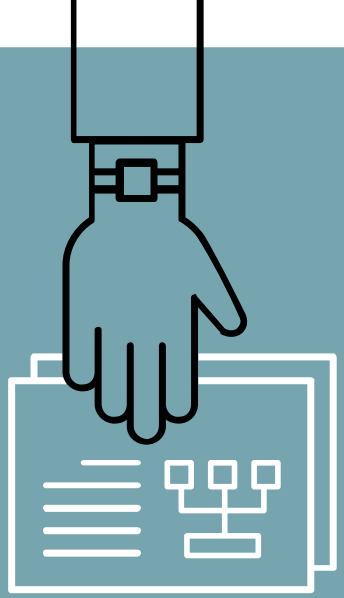
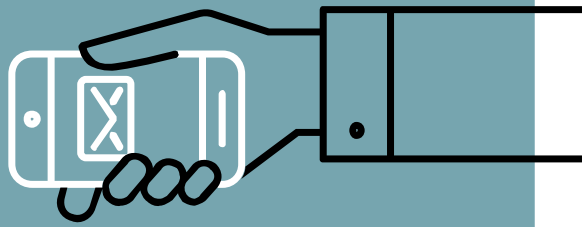
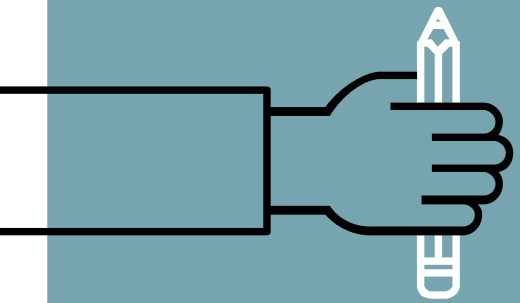
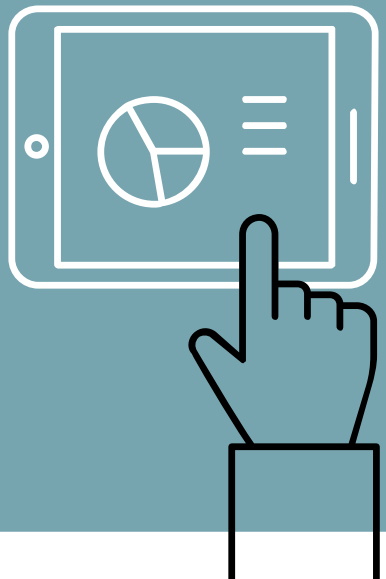




BST 270
Reproducible Data Science

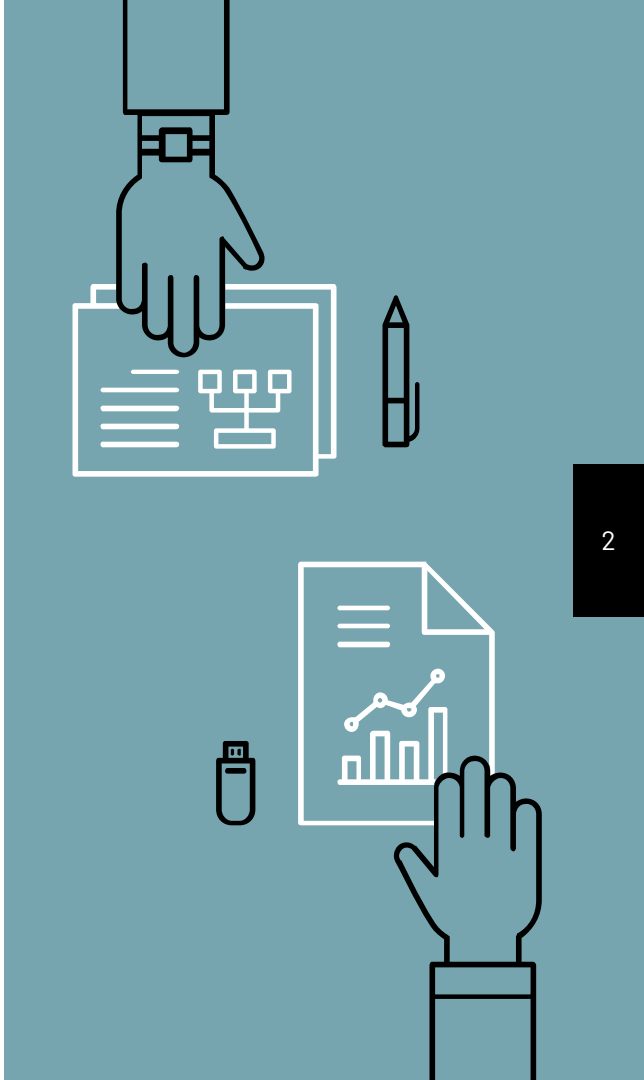
Winter 2022
Session 2





Comments

- I have a bone to pick with the poorly-formatted confidence intervals in Table 3 of the optimism/lipids paper
- Making typos in data analyses is inevitable. Instead of going through the code many more times trying to catch a small error, the video reminds me that a better way to check the validity of the analyses is to use positive/negative controls, simulation, summary statistics and/or diagnostic visualizations at every step of the analyses to make sure every step is doing what we want it to do to the data.
- I really liked the point about thinking of reproducible research in terms of software development. As someone with a background working at a software company, I found thinking about unit tests and automated validations a priori (or during development) helped me to write more robust and flexible code, as well as given myself and my team greater confidence in the accuracy of the results. I'd love to hear more about what standards are for unit tests (both positive and negative validations) on academic software that is to be shared.
- In many research I've known, people used dockers to ensure the reproducibility of both data and analysis. I believe this could be a better way of sharing our experiments, codes and results, where other people could get rid of building up the annoying computational environment.



Comments

- I found discussion about proper organization/workflow practices interesting and helpful. I wanted to emphasize one other common challenge researchers face, and why it is often useful to archive input data over time: data can change over the course of a study. Examples include clinical trials where subjects are added over time, and demographic/economic data that can be subject to revisions. In the past, I've worked on projects where the main results changed substantially as a result of significant revisions to some of the publicly available data we'd been using. Luckily, we had archived previous data versions and were able to trace back the change in results to the data revision, but had we not it could have led to huge headaches in trying to identify what was causing the change!

- Along the lines of factors that influence reproducibility: I think a common issue that many in the class have witnessed/been guilty of themselves (I certainly have been) is that there is a non-insignificant time cost to making your code (commenting, structure etc) and workflow reproducible. It can be tempting to eschew good practices to "get things running" (with half-hearted plans to go back and improve our messy code that never actually come to fruition). As researchers, I doubt many would deny the intrinsic value of practices that improve reproducibility, but we often stray from these ideals due to the associated costs of implementation. What are some actions we can take, both informally (e.g. holding colleagues accountable) and institutionally (e.g. journal requirements) to incentive ourselves to consistently adopt good practices?

Comments

- While watching video 2.4.1 on Experimental Design, I noticed how Dr. Quackenbush was speaking in very general terms. This made me think of how different research topics can be, and how it is hard to have a consistent definition of reproducibility across all projects.

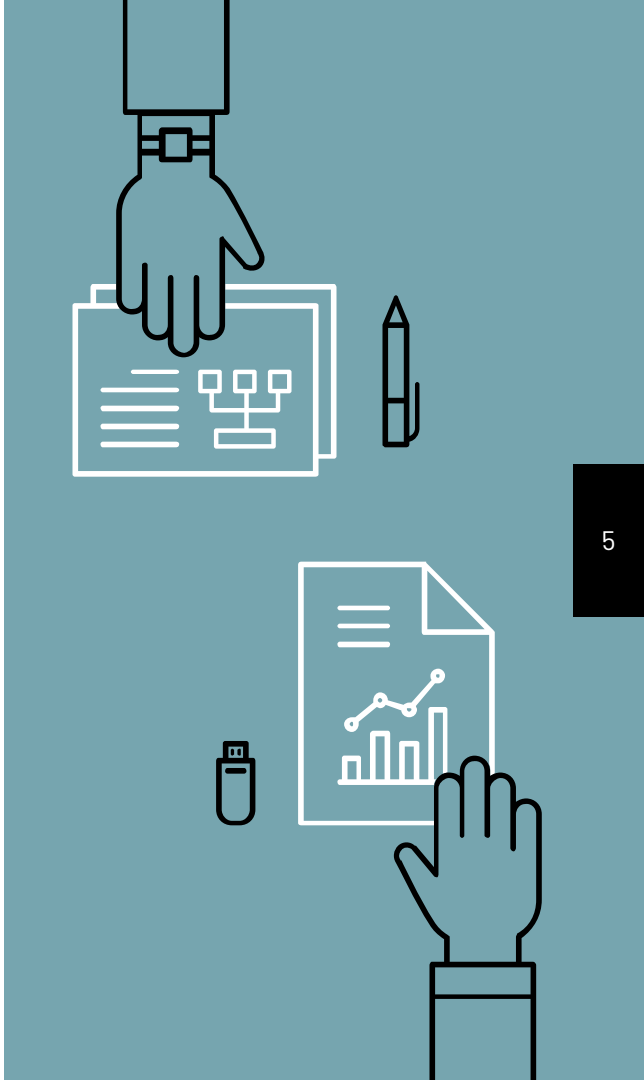
- I joined a new project in Fall 2021 that had its files organized similar to the shallow directory tree shown in video 2.4 on organization. I agree that it is a very useful structure to make things clear during onboarding and make it easy to use other people's code/data.

- I liked the organizational scheme that Curtis had mentioned in one of the videos. I have struggled with dates in file names especially, as I have a bad habit of seeing the file history to find the dates, and that does not work in a larger system when other people have to see my files as well.

Module 2 Discussion

Questions

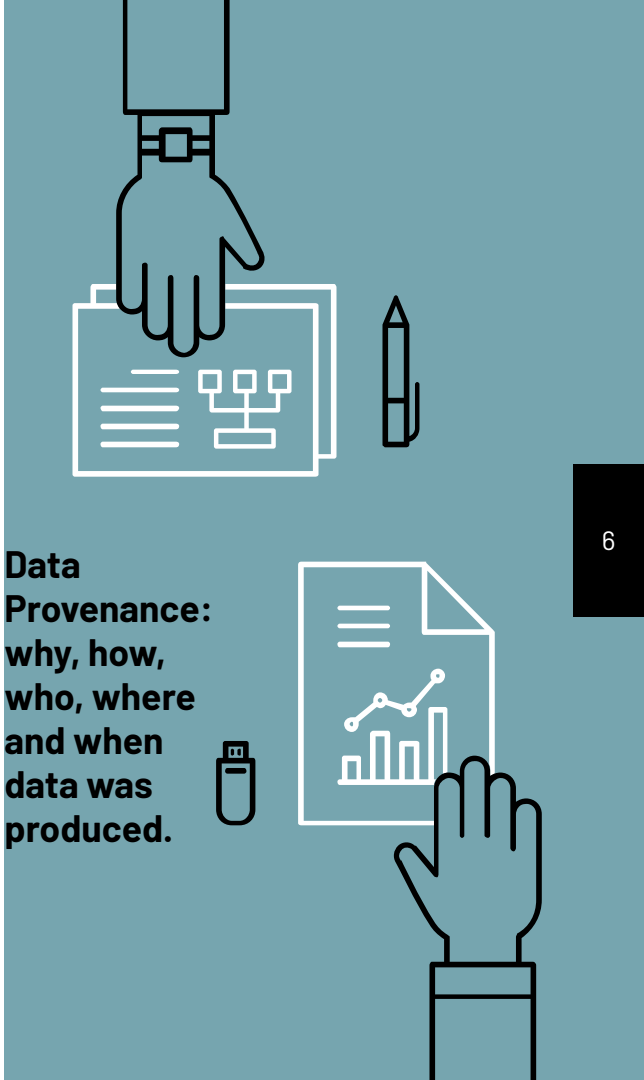
- **What is meta-data?**
  - Data about data
    - How the data was created
    - How the data was cleaned or transformed
    - Time and date of the creation/collection/transformation
    - Who collected/cleaned/transformed the data
    - Where the data is stored
    - Who has access to the data
    - File size(s)



Module 2 Discussion

Questions

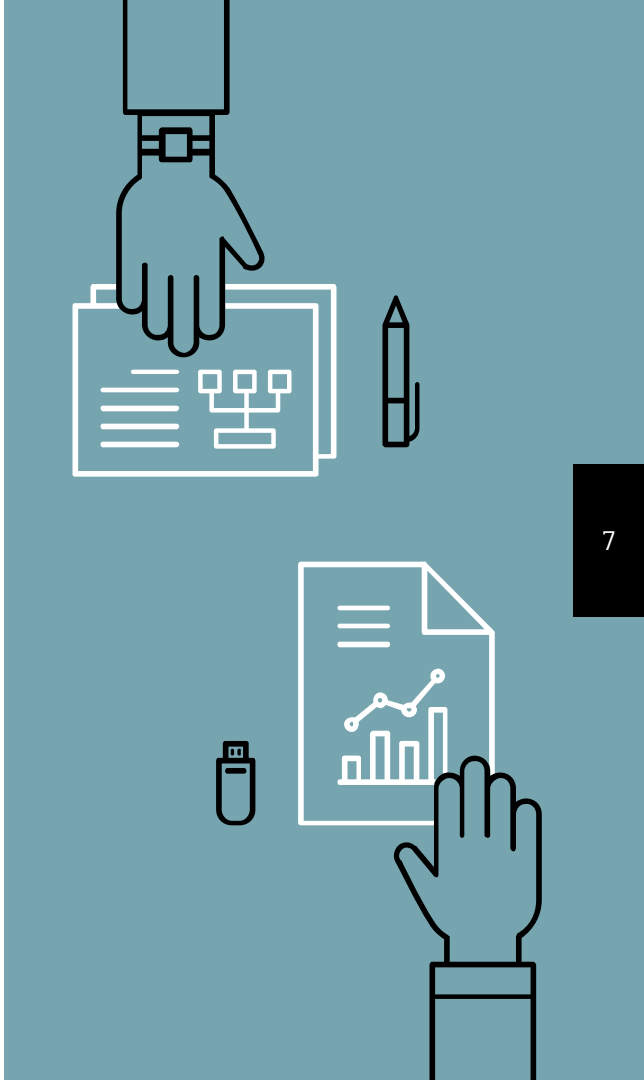
- **What is meta-data?**
  - Data about data
    - How the data was created
    - How the data was cleaned or transformed
    - Time and date of the creation/collection/transformation
    - Who collected/cleaned/transformed the data
    - Where the data is stored
    - Who has access to the data
    - File size(s)



Module 2 Discussion

Questions

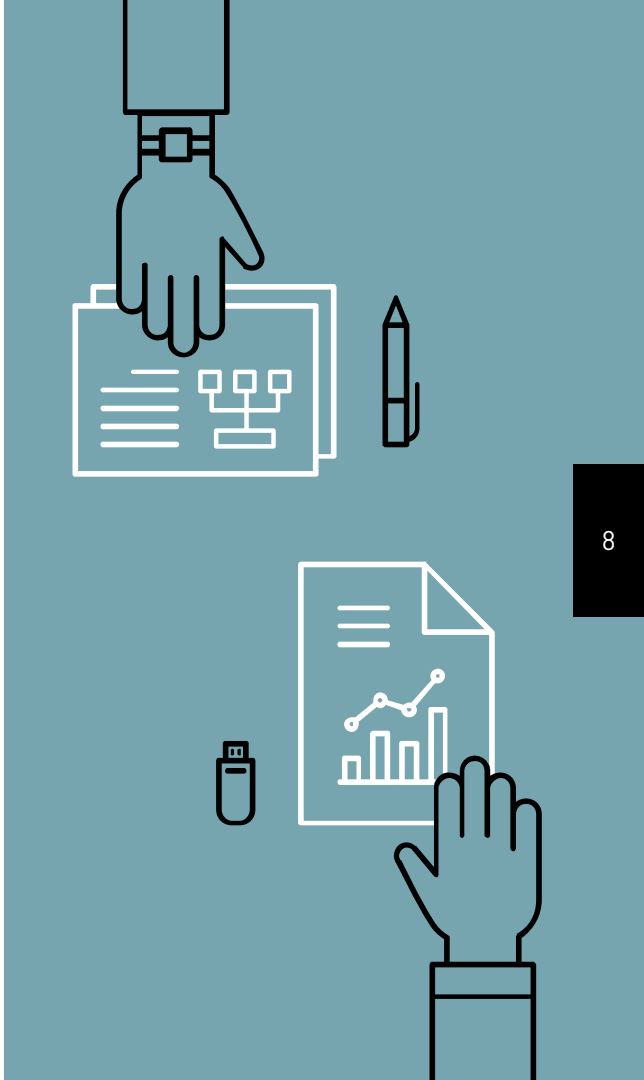
- **Problems from lack of metadata?**
  - You don't know the "history" of the data
    - This can lead to a misunderstanding/ambiguity of what variables represent or how they were transformed into a particular type which can lead to incorrect/invalid results and conclusions
    - Could lead to a privacy breach
    - Can mask any biases in data collection or cleaning



Module 2 Discussion

Questions

- **What should a data scientist put in the lab notebook?**
  - Code, including comments that help another read and understand what the code is doing
    - State what each variable is, what the inputs and outputs of a function are, the type of object you expect to get after running the code (a list, array, data frame, dictionary, number, etc.)
  - Text (not code comments) that guide the reader through your work
    - State the purpose of the project/code, why you are doing this and why you are doing it this way (why are you using that particular method?)

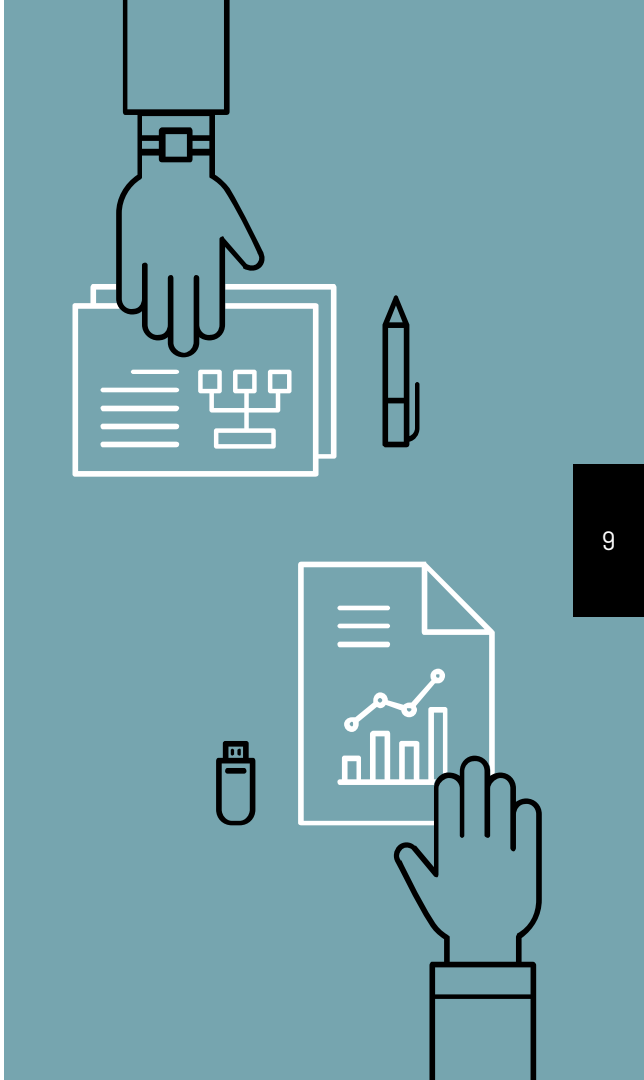


Module 2 Discussion

Questions

- **What should a data scientist put in the lab notebook?**
  - Visualizations
  - Interpretations (of plots, tables, output, etc.)
  - If your project requires running several files with code, mention where this file fits in to the workflow
  - Ideas for future work
  - Any notes from your PI or collaborators

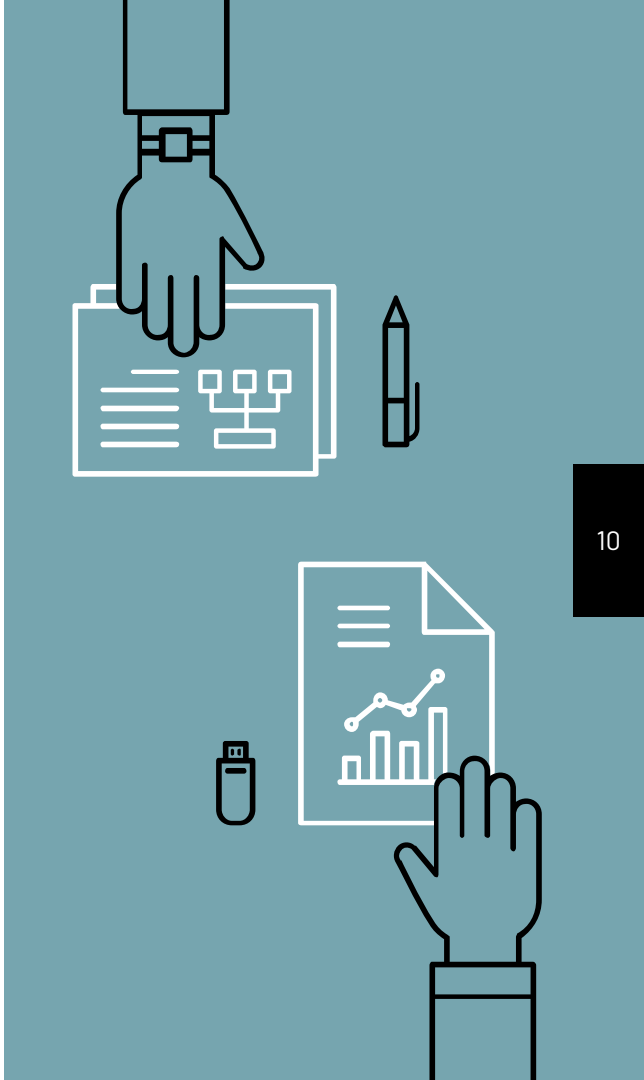
- **For reproducible code, how do you balance efficiency and interpretability? How do you write code that is readable but not extremely long and inefficient? Is brevity considered a priority in code? Is there a general way to balance organization/reproducibility for yourself versus for others?**
  - If you are someone who can write (or is forced to write because of the computational expense) very efficient code that is difficult to interpret, you should include more comments on what the code is doing, the expected inputs/outputs, sanity checks (unit, +/- tests), and general text about the purpose of the code
  - Brevity is great if you can get away with more interpretable or less efficient code



Module 2 Discussion

Questions

- **Is there a standard for well-written code?**
  - Short answer: yes!
  - Long answer: depends on your advisor/PI/team
  - Examples
    - Google's Python Style Guide
    - Google's R Style Guide
    - R Style Guide
    - The tidyverse style guide



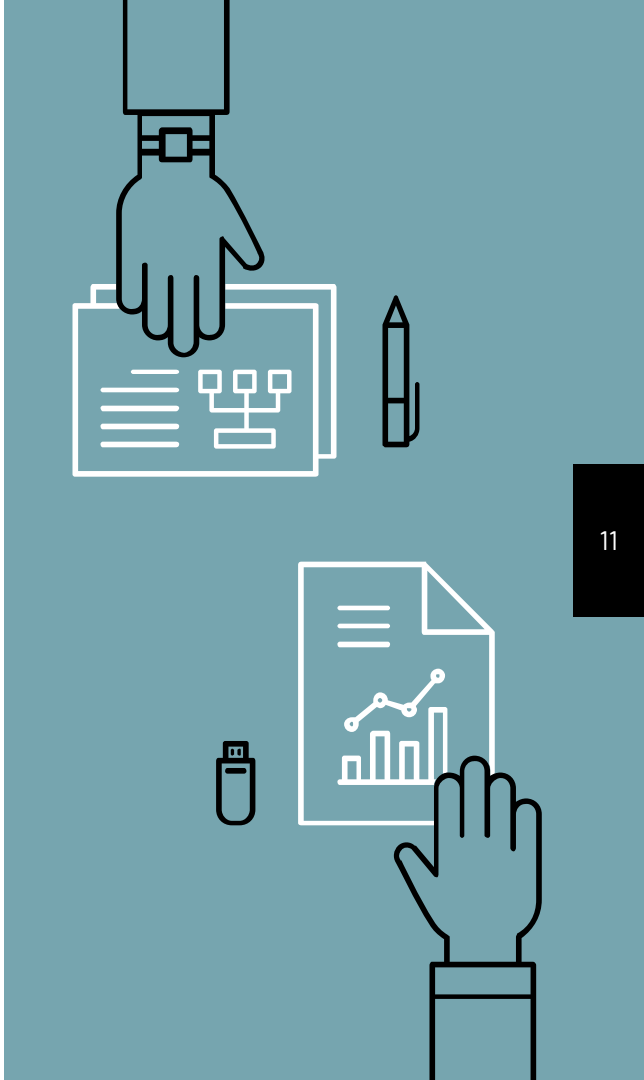
Module 2 Discussion

Questions

- **While the idea of a positive control seems familiar in standard biostatistics contexts (eg, running simulations where the model is correctly specified and checking that the model performs well), the idea of a negative control seems somewhat niche and less commonly used. Would be curious if anyone has examples where negative controls would be helpful in say their simulations or data analyses.**
  Difference between positive control and negative control with a few examples maybe?

Examples:

1. You have written a function that transforms birth dates (MM/DD/YYYY) into age (Years). You input 10 different birth dates and find that your function outputs the correct ages in years.
2. You derive a new test statistic and write a function to calculate its value. You calculate the test statistic for a data set that you know is not significant under the null, and find that it is indeed not significant.
3. You create a test to detect cancer cells. You run the test on a batch of healthy, non-cancerous cells and find the test result is negative.



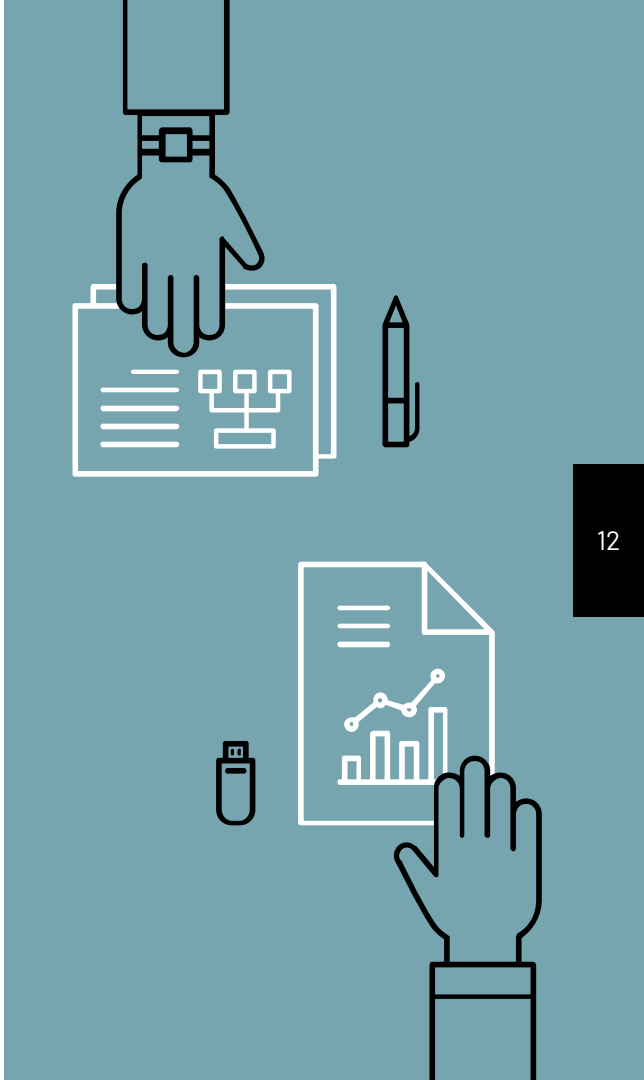
Module 2 Discussion

Questions

- **While the idea of a positive control seems familiar in standard biostatistics contexts (eg, running simulations where the model is correctly specified and checking that the model performs well), the idea of a negative control seems somewhat niche and less commonly used. Would be curious if anyone has examples where negative controls would be helpful in say their simulations or data analyses.**
  Difference between positive control and negative control with a few examples maybe?

Examples:

1. You have written a function that transforms birth dates (MM/DD/YYYY) into age (Years). You input 10 different birth dates and find that your function outputs the correct ages in years. (**Positive**)
2. You derive a new test statistic and write a function to calculate its value. You calculate the test statistic for a data set that you know is not significant under the null, and find that it is indeed not significant. (**Negative**)
3. You create a test to detect cancer cells. You run the test on a batch of healthy, non-cancerous cells and find the test result is negative. (**Negative**)



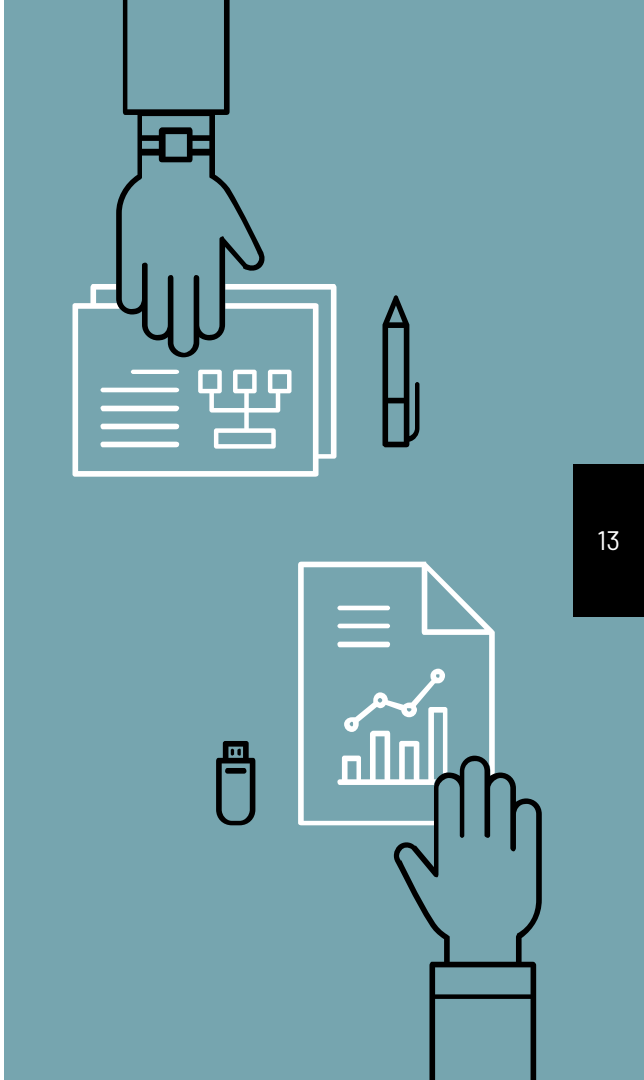
Module 2 Discussion

Questions

- **Sometimes, for a software, different computing platforms (like clusters vs. windows vs. mobile) might lead to different computation results, even though we have used the same function in our code. How can we possibly prevent this?**

  In video 2.4.5 Providing Workflows and Results, Prof. Huttenhower touches on how many cloud computing platforms are designed to move from one virtual machine to another. What does this mean exactly?

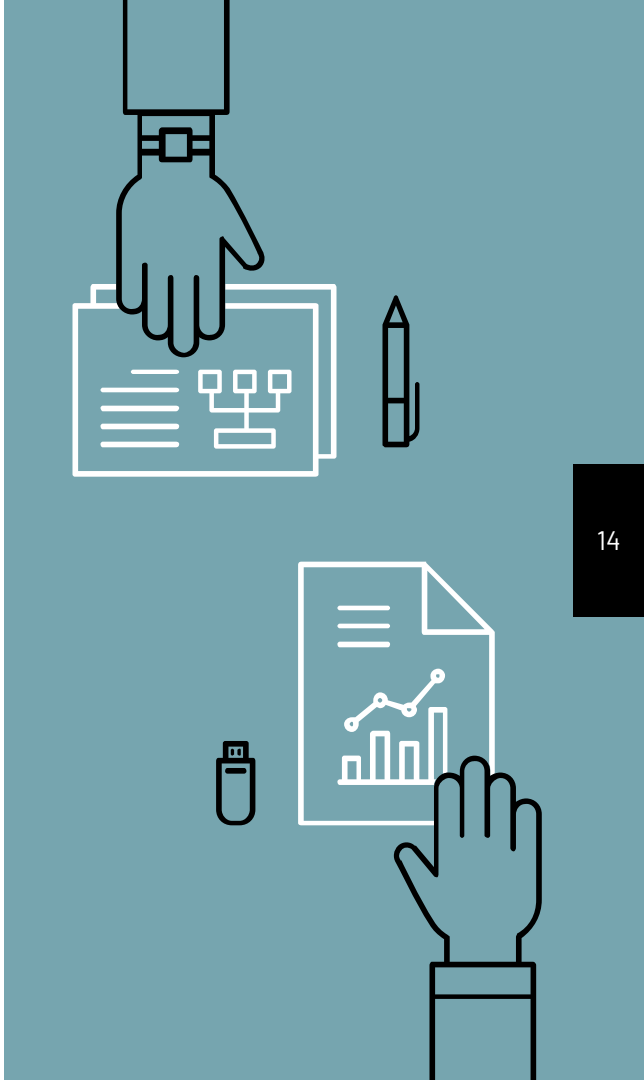
  - Virtual environments and containers
    - Docker (more info in Module 6)
    - Kubernetes
    - Python virtual environment
    - renv: Project Environments for R
  - Setting a random number generator seed (but this can also differ between machines)



Module 2 Discussion

Questions

- **What's the best strategy to test your code? Usually the code for a proposed method is long, and asking others to write the code from scratch to reproduce the results is time consuming. However, simply asking others to read the code and test it may not be enough to detect potential bugs and mistakes.**
  How common is it to have someone else fully re-run your analyses prior to publication? I understand the value of checking your workflow for mistakes and interpretability from the perspective of others. However, especially for computationally intensive analyses, it seems like this may be unreasonable for many research groups with less resources than tenured PIs at Harvard.
  - Will depend on your supervisor/team
  - Some teams have at least 1 other person check all code (some team members each look at a specific part of a script), some teams have more or run it through tests before implementing
  - Usually the best way is to break it down into smaller pieces
  - If it is computationally expensive, try to run it on much smaller datasets

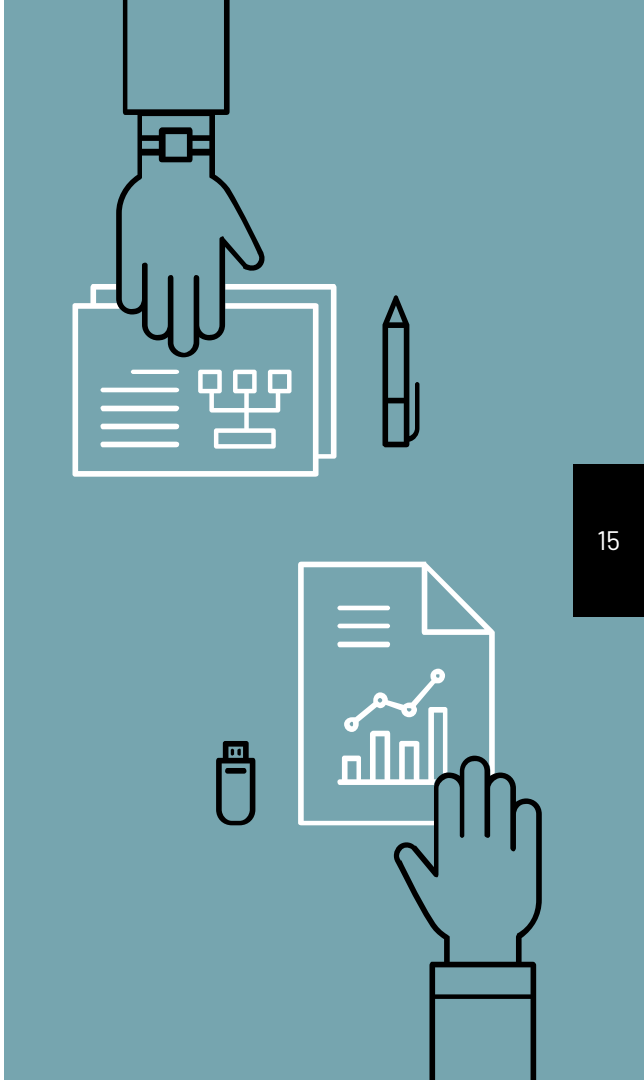


Module 2 Discussion

Questions

- **Is there a way to automatically backup code to GitHub?**
  - Yes! But $$$
  - You can also use Google Cloud Platform or AWS (also $$$)
  - Or this code (but I don't know if this works or not)

- **Automated data/results sharing sounds really useful but probably requires a lot of careful planning out. Is it likely to accidentally share the wrong information?**
  - There is always that risk, which is why there are so many training modules to complete before touching biomedical data
  - There are (or should be) several checks made by multiple individuals before releasing anything to the public or other collaborators/team members.

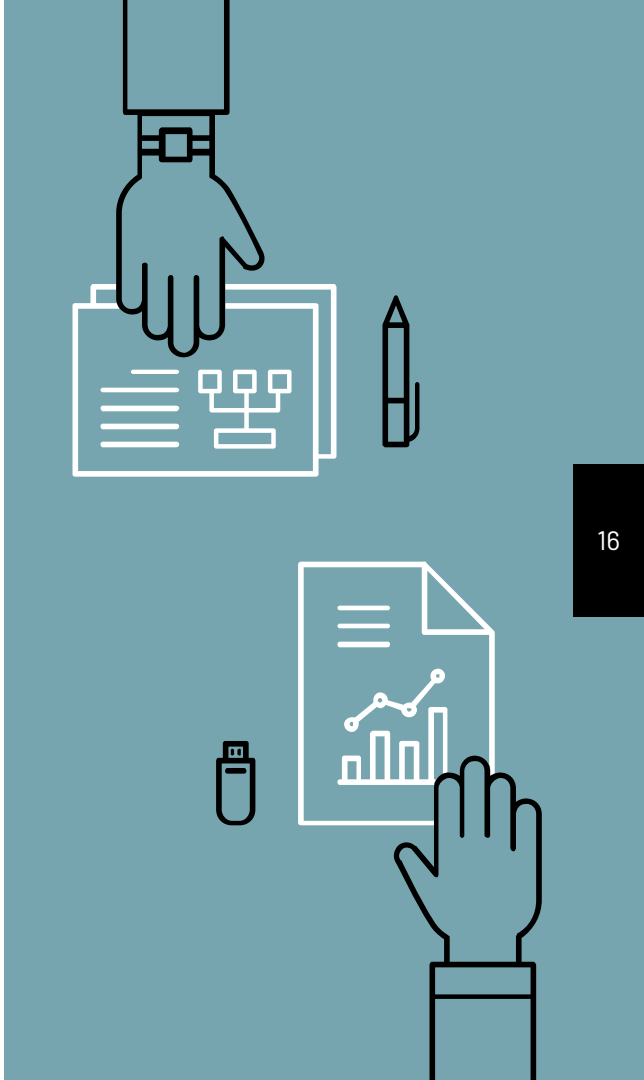


Module 2 Discussion

Questions

- **How do you organize your data/results/papers?**
  - Google Drive, Dropbox, Overleaf
  - A document that explains in detail where all of the pieces are (i.e., the code, the manuscript, the lit review, etc.)
  - I personally keep detailed notes of meetings and any thoughts or issues I experience during the project

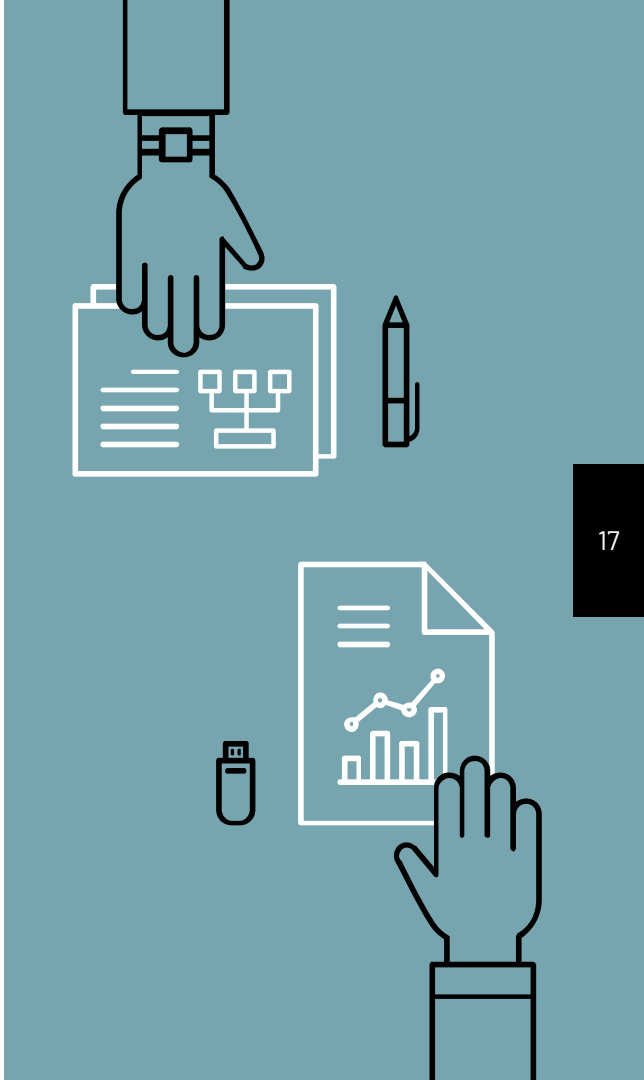
- **Are there ever problems that arise when researchers use proprietary software for analysis, where the actual code is not available to the public?**
  - There can be - you'll see this in Module 3
  - This does happen a lot and journals can require example code for synthetic data or a toy example
  - Sometimes you can request to see the code - but this is difficult



Module 2 Discussion

Questions

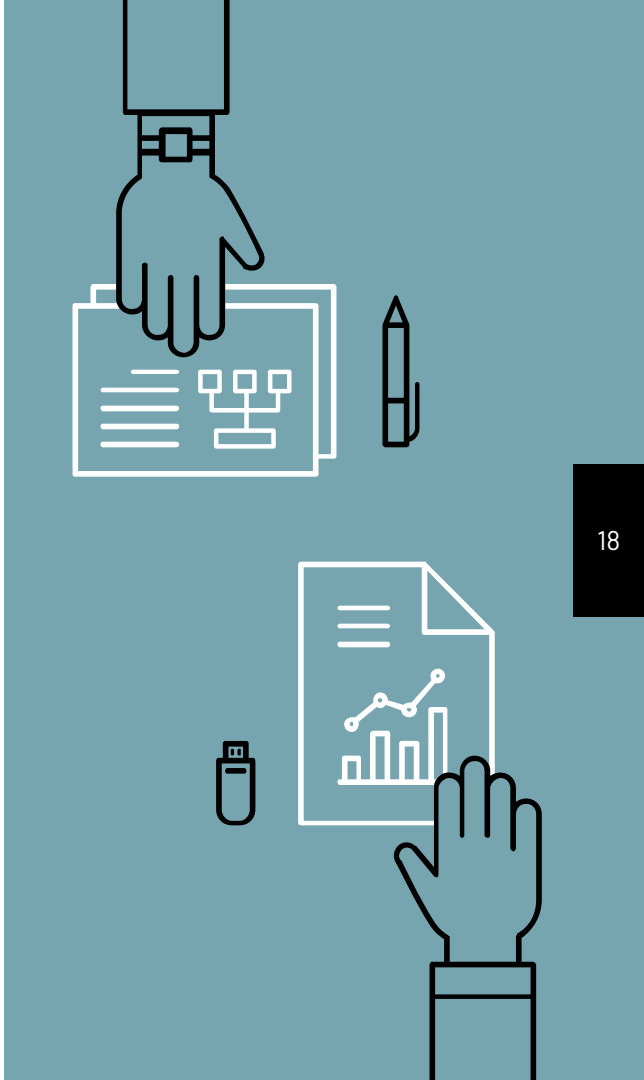
- **Video 2.3 mentions doing visual representations and tests to check the data, and I wonder what that looks like for high dimensional data or other data that makes visualization complicated.**
  - Break it down
    - Don't try to plot everything together
    - Univariate/bivariate visualizations
      - Example: plot the age distribution. If you have someone who is 300 years old, you have a data entry error
  - I like this Towards Data Science post



Module 2 Discussion

Questions

- **When submitting to a journal, are the reviewers allowed or encouraged to reproduce the results?**
  - Short answer: I don't know for sure.
  - Long answer: I *think* reviewers can do this if they really want to and have access to the data. I haven't ever experienced a reviewer request access to the code or other materials needed for the analysis. I imagine the overwhelming majority of reviewers don't have the time to do this (they are also in academia and there are (usually) short(ish) deadlines for reviews). If it's really impactful research like the case study presented in Module 3, I would hope someone would attempt to reproduce the results.
  - New Experimental Results journal



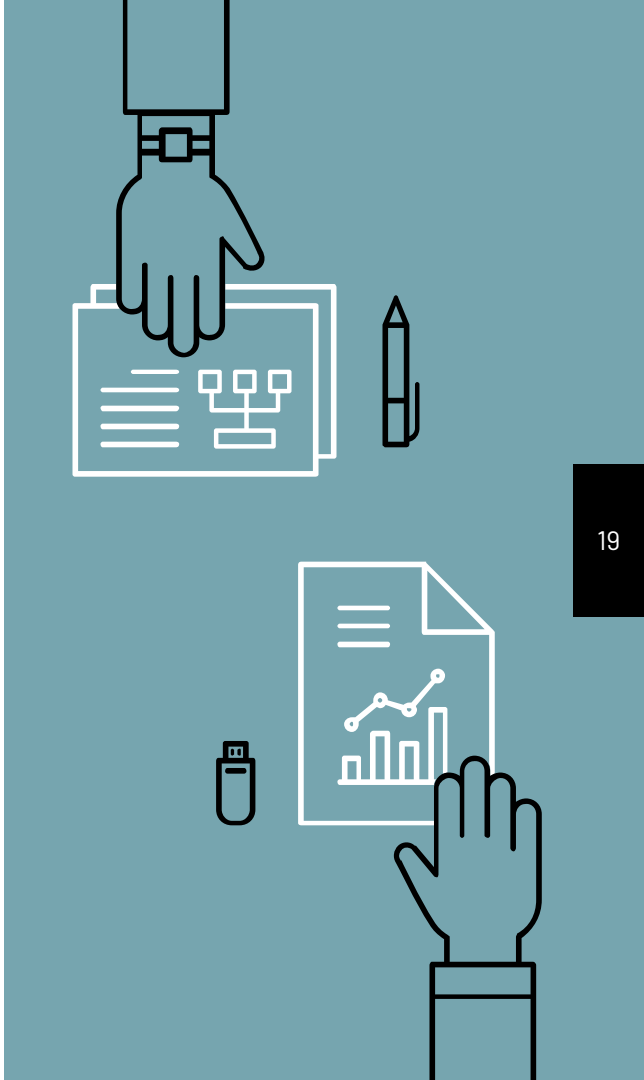
Module 2 Discussion

Questions

- **Apart from requiring authors to make their data and code publicly available, do you see journals making any more steps for enforcing reproducible research? Often these files are quite incomplete, poorly documented, not functioning, etc. The videos mostly motivate reproducible research for the sake of being a good researcher, but am curious how this can be / is enforced more generally.**

  We'll get to this tomorrow and later in the course!

  Short answer: yes!



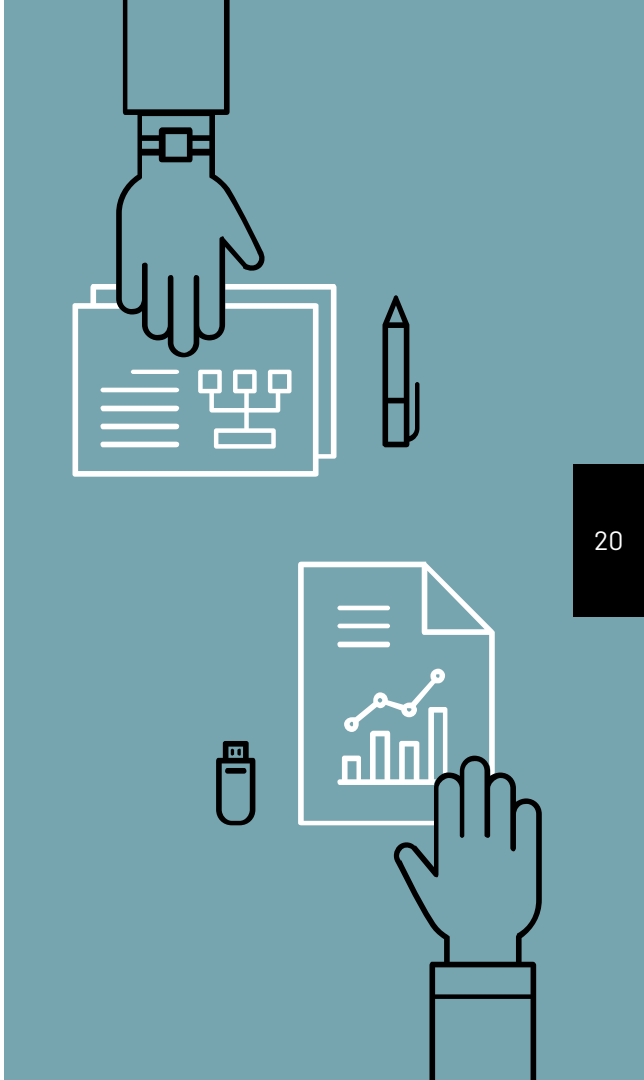
Module 2 Discussion

Questions

- **When we do analyses with identifiable data, how should we make our data available to the public for reproducibility. What is the common/best practice?**

  A question I have is regarding the sharing of code and data when collection data is sensitive, and/or someone needs explicit permission to be able to access the data. What is the process by which someone should share code for a project in this case, if it constantly references data that is not permissible to disseminate widely?

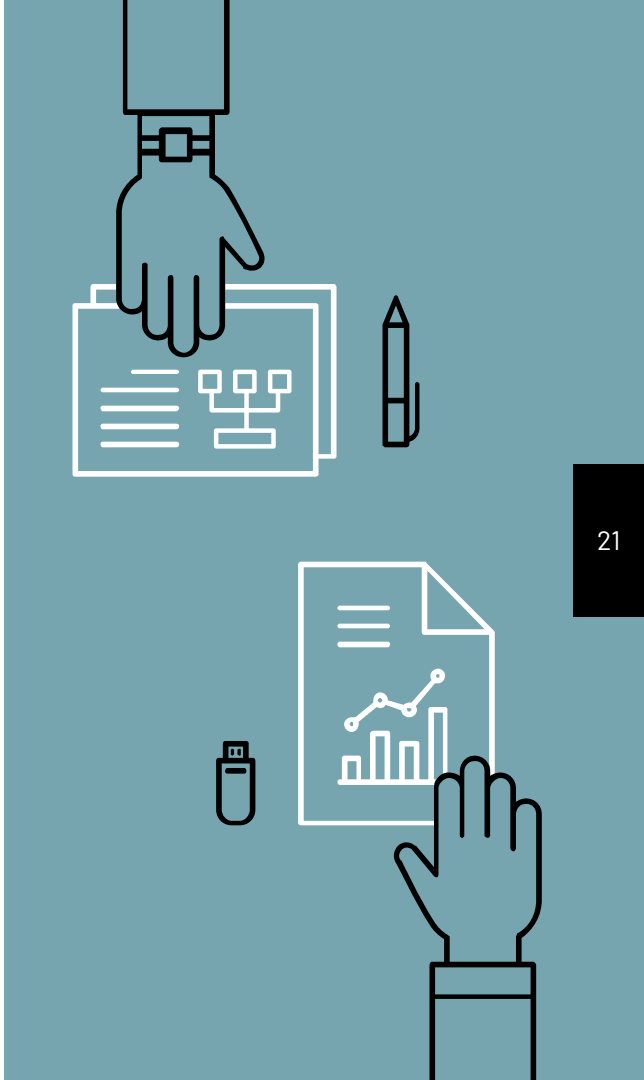
  We will get to this tomorrow!



Module 2 Discussion

Key Takeaways

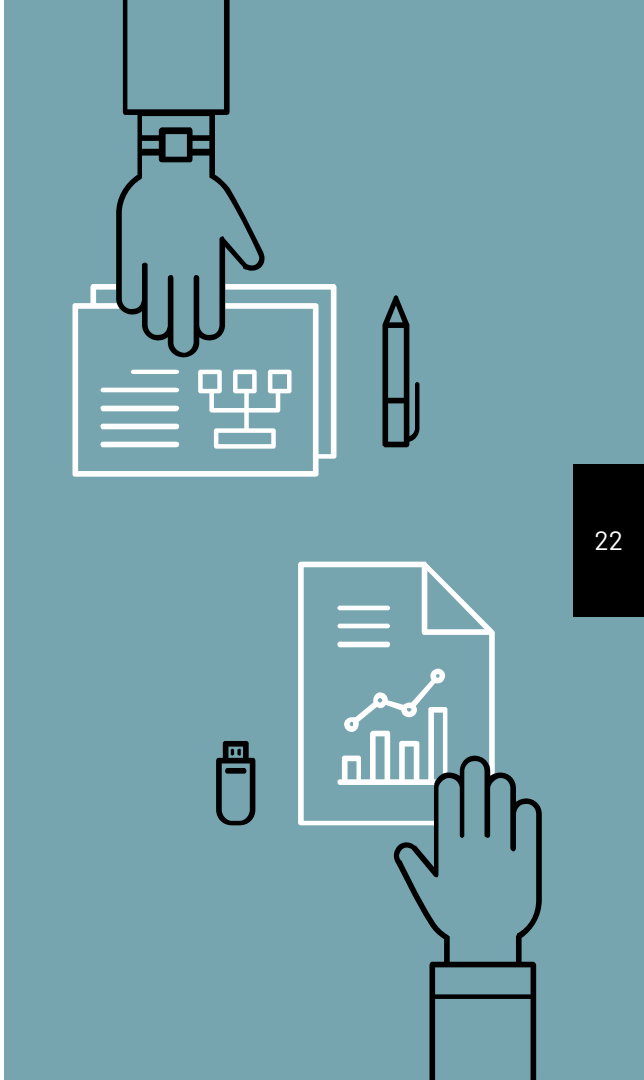
- The structure of file storage is important
- Writing code that detects its own potential errors is important
- Metadata (data provenance) is essential
- Variable naming
  - Should be as informative/intuitive as possible



Module 2 Discussion

Key Takeaways

- **Reproducibility: Data vs Analyses**
  - Reproducibility of data: more about *files* – where you put them and how you manage them
    - Example: databases with notations for data
  - Reproducibility of analyses: more about *programs*, *code* and *workflows*
    - Example: literate programming or revision control repositories for code that carries out analyses

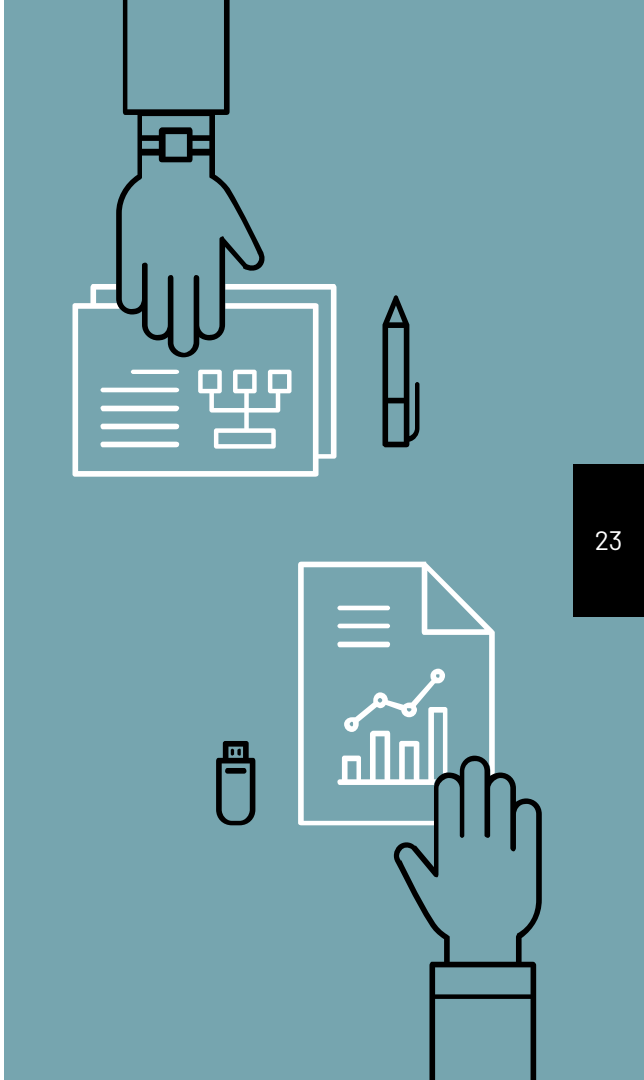


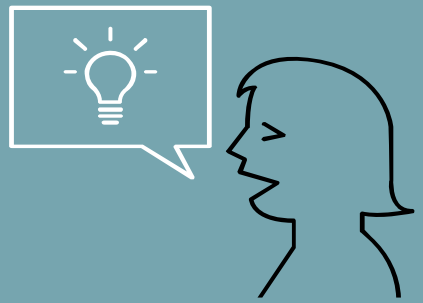
Module 2 Discussion

Key Takeaways

- **Reproducibility: Electronic vs Protocol**
- Electronic reproducibility: activities that a computer can carry out for you.
  - Checking assertions, control results, or unit tests
  - Storing documentation or data in a particular repository
  - Managing revision control history for an analysis workflow.

- Protocol reproducibility: activities and best practices you must do yourself.
  - Using a revision control repository or a public database
  - Designing your experiment to support convenient reproducibility by others
  - Picking a consistent naming scheme for your files and folders
  - Writing enough documentation for another user or your future self to read and understand





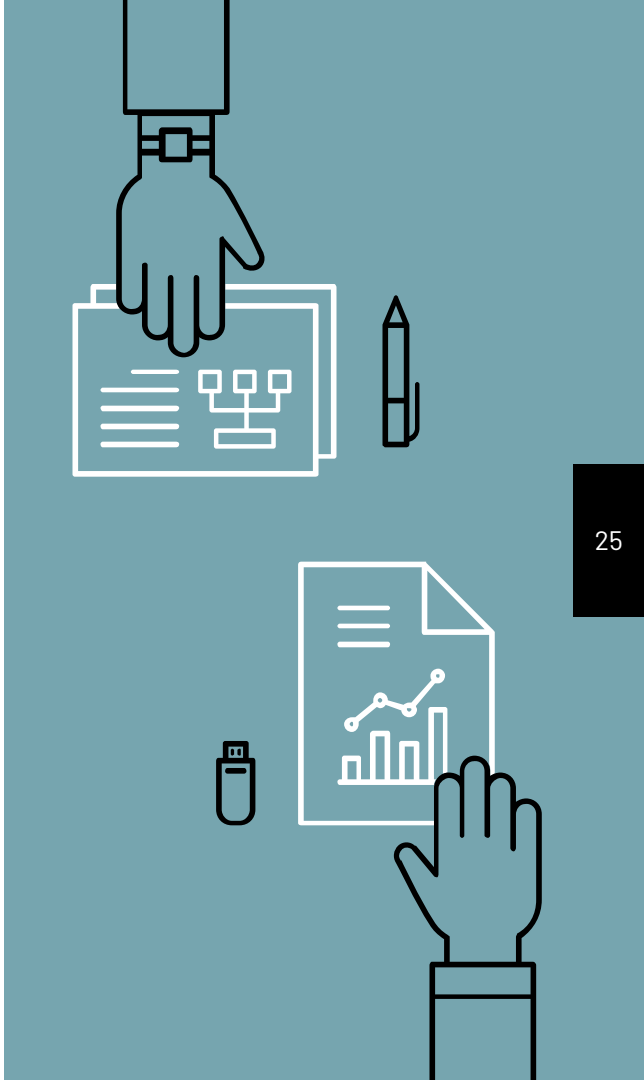
In-Class Project

We will attempt to reproduce the results from and critique the reproducibility of:

[1] Boehm, J. K., Williams, D. R., Rimm, E. B., Ryff, C., \& Kubzansky, L. D. (2013). Relation between Optimism and Lipids in Midlife. The American Journal of Cardiology, 111(10), 1425-1431.

Specific Tasks:

- Create a data dictionary
- Wrangle data
- Recreate Figure 1
- Recreate Tables 1-5
- Critique reproducibility

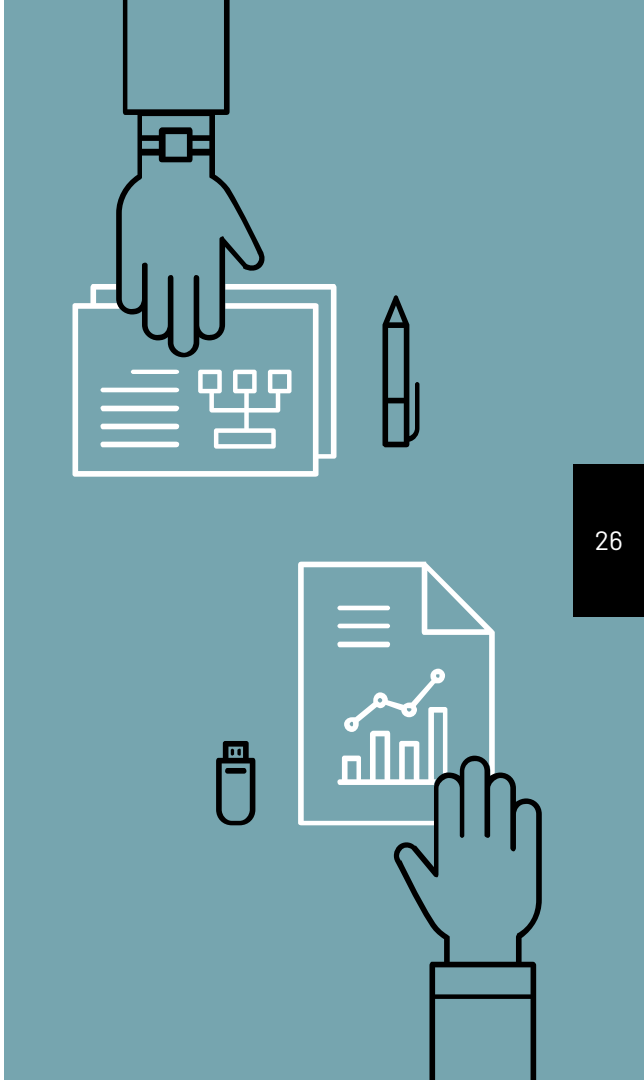


We will attempt to reproduce the results from and critique the reproducibility of:

[1] Boehm, J. K., Williams, D. R., Rimm, E. B., Ryff, C., \& Kubzansky, L. D. (2013). Relation between Optimism and Lipids in Midlife. The American Journal of Cardiology, 111(10), 1425-1431.

Specific Tasks:

- Create a data dictionary
- Wrangle data
- Recreate Figure 1
- Recreate Tables 1-5
- Critique reproducibility

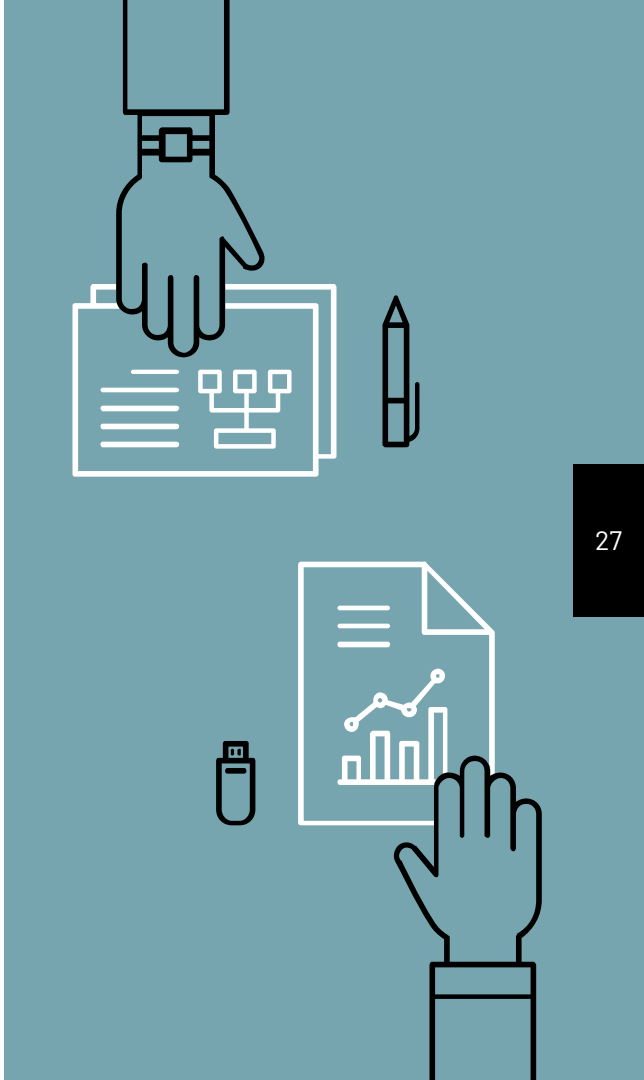


MIDUS II Data Sets

1. Data and supporting codebook and other documents

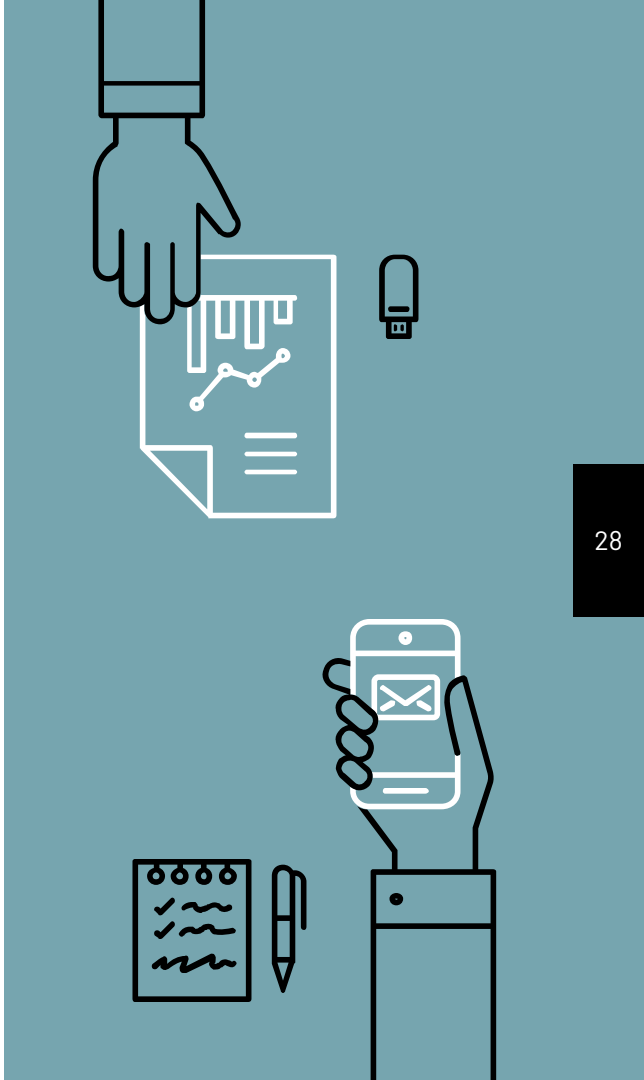
2. Biomarker data

This particular article focuses only on MIDUS II data, including biomarker data, and investigates the relationship between optimism and lipids. You can download the data in multiple formats. We will be using the **R files in class** and performing all data cleaning and analyses in R and an RMarkdown file.



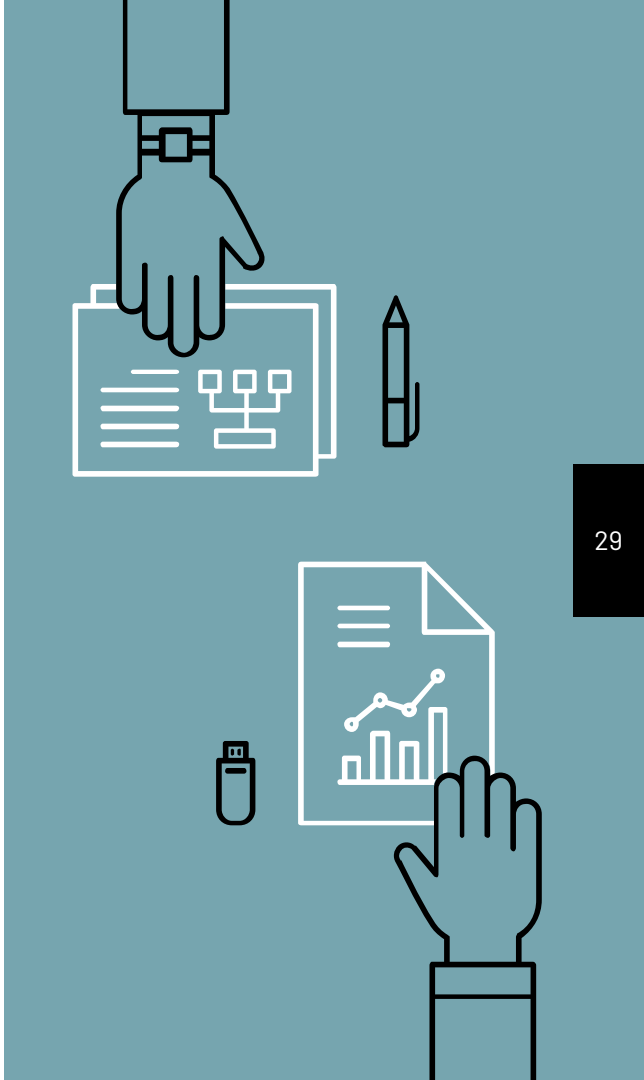
Data Dictionary

- We will be working in teams to create a data dictionary for this project
- All variables used for the analysis should be posted here with any notes you think are informative/necessary
- Assignments
  - Group 1: Sarika, Ellen, Raphael, Nick, Madhav, Lin
  - Group 2: Xiang, Michael, Parker, Addison, Carolin, Yujie Wu
  - Group 3: Keith, Emma, Jenna, Sean, Zhu, Qingru
  - Group 4: Luke, Rebecca, Jinglun, Carmen, Sofia, Yujie Zhang
  - Group 5: Tingyi, Zhiyun, Tayler, Nutan, Cece, Tianxiao



Data Wrangling

- Once we have the variable names organized, we need to wrangle the data and determine our sample
- In the same groups, write code to deal with any missing values, weird values, recoding, etc., according to the paper
- We'll discuss all code tomorrow



Homework

- Watch Module 3 videos
- Submit Module 3 discussion points