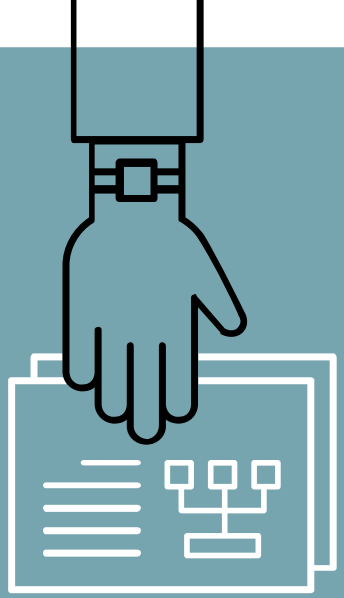
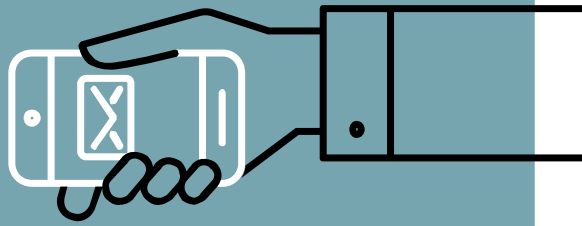
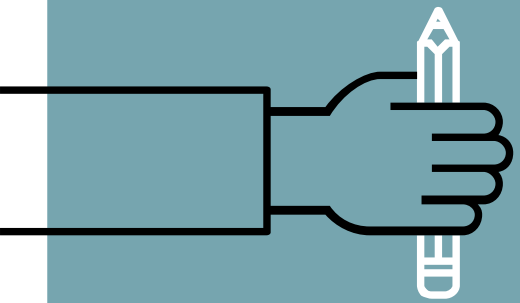
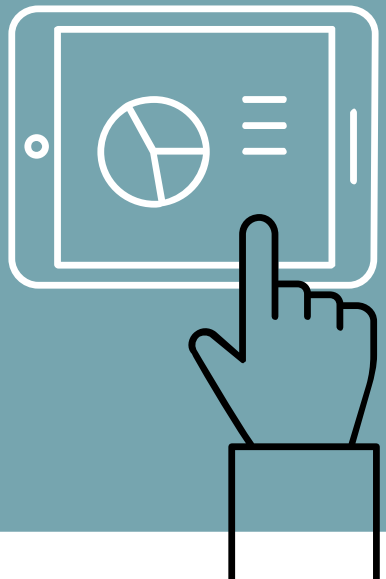




BST 270
Reproducible Data Science

Winter 2022
Session 8





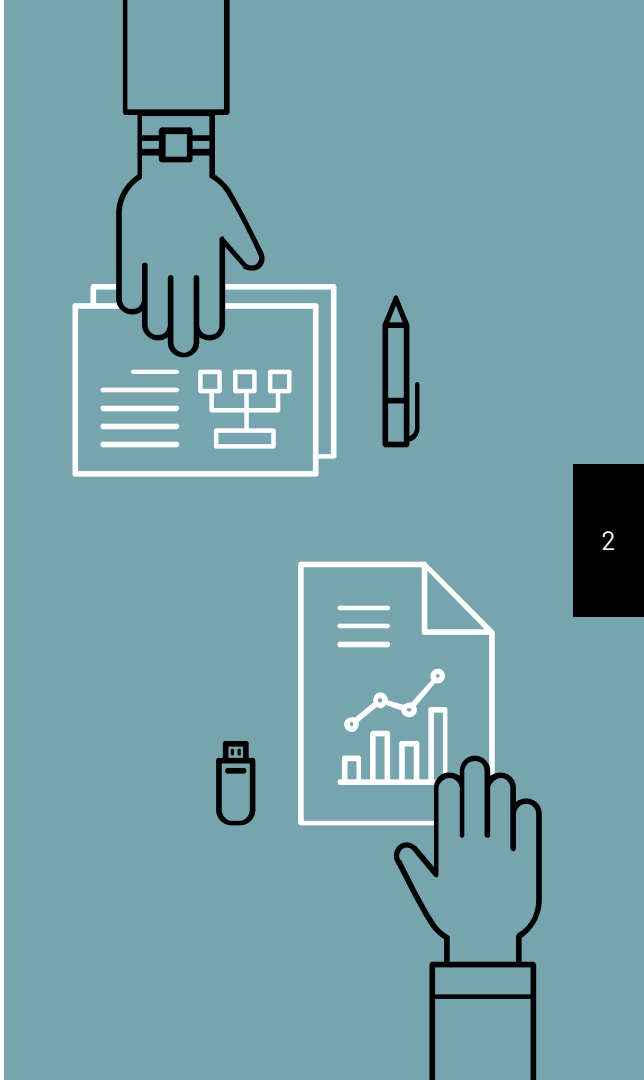
Module 6 Part 3 Comments

The "auto-updating" feature of Code Ocean really does alleviate a lot of stress I have with saving the correct versions of code. Many times I have overwritten working code, and forgotten how to retrieve the original version.

One thing I think is really nice about Code Ocean is the 'freezed version' of the Docker container when a paper is published, you only have to refer to that version if you're looking for something specific from the publication rather than looking through an 'updated' version that could have more data files, thousands of lines of extra code, etc.

It would be great if Code Ocean could collaborate with top statistics journals such as AOS, JRSSB so that those cool methods can be widely used by public.

If journals require code to be made public through platforms such as Code Ocean, we might see less unreproducible results since they cannot falsify results when the public can test and run their code.



Module 6 Part 3 Comments

Simon Adar talked about the time-consuming task of learning somebody's code only to realize that you don't need it after all. This is why I like the really really simple toy examples that the R help pages give. Sometimes you need to go out of your way to use this code (just to get it into your vocabulary), but it will pay off when you can tackle more complex tasks using code that you forced yourself to practice. Changing coding habits is hard!!!

Simon Adar's discussion of Jupyter notebooks was interesting. I have found that Jupyter notebooks don't work as well with version control systems (i.e. Git) since they aren't exactly plain text files, as opposed to something like R Markdown or Org.

I've been working off of someone’s GitHub repo recently and one thing I wanted to point out is the importance of vignettes that demonstrate all the of capabilities of the package/analysis. It can be frustrating (and what is happening in my case) when the vignette doesn't seem to work as it is supposed to, because that quickly can turn you off of using the function.

Module 6 Part 3 Comments

Code Ocean is amazing! Putting everything - code, data, and documents - on the cloud would definitely be the best solution for every researcher. It could be even better if it can have any version control function.

There are so many useful tools to learn for reproducible data science (Make, git, Rmarkdown, code ocean, etc...) that it's honestly a bit overwhelming to try to learn all at once. I just need to try to learn one at a time!

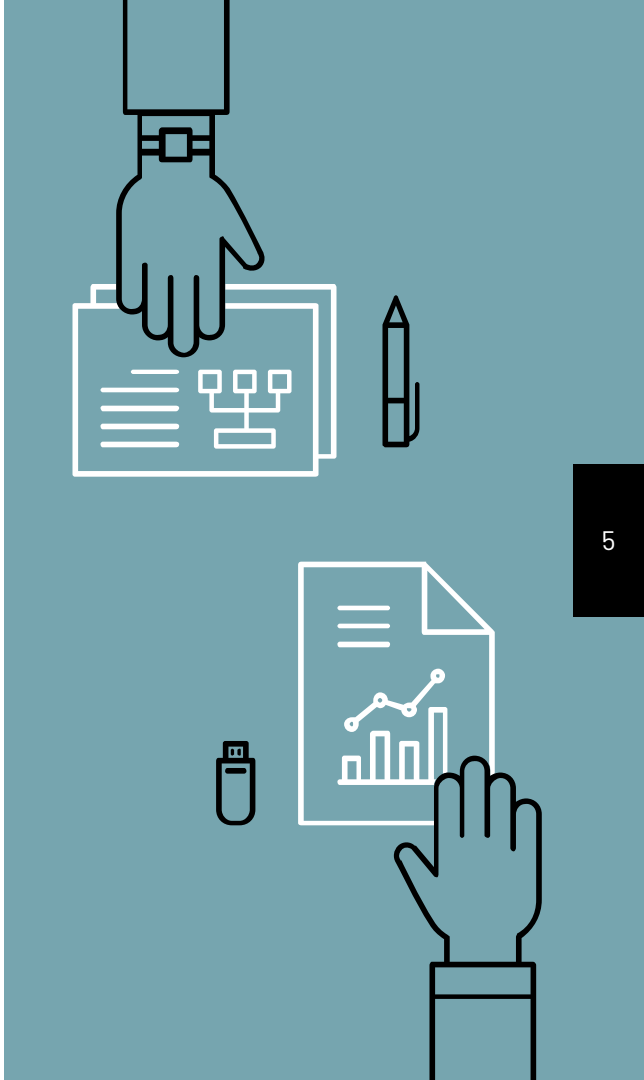
I think the idea of Code Ocean that each time a code is run to generate some results, the code itself will be stored in the log along with the result is brilliant. It reduces manual errors or overlooking in terms of keeping track of research codes.

Module 6 Part 3 Comments

It is interesting to hear about Dr. Adar's experience and how he was motivated to begin his start up Code Ocean. I also had similar experience, where in one of our statistical genetics studies, we tried to use a tool published by others. Though we were able to access their code on GitHub, we had trouble running them as expected because our laptop was set up differently than theirs. So it will be great as Code Ocean prompts users to not only upload their code but also a list of dependencies for the code to work properly.

It is a surprise for me to see so many aspects including in the industry area have made contributions to Reproducible Science.

Sample data is very important for me link to my own data. If only the code, sometimes it is hard to use and read them very quick.

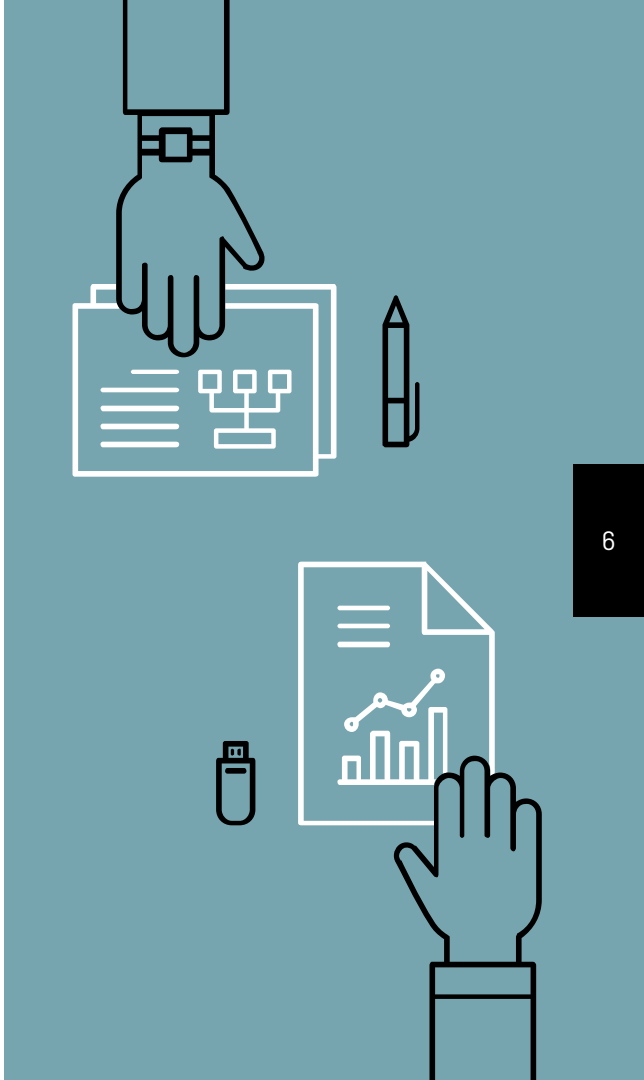


Module 6 Part 3 Comments

GitHub will send you notifications that your dependencies are vulnerable or out of date, as long as you have a manifest file in your repository (like package.json for Javascript or requirements.txt for Python).

It seems that what reproducibility comes down to is documentation. Even if there's a roundabout way to access the data, multiple steps to process it, and a ton of software to install before getting started, it can be reproduced as long as each step is carefully documented. The rest of what we've learned in these modules is ways to make it quicker/easier for yourself in the future or for the next researcher, but the most baseline requirement should be documentation.

I enjoyed the conversation about the limitations of Jupyter Notebooks because I used to think that they were almost a perfect tool, but I now realize that when you're developing tools for a wide range of complicated workflows there are many other things to consider.

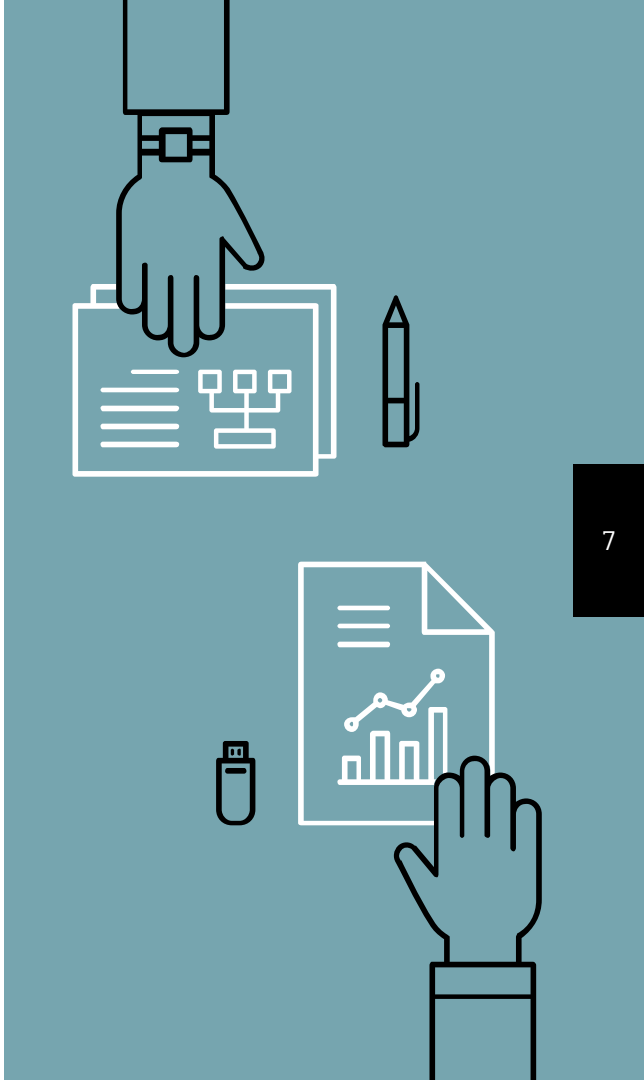


Module 6 Part 3 Comments

I am surprised that I have never heard of Code Ocean before. The demo by Dr Adar is very helpful. This looks like a very powerful tools that can interact with users, which is a feature that GitHub does not have, and also allow users to check the input and output of the algorithm, which paper text cannot describe very thoroughly.

Code ocean is a user friendly interface and allows researchers with different programming experiences (in different languages) to collaborate and publish their code and research. This platform can be used to standardize research workflow, track code and reproduce research.

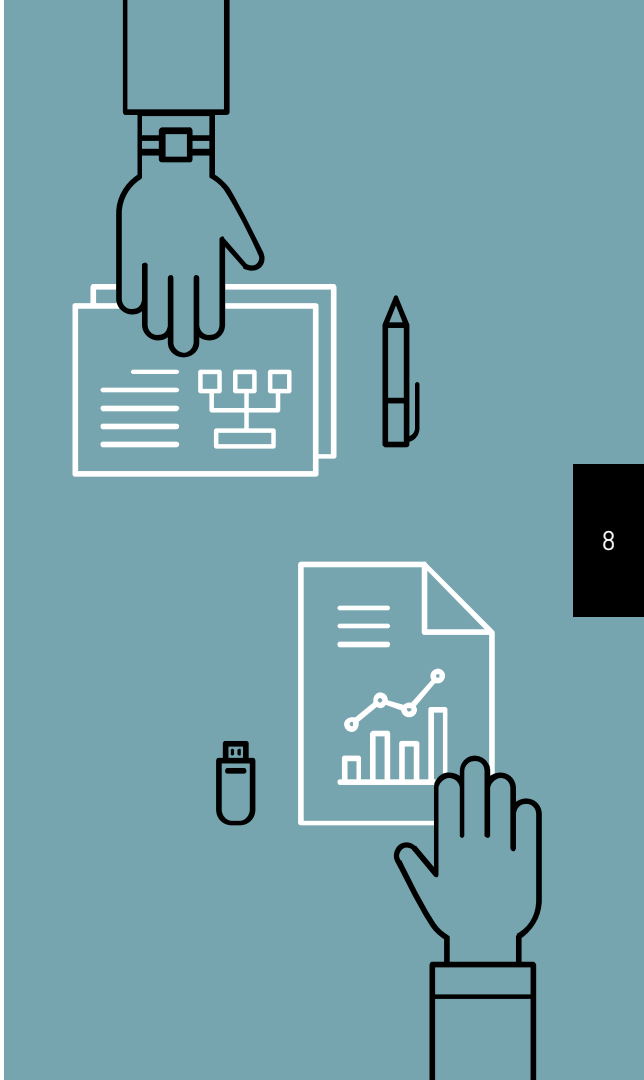
I love how Code Ocean is set up to show the entire analysis process for papers, allowing the authors to explain the entire workflow and what each file does (I thought this README file was a particularly good example: https://codeocean.com/capsule/1887579/tree/v3). I particularly like the way that they set up the file structure on the left, so you can expand or collapse it as you want (one of my pet peeves about GitHub is that there's no easy way to view the entire file structure all at once - you have to dive into each folder).



Module 6 Part 3 Comments

I hope that the push to publicize more code also leads people to review each other's code more - it's hard to hide sloppy code if you know you have to publish it for the world to see!

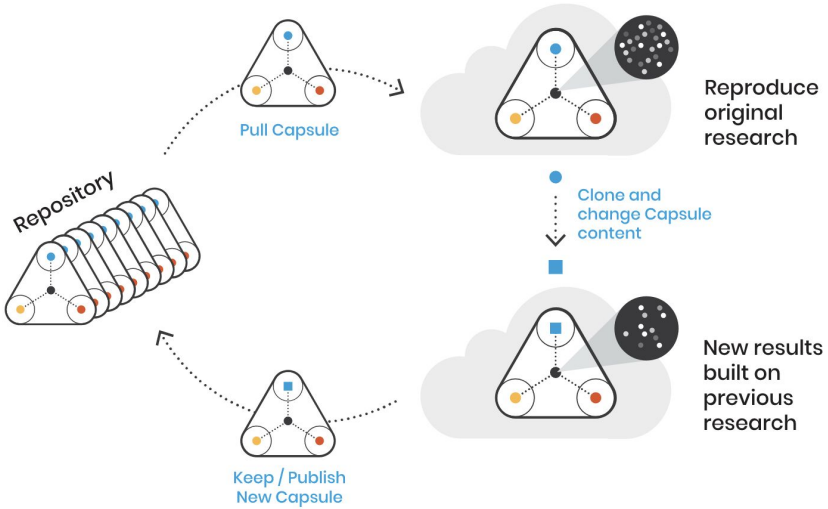
I wonder what companies think about Code Ocean and other movements to publish analysis code. So much code in industry is 'proprietary-lite' (not actually protected IP, but treated as company property) that I feel like there would be pushback from being forced to share code that could be accessed by other industry competitors.

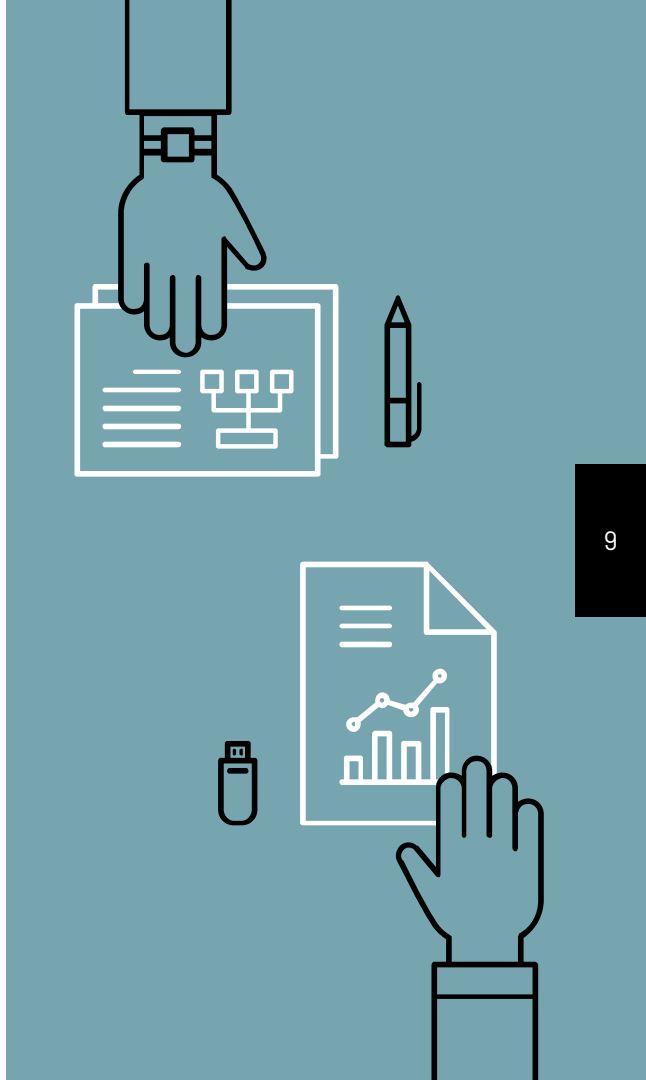


Module 6 Part 3 Discussion

What exactly do people use Code Ocean for?

- Code Ocean is a centralized platform for the creation, sharing, publication, preservation and reuse of executable code and data.





Module 6 Part 3 Discussion

So, if I'm understanding correctly, the key contribution of Code Ocean (as opposed to GitHub) is to have frozen versions of code, with all the dependencies available, so people can run your code in its final version and get a sense of what it does. Am I missing anything?

Is Code Ocean almost like an interactive GitHub, where it goes beyond just file storage?

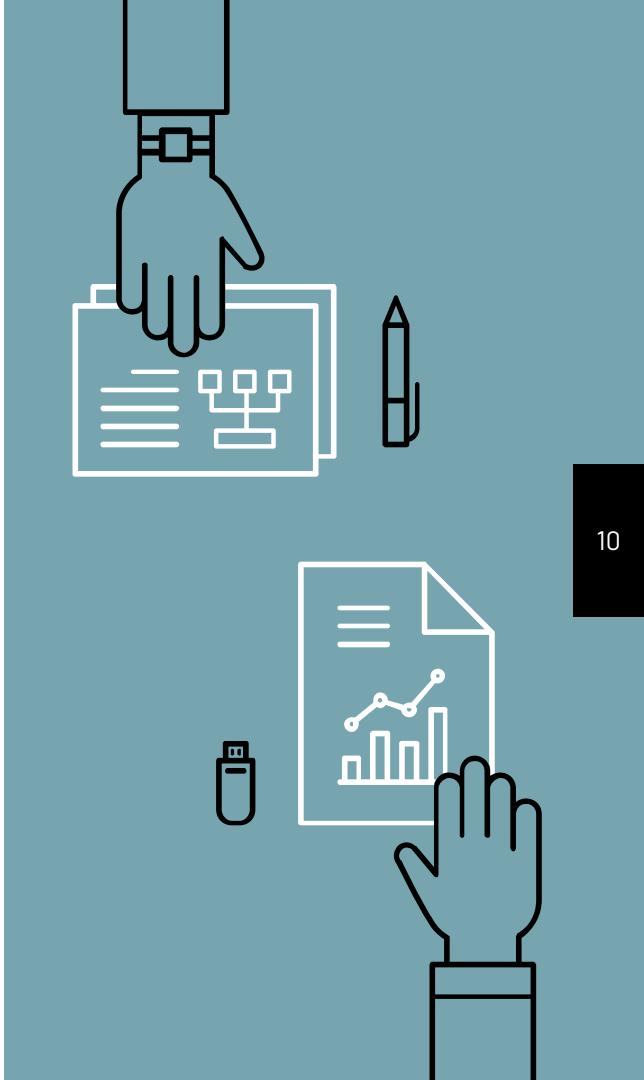
- All of that plus it makes collaborating with others on code a lot easier because you can all access the same computing environment and not have to worry that collaborators working on different computers have different environments.

What happens on Code Ocean if your code takes a lot of resources (i.e. memory and time) to run?

- Everything is run using AWS cloud resources so I don't think it will make a difference, but I don't know the nuances of their pricing tiers - they may charge for code that takes a really long time to run.

The pricing structure of Code Ocean seems to make sense. What are your thoughts on it?

- For academia it sounds nice. I think it would probably take at least a month to get used to the platform so I would work on a project first and then upload everything to Code Ocean and then publish. After that it would probably make sense to work on everything using Code Ocean, or at least running the same version on your personal computer and Code Ocean.



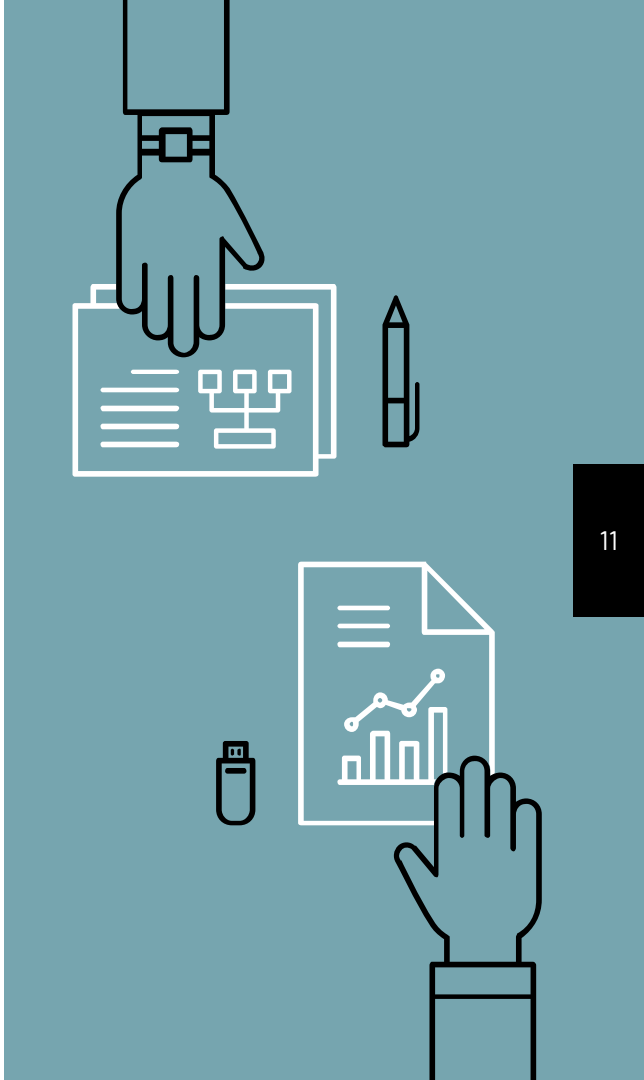
Module 6 Part 3 Discussion

Code ocean seems to be a really great unifying framework for reproducible research results. Negative seems to be for bigger experiments, I'm not sure that everyone would have the access without paying to run intensive deep learning models?

- I found this answer in the FAQs page. Briefly, you can avoid it using the tips suggested in that answer, but this may end up being a drawback.

Another negative seems to be that some would only be able to work on secure servers, or just servers in general. What are the barriers to using this on those kinds of environments in Code Ocean? (may have missed this). Ideally everyone would be using something like Code Ocean for their research.

- Great question. At this point, I'm not sure. I did some searching and I couldn't find an answer. If I find something I'll let you know

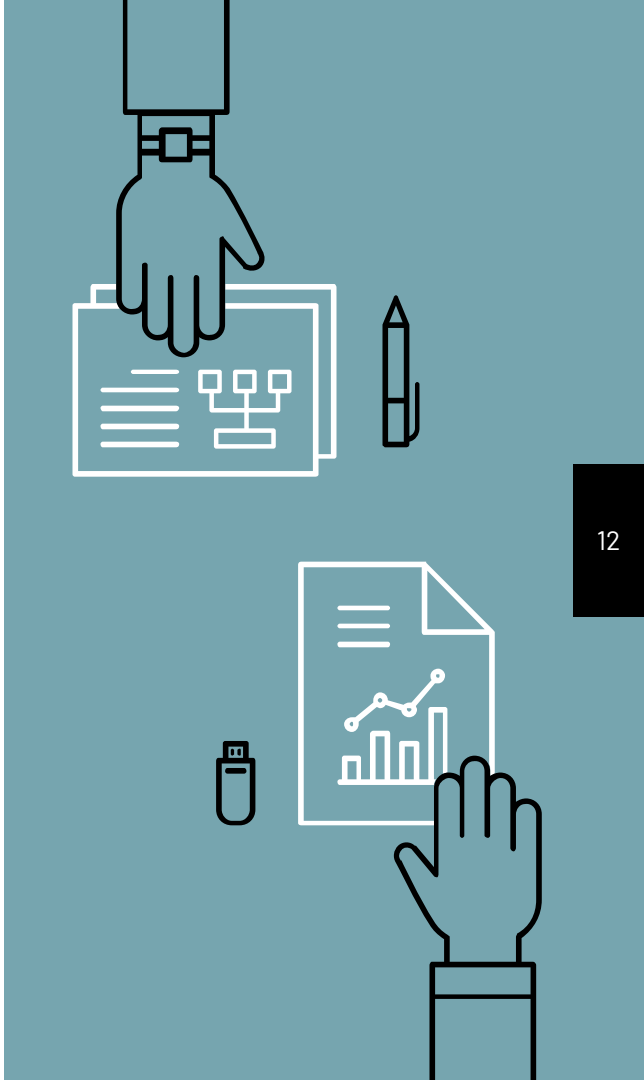


Module 6 Part 3 Discussion

What (if there is one) is the consensus within the academic community on Code Ocean — is it something you could see being widely used in the next 10 years or so?
Do you think we will see journal's adopt code ocean (or their own software) as part of the submission process? This seems like a great move in the right direction for reproducibility, but I wonder how accessible these resources are.
The Code Ocean demonstration was amazing, and it seems that big journals are using it to an extent, so can we expect this to be the norm for all published research in the future?
I wonder is Code Ocean an increasingly popular platform? Are there a lot of publications and research use this platform for code sharing? It seems to be very useful for reproducibility of research, but I am surprised that I never heard about it before; the code sharing platform I encountered in all publications is GitHub. I believe there are a lot of concerns for people who decide whether to use it, including the sensitivity of data and intellectual property concerns.

- I don't know if there is one yet - I personally haven't met many academics who use it yet (to my knowledge). I do think it will be used more widely in the next several years. Cambridge University Press, Taylor and Francis, and academic journals have partnered with Code Ocean since the recording of the videos. Here's a quote from a paper:

  Recently, several *Nature* journals, including *Nature Methods*, *Nature Biotechnology*, and *Nature Machine Intelligence* completed a trial with Code Ocean, a cloud-based reproducibility platform that aims to help reviewers and authors facilitate peer review of source code. This trial has since been made permanent and expanded to other journals at Springer Nature, including *BMC Bioinformatics* and *Scientific Data*. *Genome Biology* is excited to announce that at the end of 2020, we also have partnered with Code Ocean, with the aim of making the peer review of source code easier for both reviewers and authors.



Module 6 Part 3 Discussion

I'm amazed at how fast the Code Ocean platform works. Is running code on the cloud faster in general than running on local systems?

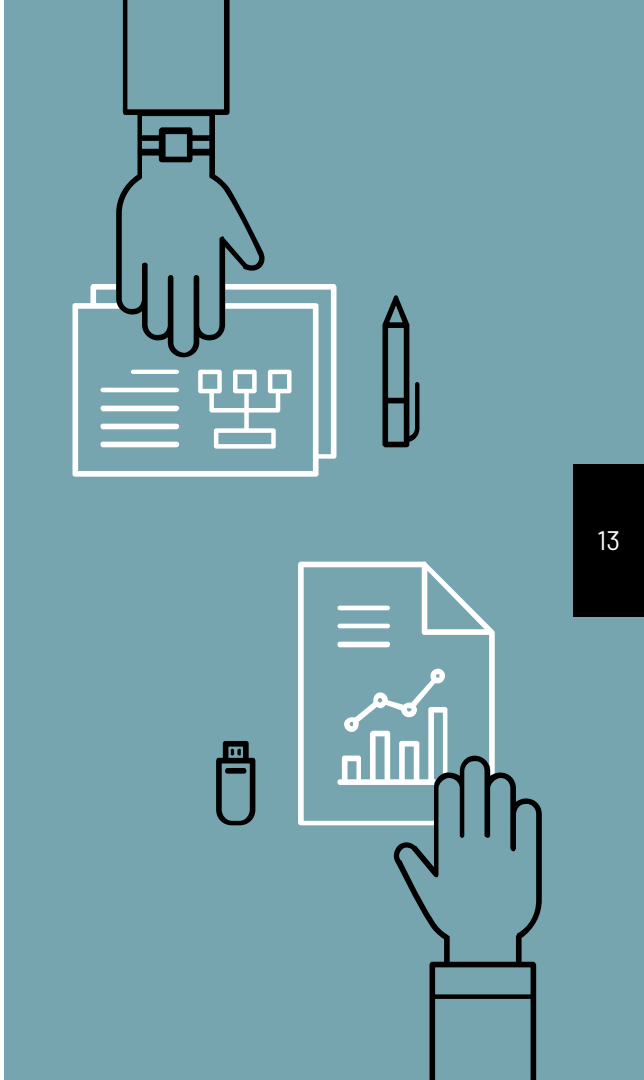
- So. Much. Faster. Unless you are using a local system with a lot of computing resources.

Do many professors at HSPH use Code Ocean?

- Not that I know of, but I haven't talked to every faculty member. I know JP Onnela uses AWS and I think JQ uses GCP. Curtis might use Code Ocean but I'm not sure. I'm sure some at HSPH do.

How much do people use Code Ocean? It seems useful but fairly new.

- I would check out all of the documentation and videos and information available on their website. Code Ocean has come a long way since the videos were recorded.



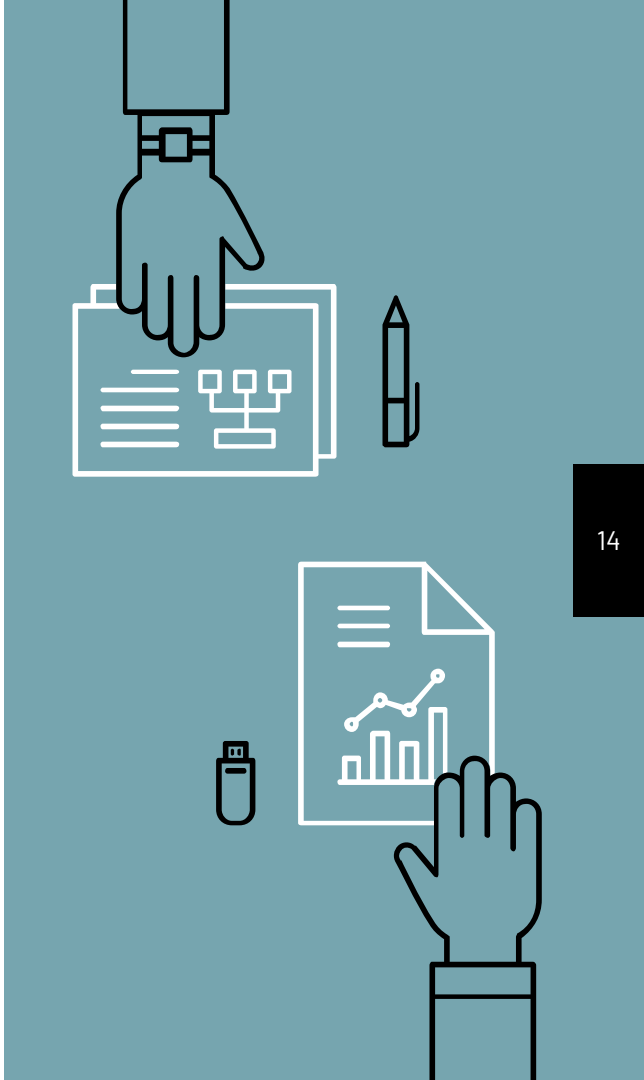
Module 6 Part 3 Discussion

In video titled, "A conversation with Simon Adar part 6", What exactly is a docker image? Is it a virtual machine which has a specified operating system, technology (R, python, etc.) along with the packages installed, which anyone can use from anywhere?

- It is a virtual environment which has a specified operating system, technology (R, python, etc.) along with the packages installed, which anyone can use from anywhere. The virtual machine to use the image may change.

Using docker is like packing all your codes and data into one application, which can be run by anyone. Since today more and more researchers like publishing interactive applications or interactive websites, docker seems to be the second choice. How would you like to compare between packing everything into a docker and building up your own software/app?

- Good question. I personally don't have experience building my own app (unless we're talking about a Shiny app). I guess it would come down to preference and ability to create your own.



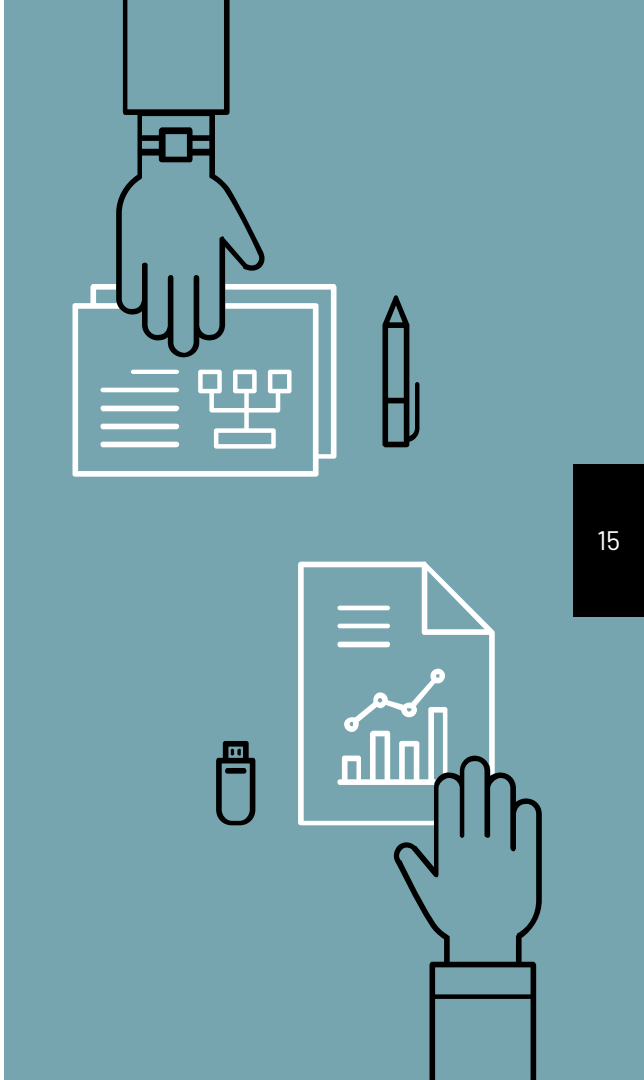
Module 6 Part 3 Discussion

Can Code Ocean support computationally expensive algorithms (say, requiring several hours on a compute cluster)? It seems that all of the examples he discussed were ones that can be run locally.

- Yep!

Simon mentioned that Code Ocean has been collaborating with IEEE so that data and papers are published simultaneously. However, biological/health science data can be very different from data in electrical engineering studies. We especially need to take the privacy and sensitivity of biological data into account before we just simply let the data go public. I'm really curious if Code Ocean has any accommodations that address this issue.

- I did a lot of searching and couldn't find anything. This may be something a researcher would have to chat with the Code Ocean customer service team about first.



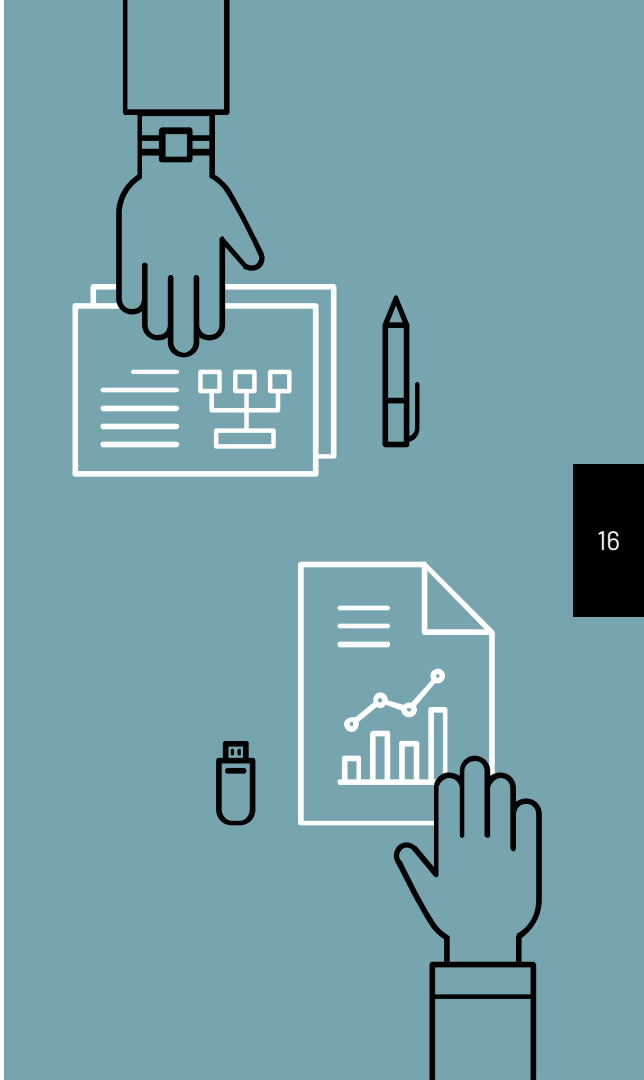
Module 6 Part 3 Discussion

In the video titled, "A conversation with Simon Adar part 6", Simon mentions of a collaboration between IEEE and code ocean. After the collaboration are the researchers required to make their code available via Code Ocean?

- I don't think every author is expected to make their code available. I think the collaboration is more for promoting researchers to use Code Ocean if they want to publish in IEEE, and having Code Ocean take into consideration suggestions made by IEEE editors/staff.

Do journals potentially lose any rights / intellectual property when you release your algorithms, software, and data before submitting to the journal, compared to if you just included these items in appendices/supplemental materials of your article?

- I don't think so. I think journals keep the rights/intellectual property of everything they publish, even if it was a preprint published somewhere else. Once they publish it, they own it.



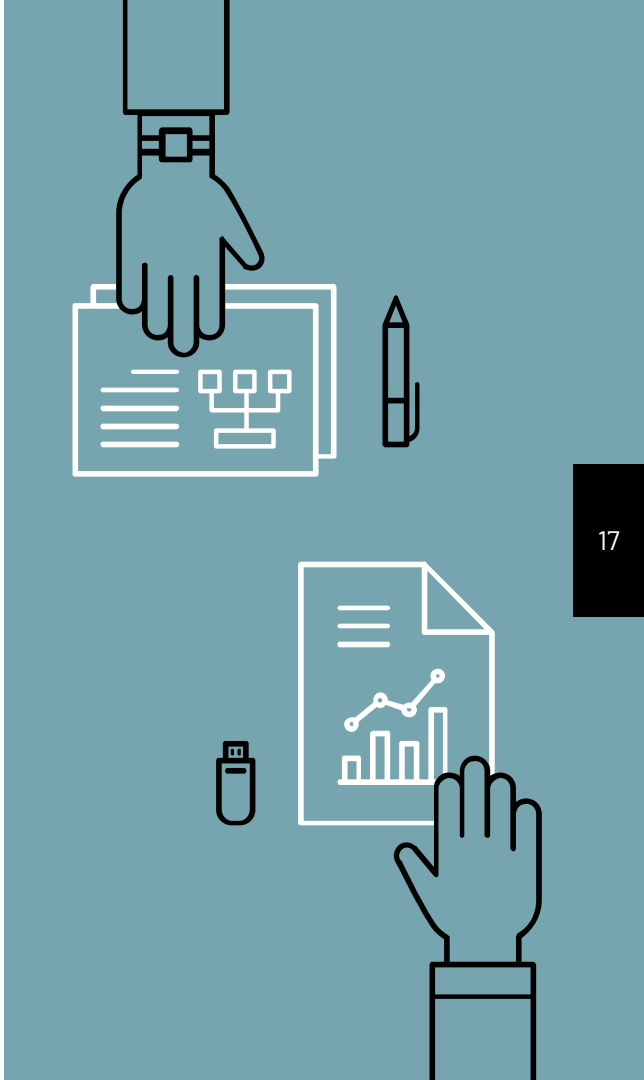
Module 6 Part 3 Discussion

Regarding workflows, I've also seen Snakemake used often in bioinformatics/computational biology - do you know of any good resources for learning Snakemake?

- I'd start with their slides and checking out their documentation. Here's a (fairly long but detailed) YouTube tutorial.

I know a lot of research groups work on the FAS RC cluster. Is the FAS RC virtual desktop equivalent to something like Code Ocean?

- I don't think so. I think it is only for running code/jobs and doesn't have any tools for documentation or workflows. I also don't think you can collaborate with anyone using the same virtual desktop.



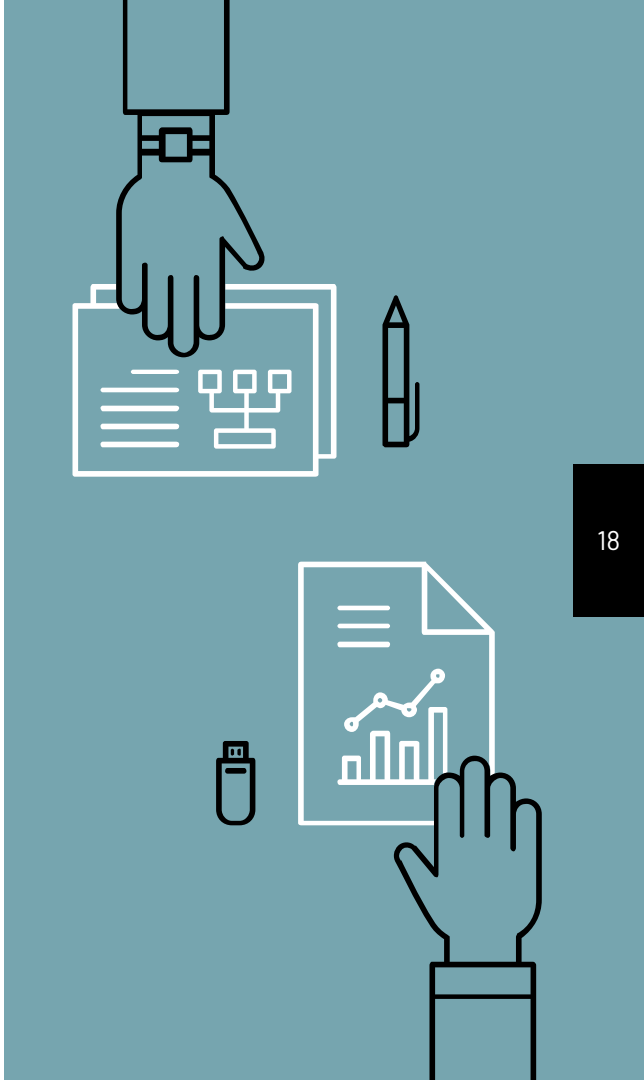
Module 6 Part 3 Discussion

One of my favorite RMarkdown features is that it detects uninstalled but necessary libraries when you open code and asks you up front if you want to install them. Does Jupyter notebook have something similar?

- I don't think so. I haven't been able to find anything about it being able to do this.

Simon Adar mentions that in their current form, Jupyter Notebooks have a long way to go to being a standalone tool for reproducible research. Is this even attainable or, given the current utility of Jupyter Notebooks, desirable? Jupyter Notebooks already seem to be incredibly effective for niche things like package vignettes, interactive demonstrations etc.

- Attainable? Most likely. Desirable? Depends on the user, I think. Jupyter Notebooks have progressed a lot the last few years but they still have a long way to go. I don't know if enough people would be interested in them being a standalone tool for developers to make them into one.

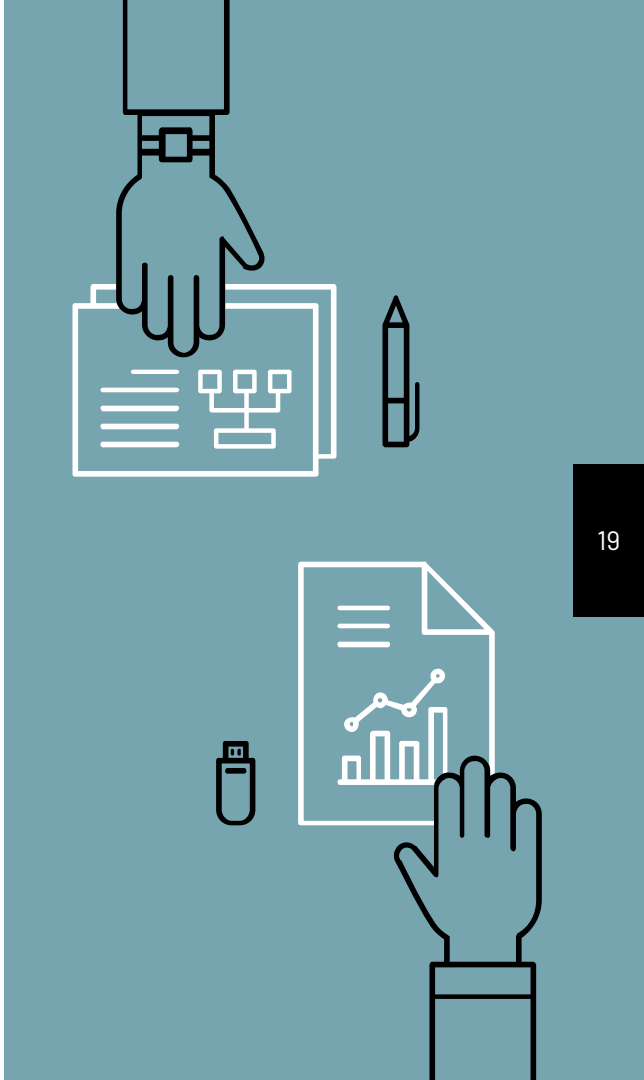


Module 6 Part 3 Discussion

Code Ocean is an exciting project with a lot of interesting features — the collaboration with IEEE is especially great. It seems like it's positioning itself as more of a substitute to GitHub than a complement. It's feasible that down the line we'll have access to numerous tools like GitHub, Code Ocean, etc that have similar functionality (with different relative merits). Could this be problematic (e.g. leading to researchers "picking teams" / making collaboration more difficult for people more well-versed in one tool than another), or should the availability of numerous (similar) tools be considered a good development?

Dr. Adar introduced that Code Ocean had this versioning function, so does that mean Code Ocean can be used as a replacement of GitHub? If so, what are some pros and cons of each tool and which one is easier to learn?

- I do think Code Ocean can be used as a replacement for GitHub, but that a lot of people are hesitant to switch over because you don't need to know much to get started with GitHub, i.e., the up-front time/effort cost is a lot less for GitHub. There are a TON of tools available now and it does make collaboration more difficult at times. The popularity of tools changes over time (e.g. SAS is dying out now) so I'm not sure what will happen in the future.



Module 6 Part 3 Discussion

What are tips and tricks to help my advisor follow more principles of reproducible research?

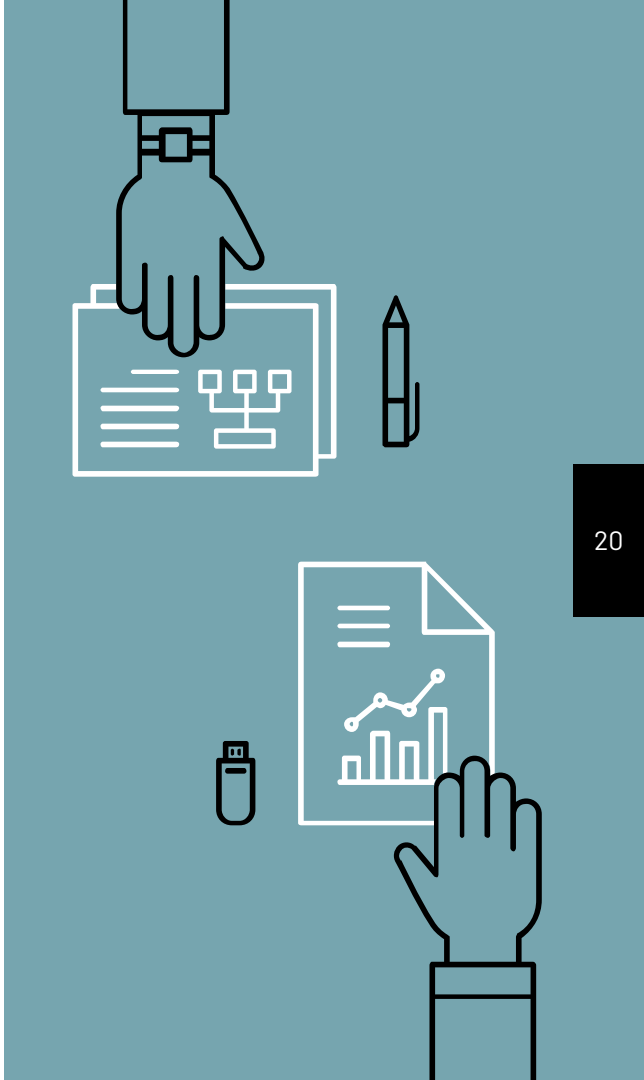
- Communicate (and demonstrate if possible) the importance of reproducible research and the amount of time saved. You can also mention what you learned in this course ;)

What do you think is the biggest practice in research that is common practice now and will no longer be common in the future due to fostering bad reproducible research habits?

- I think I'm being thrown off by the wording - I need more context.

Do you have a document with steps or any final suggestions for starting a project that can reproduced and with reusable code? I am going to start my Spring research and I really want to make sure I use all the resources we learned in class, and make sure my project is reproducible and can be applied to other data.

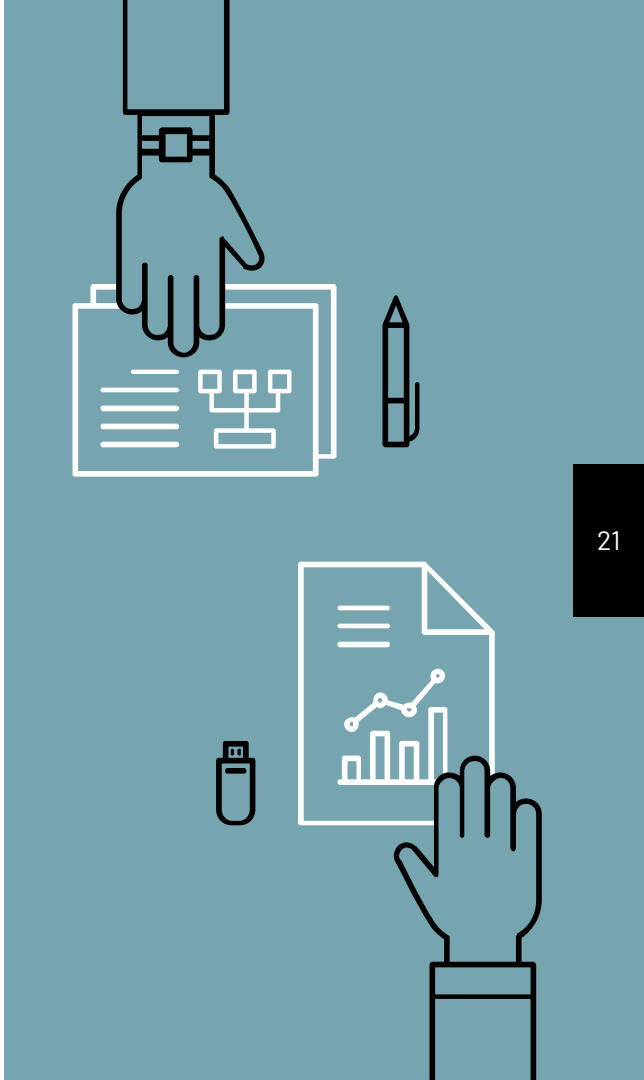
- Yes! There is a "Planning for Reproducible Science" tab on the edX course page. It contains a doc I wrote as a kind of checklist and Dos/Don'ts for reproducible research. I'll add the doc to our course GitHub repo and Canvas (under Session 8).



Course Summary

Reproducible Research DOs:

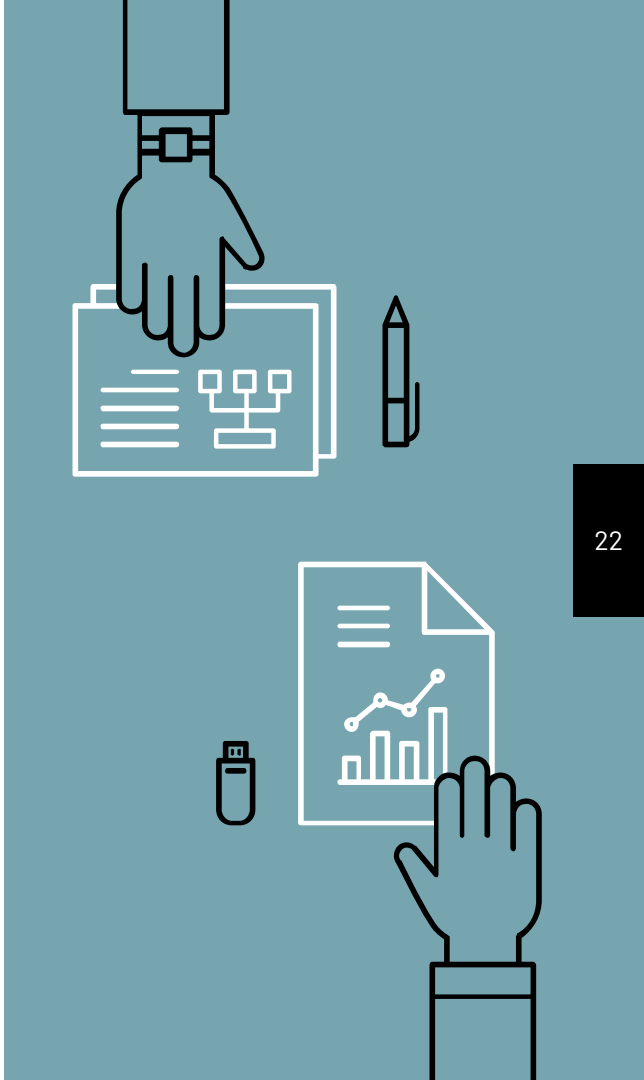
- Start with good science
  - Garbage in, garbage out
  - Coherent, focused questions simplify many problems
  - Working with good collaborators reinforces good practices
  - Something that's interesting to you will (hopefully) motivate good habits
- Use version control
  - Add changes in small chunks (don't just do one massive commit)
  - Track / tag snapshots; revert to old versions
  - Software like GitHub / BitBucket / SourceForge make it easy to publish results
- Teach a computer
  - If something needs to be done as part of your analysis / investigation, try to teach your computer to do it (even if you only need to do it once, like downloading a data set)
  - In order to give your computer instructions, you need to write down exactly what you mean to do and how it should be done
  - Teaching a computer almost guarantees reproducibility



Course Summary

Reproducible Research DOs:

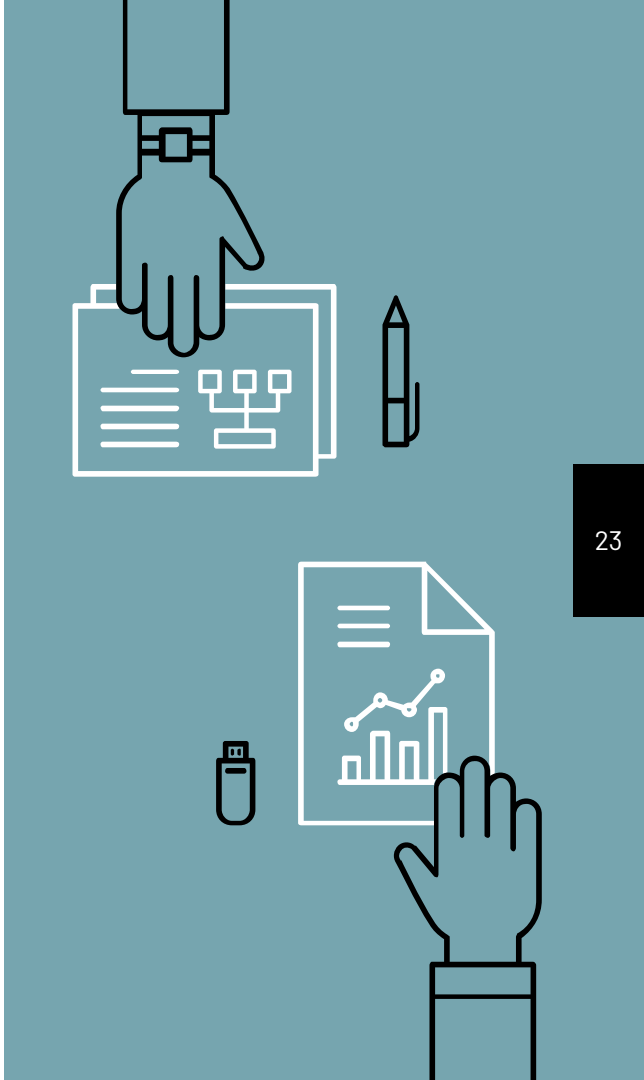
- Keep track of your software environment
  - If you work on a complex project involving many tools / datasets, the software and computing environment can be critical for reproducing your analysis
  - Computer architecture: CPU (Intel, AMD, ARM), GPUs
  - Operating system: Windows, Mac OS, Linux / Unix
  - Software toolchain: Compilers, interpreters, command shell, programming languages (C, Perl, Python, etc.), database backends, data analysis software
  - Supporting software / infrastructure: Libraries, R packages, dependencies
  - External dependencies: Web sites, data repositories, remote databases, software repositories
  - Version numbers: Ideally, for everything (if available)



Course Summary

Reproducible Research DOs:

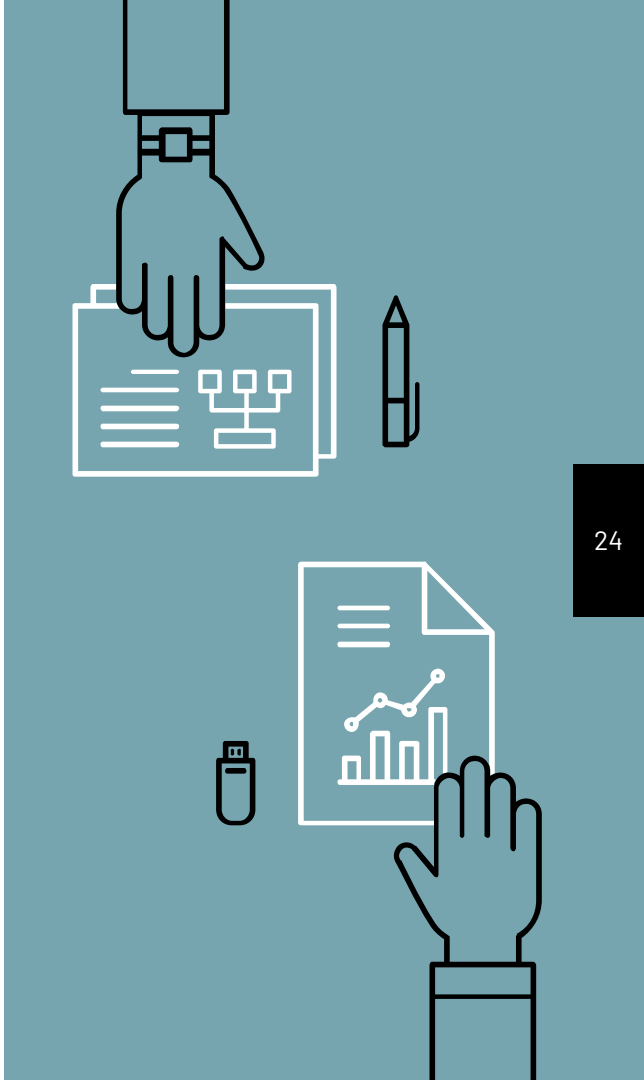
- Set your seed
  - Random number generators generate pseudo-random numbers based on an initial seed (usually a number or set of numbers)
    - In R you can use the set.seed()
  - Setting the seed allows for the stream of random numbers to be exactly reproducible
  - Whenever you generate random numbers for a non-trivial purpose, always set the seed
- Think about the entire pipeline
  - *Data analysis is a lengthy process; it is not just tables / figures / reports*
  - *Raw data → processed data → analysis → report*
  - *How you got the end is just as important as the end itself*
  - *The more of the data analysis pipeline you can make reproducible, the better for everyone*



Course Summary

Reproducible Research DON'Ts:

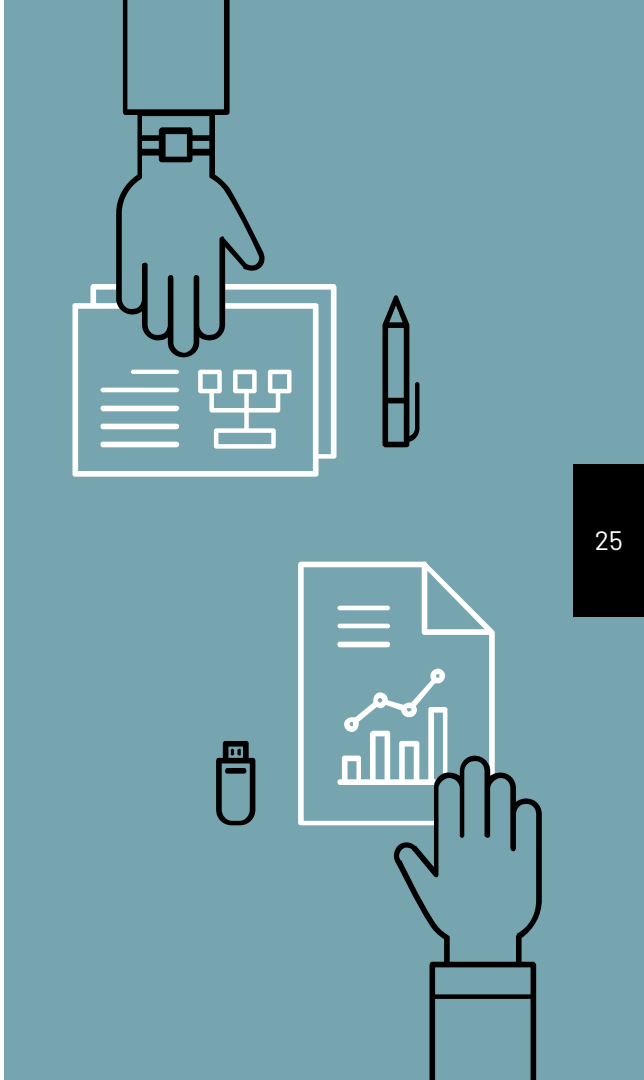
- Do things by hand
  - Editing spreadsheets of data to "clean it up"
- Removing outliers
- QA/QC
- Validating
  - Editing tables or figures (e.g. rounding, formatting)
  - Downloading data from a web site (clicking links in a web browser)
  - Moving data around your computer; splitting/reformatting data files
  - "We're just going to do this once …"
  - Things done by hand need to be precisely documented, and this is much harder than it sounds



Course Summary

Reproducible Research DON'Ts:

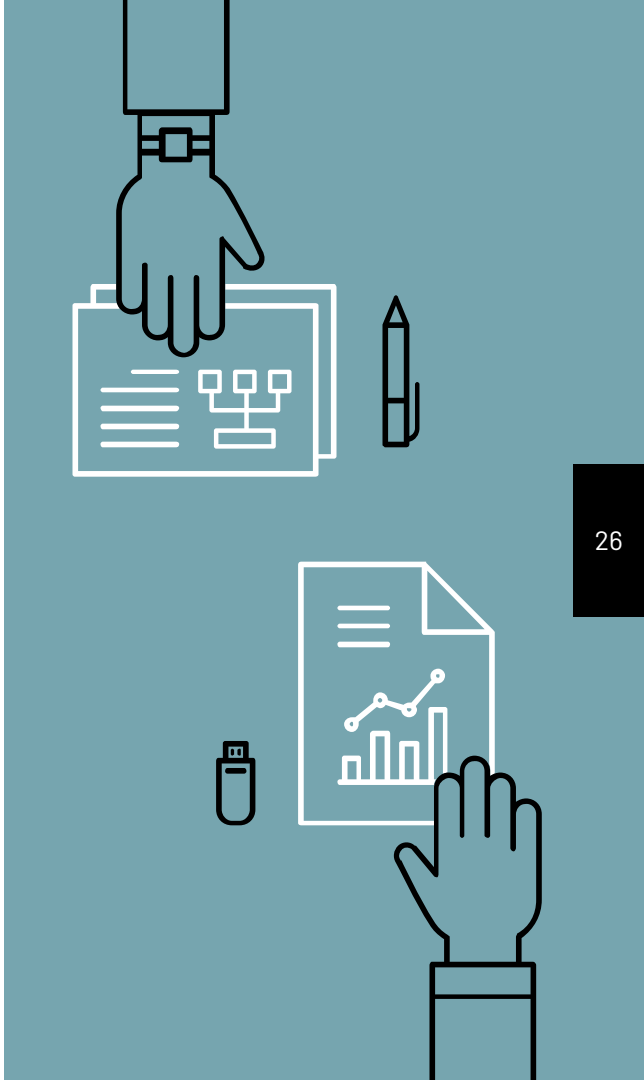
- Point and click
  - Many data processing / statistical analysis packages have graphical user interfaces (GUIs)
  - GUIs are convenient / intuitive but the actions you take with a GUI can be difficult for others to reproduce
  - Some GUIs produce a log file or script which includes equivalent commands; these can be saved for later examination
  - In general, be careful with data analysis software that is highly interactive; ease of use can sometimes lead to non-reproducible analyses
  - Other interactive software, such as text editors, are usually fine



Course Summary

Reproducible Research DON'Ts:

- Save output
  - Avoid saving data analysis output (tables, figures, summaries, processed data, etc.), except perhaps temporarily for efficiency purposes.
  - If a stray output file cannot be easily connected with the means by which it was created, then it is not reproducible.
  - Save the data and code that generated the output, rather than the output itself
  - Intermediate files are okay as long as there is clear documentation of how they were created



Homework

- Submit individual project
- Submit course evaluation