".

Refactoring can include better commenting and documentation and removing hard-coded filepaths. It can also mean moving away from long for-loops and towards smaller functions that are called to repeat operations. Refactoring makes your code more portable between people and computers. It can also make it faster, if you refactor so that pipelines can be run in parallel rather than sequentially.

Code Smells & Feels

Jenny Bryan's 2018 UseR! Keynote "Code Smells & Feels"

slides and video.

Abstract:

> "Code smell" is an evocative term for that vague feeling of unease we get when reading certain bits of code. It's not necessarily wrong, but neither is it obviously correct. We may be reluctant to work on such code, because past experience suggests it's going to be fiddly and bug-prone. In contrast, there's another type of code that just feels good to read and work on. What's the difference? If we can be more precise about code smells and feels, we can be intentional about writing code that is easier and more pleasant to work on. I've been fortunate to spend the last couple years embedded in a group of developers working on the tidyverse and r-lib packages. Based on this experience, I'll talk about specific code smells and deodorizing strategies for R.

Joy of Functional Programming

Hadley Wickham's 2019 ACM talk "Joy of Functional Programming for Data Science"

slides and video

Abstract:

> Functional programming (FP) provides a rich set of tools for reducing duplication in your code. The goal of FP is to make it easy to express repeated actions using high-level verbs. I think that learning a little about FP is really important for data scientists, because it's a really good fit for many problems that you'll encounter in practice. In this talk, I'll introduce you to the basics of functional programming in R, using the purrr package. I'll begin by briefly dissecting the for loop that you're already familiar with, then continue to show why functional programming provides elegant alternatives. I'll next dive into two examples showing where FP is particularly useful in data science: when ingesting unruly datasets spread across multiple files, and producing multiple reports for different stakeholders. You'll get the most out of this talk if you're familiar with R, or you've done data science in other languages like Python.

```
Warning in readLines(file): incomplete final line found on
'how-do-i/create-publish-quarto.qmd'
```

Start with Quarto

We've created a tutorial on how to copy (fork or download) an existing Quarto website like this one and adapt it for your own uses. Please see: https://openscapes.github.io/quarto-website-tutorial/.

Ask questions

Where do I go when I need to get unblocked? Where do I go when I want to learn how to unblock myself? Here are some resources. Most examples are with R, but the principles of how and where to ask questions apply broadly.

How to get help in R

How to get help in R, a presentation for R-Ladies Coventry, by Ella Kaye, 2021.

In the session, Ella will talk us through different ways to get help when you get stuck on a problem in R. We'll cover how and where to look for answers and how to ask and post good questions that make it easy for others to help you, including a demo of the excellent reprex and datapasta packages. So whether you're an R novice or more experienced R user, hopefully there'll be some useful tips for you!

How to ask questions so they get answered! Possibly by yourself!

How to ask questions so they get answered! Possibly by yourself! 2017, an rOpenSci Community Call with Jenny Bryan, JD Long, Scott Chamberlain.

Where to get help with your R question?

Where to get help with your R question?, Maëlle Salmon blog post 2018 (good general approach)

The reprex R package

The reprex R package. What is a reprex? It's a **rep**roducible **ex**ample. This package helps you prepare reprexes for posting to GitHub issues, StackOverflow, in Slack messages or snippets, or even to paste into PowerPoint or Keynote slides.

Teach me how to Google

Teach me how to Google by Sam Csik of the National Center for Ecological Analysis & Synthesis and RLadies Santa Barbara.

- Original talk developed for the Master of Environmental Data Science Program (MEDS) at UC Santa Barbara's Bren School of Environmental Science and Management.

Discuss community

Conversations about community building and open science is really important for getting support and supporting others with this work.

Community building

How do you talk to others (managers, supervisors) about the importance of community building and building social infrastructure? Here are some starting resources for reading about community building with Openscapes.

They are in order of article length and increasing specificity:

- Shifting institutional culture to develop climate solutions with Open Science (Lowndes et al 2024) - An Open Access Editorial written collaboratively by Mentors across the Openscapes community
- Supercharge your research: a ten-week plan for open data science (Lowndes et al 2019) - community building within teams, based on our flagship program Openscapes Champions
  - These slides - Documenting things: openly for future us share what's possible from focusing on documentation and community
- The Openscapes Flywheel: A framework for managers to facilitate and scale inclusive Open science practices (Robinson & Lowndes 2022) - community building within NASA with a framework we developed and refine.
  - These slides from an internal NASA meeting in March 2023 (linked from openscapes.org/media) tell some of this story

Reuse Openscapes Approach

We designed, facilitate, and teach building from what has worked in many other programs and resources. In the open science world, this is often called "reuse and remix" - reusing parts and tailoring and building them out for your needs. We reuse/remix heavily from Mozilla Open Leaders, RStudio, RLadies, rOpenSci, and many others. It's exciting to see these ideas spread! See the Forked Lessons chapter for inspiring examples of the Openscapes community making real change in their teams and organizations.

The Openscapes Approach Guide is where we're starting to document our approach so that others can lead their own Champions Cohorts. Even if you're not leading your own Cohort, there are many things we do that you can reuse. For example:

- Have Google Doc agenda with live notetaking during your meetings (if this is a recurring meeting, organize it in a shared folder, either with one doc per meeting like our Cohort Call Agendas or many meetings in the same doc with the most recent at the top, like our Co-Working notes)
- Begin a meeting with a Roll Call/Check in and Code of Conduct
- Share slides before giving your talk so that folks can see them in high resolution and click on links
- Teach using other people's slides

6 Inspiration

Resources that influence us

Some resources that influence our thinking.

Talking about data science: Hilary Parker & Roger Peng

RStudio::conf(2020) keynote & NSSD podcast episode 100

> If you want to write, you read a lot, music, you listen a lot. I'ts hard to do this with data analysis.

Opinionated analysis development

Parker 2017, PeerJ

Principles for data analysis workflows

Stoudt, Vásquez, & Martinez, 2021

> workflow describes what a researcher does to make advances on scientific questions: developing hypotheses, wrangling data, writing code, and interpreting results. Workflow: The process that moves a scientific investigation from raw data to coherent research question to insightful contribution. This often involves a complex series of pro- cesses and includes a mixture of machine automation and human intervention. It is a nonlinear and iterative exercise.
>
> Importantly, the difficulties we encounter in this [Explore] phase help us build empathy for an eventual audience beyond ourselves. It is here that we experience firsthand the challenges of processing our data set, framing domain research questions appro- priate to it, and structuring the beginnings of a workflow. Documenting our trial and error helps our own work stay on track in addition to assisting future researchers facing similar challenges.

Vulnerability: Brené Brown

Power of Vulnerability: TED Talk

Dare To Lead Podcast

Flywheel & Hedgehog concepts: Jim Collins

Flywheel concept

Hedgehog concept

All We Can Save: Ayana Johnson & Katharine Wilkinson

All We Can Save Project

> There is a renaissance blooming in the climate movement: leaderhip that is more characteristically feminine and more faithfully feminist, rooted in ompassion, connection, creativity, and collaboration. …To change everything, we need everyone — All We Can Save

> Johnson's frustration with the climate movement isn't about the current leaders doing a bad job—it's just that we need more leaders. Her vision of the world includes people from every community in climate leadership roles. — The Marine Biologist Building an Inclusive Climate Movement, Vice

> All We Can Save is basically a community bound between two covers, and a gift to any who wishes to join in. - Eric Roston, Bloomberg

The Power of Welcome

The Value of Welcome — Stef Butland, rOpenSci

The moment of lift: Melinda Gates

Architecture of Participation: Tim O'Reilly

It's Not About You: The Truth About Social Media Marketing (2012). Strategy on community building through modern channels

> "We tell big stories that matter to a community of users, and together we use those stories to amplify a message that we all care about…And once they start telling their story as part of the bigger story, it suddenly looks like a parade."

Open source paradigm shift

> I've come to use the term "the architecture of participation" to describe the nature of systems that are designed for user contribution.

Systems Change: Donella Meadows

Leverage points: places to intervene in a system: (in increasing order of effectiveness, 12 -> 1)

12. Constants, parameters, numbers (such as subsidies, taxes, standards).
13. The sizes of buffers and other stabilizing stocks, relative to their flows.
14. The structure of material stocks and flows (such as transport networks, population age structures).
15. The lengths of delays, relative to the rate of system change.
16. The strength of negative feedback loops, relative to the impacts they are trying to correct against.
17. The gain around driving positive feedback loops.
18. The structure of information flows (who does and does not have access to information).
19. The rules of the system (such as incentives, punishments, constraints).
20. The power to add, change, evolve, or self-organize system structure.
21. The goals of the system.
22. The mindset or paradigm out of which the system — its goals, structure, rules, delays, parameters — arises.
23. The power to transcend paradigms.

> So how do you change paradigms? Thomas Kuhn, who wrote the seminal book about the great paradigm shifts of science,7 has a lot to say about that. In a nutshell, you keep pointing at the anomalies and failures in the old paradigm, you keep coming yourself, and loudly and with assurance from the new one, you insert people with the new paradigm in places of public visibility and power. You don't waste time with reactionaries; rather you work with active change agents and with the vast middle ground of people who are open-minded.

Organizational architecture

Cautionary Tales Podcast Ep 6 – How Britain Invented, Then Ignored, Blitzkrieg.

This tale is about how the organizational architecture of existing entities - whether the British army, Sony, Kodak, or Xerox - cannot always support their own innovation because of the social structures they were built upon. Fascinating to think about in terms of how open science has not been embraced by scientific communities within the existing academic structure.

Disruption can feed creativity

Cautionary Tales Podcast Ep 7 – Bowie, jazz, and the unplayable piano.

This tale is about music: how Keith Jarrett reluctantly played on a broken piano and how David Bowie and Brian Eno's take on collaboration led to brand new sounds and ideas. I

think about this for science and openness - working out of your comfort zones and mixing up how you do it and who you do it with.

Kaitlyn Thaney

Funding research infrastructure

> there's also the fact that the current funding model has led to a perceived sense of scarcity, pushing open projects to compete rather than collaborate, to build new features instead of maintaining their work and deepening their level of service for their communities. An additional dimension to our work involves looking at the staffing and human infrastructure powering open technology development, maintenance, governance and stewardship. That volunteer labor and community engagement is often an invisible cost we gloss over in our estimations and recommendations, while also being a core pillar in this work.

Mentorship vs Sponsorship

https://larahogan.me/sponsors/

Practical Computing for Biologists

Practical Computing for Biologists

Introduction to the Terminal/command line, introduction to regular expressions. Chapter 2 alone is incredibly powerful for building immediate cross-software skills and thinking differently (Disclosure: Lowndes was an alpha reviewer in grad school).

Virtual meetings

- https://opensource.com/article/20/6/remote-meetings
- https://aspirationtech.org/blog/virtualmeetingpowerdynamics