Principles, Statistical and Computational Tools for Reproducible Data Science

HarvardX – PH527X

Black Raven (James Ng)

30 Dec 2020 · 31 min read

This is a memo to share what I have learnt in Principles, Statistical and Computational Tools for Reproducible Data Science, capturing the learning objectives as well as my personal notes. The course is taught by Curtis Huttenhower and John Quackenbush from HarvardX, and it includes 7 chapters:

Chapter 1: Introduction to Reproducible Science
  Overview of course and instructors
Chapter 2: Fundamentals of Reproducible Science
  Concepts, background and language of reproducible data science
Chapter 3: Case Studies in Reproducible Research
  Stories of success and failure that can motivate the need for reproducible data science
Chapter 4: Data Provenance
  Define reproducibility needs specific to modern scientific data analysis and publishing
Chapter 5: Statistical Methods for Reproducible Science
  Theoretical background affecting analysis, machine learning and mathematical approaches
Chapter 6: Computational Tools for Reproducible Science
  Hands-on software, database and other electronic mechanisms for reproducible research
Chapter 7: Conclusion to the Course
  Recap on what you have learnt

Chapter 1. Introduction to Reproducible Science

This course is to make you aware of pitfalls in reproducible data science, some failure/success stories in the past, and tools and design patterns that might help make it all easier. But ultimately it'll be up to you to take the skills you learn from this course to create your own environment in which you can easily carry out reproducible research, and to encourage and integrate with similar environments for your collaborators and colleagues.

Experimental design, data provenance, analytic methods and tools, and reporting science play a critical role in the research ecosystem to ensure scientific rigor, robustness and transparency. Statistical and computational methods and tools are fundamental for making scientific results reproducible.

Course Objectives
Upon successful completion of this course, you should be able to:
- Describe the fundamentals and importance of reproducible research
- Assess the reproducibility of others' research
- Create a fully reproducible research project
- Develop new methods and tools for reproducible research

1.1.2 Pre-Course Evaluation

Welcome to Reproducible Data Science! Before you start this course we would like to gauge your knowledge prior to the course. Please fill out these questions to the best of your ability. We do not expect you to know the answers.

How do you define reproducible?

Reproducible means to be able to anticipate, predict, and yield the expected outcomes given the inputs and initial conditions.

Why does reproducibility matter?

To ensure predictability, to confirm that the understand and knowledge is correct, to be able to yield the expected outcomes when executing the matter.

Which of the choices below are journal requirements for reproducibility? Check all that apply.

- ☑ Require the use of Standards, such as the Minimal Information About a Microarray Experiment (MIAME) or ARRIVE (Animal Research: Reporting of In Vivo Experiments) as applicable.
- ☑ Require that authors report their replication strategy—how often each experiment was performed and whether the results were substantiated by repetition under a range of conditions.
- ☑ Require that statistics be fully reported in the paper, including the statistical test used, exact value of N, definition of center, dispersion and precision measures (such as mean, median, SD, SEM, confidence intervals)
- ☑ Require authors to state whether the samples were randomized and specify method of randomization, at a minimum for all animal experiments.
- ☑ Require authors to deposit data sets in public repositories.
- ☑ Require authors to state whether experimenters were blind to group assignment and outcome assessment, at a minimum for all animal experiments.
- ☑ Require authors to state whether an appropriate sample size was computed when the study was being designed and include the statistical method of computation. If no power analysis was used, include how the sample size was determined.
- ☑ Require authors to state whether an appropriate sample size was computed when the study was being designed and include the statistical method of computation. If no power analysis was used, include how the sample size was determined.
- ☑ Require authors to state whether experimenters were blind to group assignment and outcome assessment, at a minimum for all animal experiments.
- ☑ Require authors to clearly state the inclusion and exclusion criteria that were used for exclusion of any data or subjects. Include any similar experimental results that were omitted from the reporting for any reason, especially if the results do not support the main findings of the study. Describe any outcomes or conditions that were measured or used and are not reported in the results section.
- ☑ Require authors to share their software and code in public repositories

Please name the code and/or data repositories you have used before (if any) below:

Code/Data repository 1   `data.gov.sg`

Code/Data repository 2   [ ]

Code/Data repository 3   [ ]

Please describe the difference between privacy and security to the best of your ability

Privacy is the state of being able to protect personal information and sensitive data from being accessed by external parties. Security is the enabling environmennt for privacy.

What are the advantages of boot strapping? Answer to the best of your ability.

- [ ] It is a way to derive estimates of standard errors and confidence intervals
- [ ] It is asymptotically more accurate than the standard intervals obtained using sample variance and assumptions of normality
- [x] It is always asymptotically consistent
- [x] It is a way to check the stability of results

What is the meaning of a "pull" of a git repository? Answer to the best of your ability.

- ( ) To make a repository unavailable to access
- (•) To get the newest version of a git repository
- ( ) To merge new changes onto a git repository
- ( ) To make a repository available to access

Why did you enroll in this course? What do you hope to get out of this course?

I believe in continuous learning. The world is constantly evolving and I need to keep up with the state-of-the-art stuff and deepen my knowledge. I hope that this course will help in my career progression in the data science field.

1.1.4 Pre-Course Survey

What tools or skills are necessary for reproducible science?
- bash (command line utilities in unix/linux), python (open-source software ecosystem)
- code hygiene and testing, good practices for scientific software development
- statistical knowledge and analysis
- knowledge of licensing options for sharing software
- github, software version control

What does reproducible research mean to you?
- reproducibility supports collaboration
- communication of methods, processes and results
- reproducibility supports efficient way of working
- reproducibility enables learning
- reproducibility builds trust in people and methods

Chapter 2. Fundamentals of Reproducible Science

2.1 Why Reproducible Science Matters

When we publish papers we want to make sure that we have an online supplement where the data is available and the software is available so that someone can actually reconstitute the analysis.

We have a series of design patterns that start at the beginning, make sure that the experiment is built in such a way that you'll have replicates, you'll have data, positive or negative controls, data that supports reproducibility.

That you're recording an experimental protocol, that you have a plan for a computational experimental protocol. That you have inviolate input data.

If you run an experiment once, those data are very special after that. As much computational data as you need with provenance, remembering where it came from and how it was generated.

A protocol capture process for the computational portion of the experiment. Documentation at the end explaining to somebody else, to your future self, here's what we did.

And I think one of the really positive things about the way the entire scientific field has moved is that the journals are increasingly recognizing that when papers are published they have to be reproducible. So I mentioned the fact that we tried to make software and code available with every paper we publish.

As a community, we needed to create a culture that's aware of the challenges of data analysis and data reproducibility and research reproducibility. There is no recipe you can have for creating a culture. But it really needs to be part of the research process from beginning to end.

2.2 Definitions and Concepts
Definition of "**Reproducibility**"
An experiment can be considered reproducible if a different research team can obtain its input data, its computational tools, understand them, and rerun the same methods to obtain the same result.

Definition of "**Replicability**"
Replicability means that a different research team can start with the same concept, and perhaps overlapping information, but regenerate their own data and methods that ultimately produce the same result.
It is a much more challenging and costly process. But it relies heavily on the best practices introduced with simple reproducibility.

Difference between **reproducibility of data** versus **reproducibility of analyses**.
In quantitative research, reproducibility of data is usually more about files: where you put them, and how you manage them. Whereas reproducibility of analyses is more about programs, code, instruction and workflows.

Controls, both positive and negative, can and should always be included, and checked automatically, if possible, at each stage of a study intended to be reproducible.
Reproducible data science analyses should include **positive controls**, or unit tests (statisticians will recognize these as simulations), that consist of defined inputs for which a particular output or result is expected. These can be pre-specified, and automatically checked throughout the analysis process.
Similarly, **negative controls**, in the context of data science, generally represent null unit tests-- inputs for which no or an uninteresting result is expected, and can be checked.

Protocol reproducibility best practices, such as using a revision control repository or a public database in the first place, designing your experiment to support convenient reproducibility by others, picking a consistent naming scheme for your files and folders, or writing down enough documentation for another user, or your future self, to read and understand.

2.3 Factors Affecting Reproducibility

- Data **volume**. Analyses of big data are challenging for many reasons, and processes that are slow or difficult to keep track of as a result can certainly impede good reproducibility practices. Consider using a distributed file system in parallel processing, as well as streamed processing that can save progress, start and stop as needed.

- Data **velocity** means data that change frequently. You can include an explicit annotation (automated) process that captures where new files are coming from, at what time, and who is responsible for them.

- Data **variety** as different aspects of data complexity. Never underestimate the power of simple typos to disrupt science. They're impossible to prevent, but you can and should check for their unwanted effects. Handling data complexity reproducibly can be automated by including positive and negative controls, simulations and tests. But you should check with your own eyes that each step is working, even if it means looking at just a small fragment of data, maybe just chosen randomly or from a simple test case.

- Data **veracity** (accuracy) can be helped by coming up with clever diagnostic visualizations that are appropriate for your complex data. Make plots or summary statistics that will reveal problems or outliers or unwanted technical patterns sooner rather than later.

- Data **provenance**, the natural history of your data can affect reproducibility. Make sure that you always know where your data came from and what's been done to it. Input data should always have an explicit source, such as a URL and a time and date stamp. All data should have the contact information for one or more responsible parties associated with it. Analysis data products should be annotated with exactly what programs and parameters were used to generate them. All of this information can be captured formally in the form of a workflow or metadata database or informally as a readme file, but making sure that it's available somewhere will improve the long term reproducibility of your experiments.

2.4.1 Experimental Design

"It doesn't matter how beautiful your theory is, it doesn't matter how smart you are.
If it doesn't agree with experiment, it's wrong."
-- Richard Feynman, physicist, Nobel Prize in physics (1918-1988)

Having a good experimental design

What we want to do in doing experiments is develop and test models that help us to understand the world around us, and ultimately to use those models to make predictions. We begin with a hypothesis and then develop an experiment that will test that hypothesis. The goal of an experiment dictates how everything from collecting the samples to generating the data are handled.
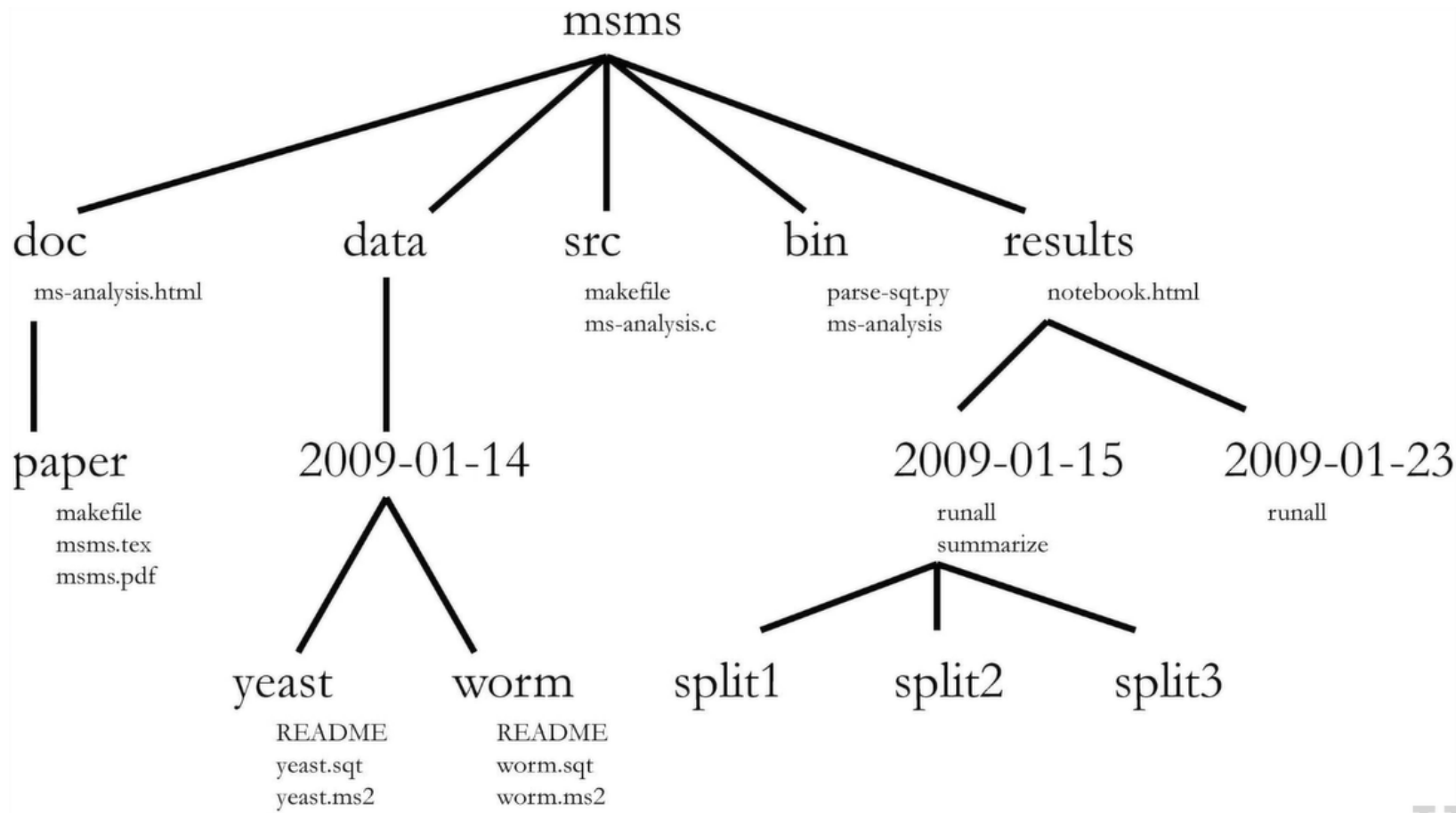
Basis of Experimental Design
- Formulate a question/goal/hypothesis in advance
- Design an experiment in which we perturb the system and then look for changes. The response of the system either validates or invalidates that fundamental hypothesis.
- Consider the experimental condition and suitable controls. We should also look at a system which isn't perturbed to understand whether the changes really occurred in response to the perturbation we introduced.
- Begin to think about your analytical plan (an analytical plan with an end goal)
- Design with sufficient replication, power calculations to help you understand whether or not you have sufficient replication to measure effects of the size that you expect.
- Include randomisation to avoid introducing confounding factors
  - blocking/stratification: arrangement of samples into groups/blocks
  - factorial experiments: testing multiple variables
  - revisit your analytical plan and ask whether you will be able to achieve your goals

2.4.2 Organisation of Research Project
To create simple protocols or personal systems for organizing data that will improve your reproducibility.
To make computational experiments easier to rerun, reproduce, communicate, generate reliable results, and publish.
- Organise files in a shallow directory structure for each project.
- Group project data into subfolders, such as input, data, documentation, configuration files, scripts, temporary output, and final output.
- Include dates and possibly time stamps in all folder and file names where appropriate.
- Make sure you're consistent between files within a project, and between projects, so that you and others don't have to guess what conventions you're using in a particular context.

```
msms
├── doc
│   └── ms-analysis.html
│       └── paper
│           ├── makefile
│           ├── msms.tex
│           └── msms.pdf
├── data
│   └── 2009-01-14
│       ├── yeast
│       │   ├── README
│       │   ├── yeast.sqt
│       │   └── yeast.ms2
│       └── worm
│           ├── README
│           ├── worm.sqt
│           └── worm.ms2
├── src
│   ├── makefile
│   └── ms-analysis.c
├── bin
│   ├── parse-sqt.py
│   └── ms-analysis
└── results
    └── notebook.html
        ├── 2009-01-15
        │   ├── runall
        │   ├── summarize
        │   ├── split1
        │   ├── split2
        │   └── split3
        └── 2009-01-23
            └── runall
```

- Supporting info in subfolders: figures, source, data
- Keep an electronic lab notebook for quantitative research, to record what experiments you're doing in a consistent way. Online wikis, code repositories, such as GitHub, or Bitbucket, or document sharing tools, such as Etherpad, are also effective. Evernote, OneNote, Google Keep, or a simple text file in Dropbox can all work.

Bill Noble's two guiding principles
1. Assume that everything that can go wrong will go wrong. Being organized about your quantitative research projects will make it less difficult if there is a need to rerun every analysis step multiple times.

2. Create your analysis environment assuming that someone else might need to look at it, and understand it at any time. But often, the someone else is yourself, many months later.

2.4.3 Understanding Yourself Later

Is your future self going to be able to reproduce your past self's analysis?

One of the most important parts of being a scientist is being skeptical. Being able to reproduce your own results at the drop of a hat is perhaps just as important as others being able to reproduce them. In the process of convincing yourself the results are reliable, it's usually easiest to make them as reproducible as possible, at every stage, from conception through publication, and revision.

2.4.4 Ensuring that Others Can Continue Your Work

"We are like dwarves sitting on the shoulders of giants.
We see more, and things that are more distant, than they did, not because our sight is superior or because we're taller than they, but because they raise us up, and by their great stature add to ours."
-- John of Salisbury, 1159

All the science we do builds on what has been done in the past. When we do experiments we want others to build on those. The goal really is to try to ensure continuity, to leave a legacy of useful results that others can build upon.

Basic principles to ensure continuity
- Start with the experiment.
    - To describe your hypotheses
    - To clearly lay out the plan that you used to execute the experiment
    - To explain your analytical protocols, so that others can replicate them
    - To describe any potential confounding factors that you might have found in analysing the data or in running the experiment.
- Think about the data.
    - To organise data in a standard format
    - To have data well annotated and consistent

- To know what the data elements are and to make sure those data elements are consistently maintained
- To name the data files in some kind of interpretable manner, including in public repository
- To have adequate metadata to interpret the results
- To provide raw data instead of processed data
- Think about the analytical software
  - To make the code available, with well written comments
  - To ensure variable names are easily distinguished and interpreted
  - To have someone else (or some system) to review and test your code
  - To store code in a repository such as Github, so that everybody knows where it is and what's necessary to run it
  - To get rid of intermediate files in the analytical pipeline
  - To make your software streamlined so you can start with raw data and produce figures at the end
  - To leave behind at least one working example
  - To commit to making code and data easily accessible along with the instructions for how to do the experiments

2.4.5 Providing Workflows and Results
- To create an automated workflow for your analysis, using a tool like Doit, Snakemake, Taverna, Kepler, IPython, Knitr, or even plain old Make-- because inevitably, you'll have to rerun it. Your input data will change, you'll get more data, you'll find a typo, or you'll want to add just that one more test.
- Workflow to automatically capture annotations, like which programs you ran, what their parameters were, and how each output file was created, because otherwise, you won't remember.
- To rerun your analysis in a new computation environment, or in new hands. Or to rerun a very similar analysis process on different data
- To automate the process of handing off results to the journals or databases where they'll go to be published. Data repositories, such as the Sequence Read archive, or the Gene Expression Omnibus, support semi-automated deposition.

Assessment 1

You have been given the data set above from a study that began on January 1st, 1999. Which of the following steps should be done to transform the data set to make it consistent and appropriate for analysis? Check all that apply.

☑ The complications column should be made consistent by creating a numeric binary variable. One option would be making 'No' = 0 and 'Yes' = 1.

☑ Rather than separating individuals by the treatment received, a column with patient ID information should be created.

☐ All individuals with any missing data should be removed.

☑ A column for treatment should be created with entries being either 0 or 1. One option would be making 'Treatment A' = 0 and 'Treatment B' = 1.

☑ A consistent indicator of missing information should be created to replace the '?', 'unknown' and 'NA' cells. One option would be designating the number -9 as an indicator of missing data.

☑ Patient gender should be made into a consistent, numeric binary variable. One example is 'Male' = 0 and 'Female' = 1.

☑ The date of enrollment should be made consistent. One option would be to convert all dates to how many months since the start of the study.

☑ All tumor stage entries should be made numeric: 1, 2, 3, 4.

☑ Height should be converted to either inches or centimeters for consistency.

Assessment 2

In outline/bullet point form, design a directory structure given the following prompt:

You are conducting a GWAS study funded by the NIH Grant #96024. The study began on May 1st, 2018. You received your first set of data titled gwas_snps_06012018_nycny.tsv for a genome wide association study on June 1st, 2018 from a cohort in New York City, New York. On June 3rd, you performed a preliminary round of data cleaning and saved the new cleaned data file. On June 18th, you conducted your first exploratory data analysis including figures and summary statistics as outputs. On June 28th, you wrote a script to conduct a statistical analysis. However, you noticed an error in the script and on June 30th, you wrote an updated version of the script. On September 3rd, you received a new set of data titled gwas_snps_09032018_stlwa.tsv from a cohort in Seattle, Washington. On September 4th, you conducted a round of data cleaning on this newly received data. Then on September 5th, you performed a statistical analysis on both sets of data and saved the output. You started preparing a manuscript on September 15th and presented a draft to your PI. They make several suggestions which include a tweak in your statistical analysis. You updated your code on October 15th, and saved new output on October 16th. A new draft of the manuscript is saved on November 1st. You present this draft to your PI. You save a third draft on November 15th, a fourth draft on November 30th, and a final draft on December 1st.

Be sure to include the names of all directories and files, similar to the figure shown in the video in submodule 2.4.2 Where These Matter. Keep in mind the organization tips presented in the videos.

gwas_nihgrant96024_20180501
data
gwas_snps_20180601_nycny.tsv
gwas_snps_20180601_nycny_cleaned.tsv
gwas_snps_20180903_stlwa.tsv
gwas_snps_20180904_stlwa_cleaned.tsv
source
clean_20180601
analysis_20180618
analysis_20180628
analysis_20180630
analysis_20181015
figures
figures_20180618
output results
summary_20180618
summary_20180905
summary_20181016
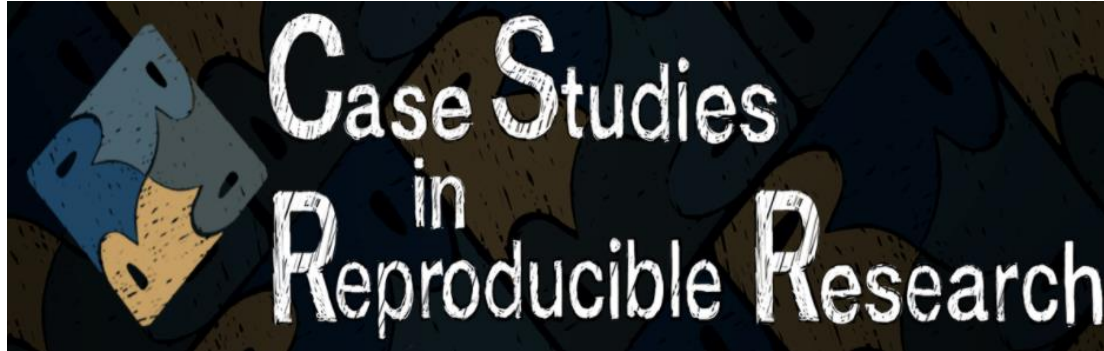**manuscript**
manuscript_20180915
manuscript_20181101
manuscript_20181115
manuscript_20181130
manuscript_20181201

Chapter 3. Case Studies in Reproducible Research

3.1 Introduction to Case Studies
To recognize good and poor reproducibility within papers.
If people find mistakes in published papers, we can correct them and we can advance the field.
As a consumer of the scientific literature, you need to recognize that not all published science is right. Nor is all of it reproducible.
The bottom line is that you as a consumer have to **read everything with a critical eye**. If you're going to build on a published study, you want to start by examining whether or not the published results can be reproduced. If they can, it's a great starting point. If they can't, you probably want to rethink what you're going to do.
Our goal really is to **understand where things can go wrong** so that if you see them and you find them, you can learn how to make your own research more reproducible, more robust, more reliable.

3.3 Baggerly and Coombes
In November of 2007, after a protracted argument with the editors of Nature Medicine, Baggerly and Coombes published a paper in the journal, in which they laid out their concerns.
   1) they could not reproduce Potti's selection of cell lines
   2) the list of genes that were reported in the supplementary information to the Nature Medicine paper on the Nature Medicine website were wrong. And Baggerly and Coombes figured out this was what's called the one-off indexing error. Essentially the lists were shifted by one when handled in Excel.

3) using software and lists of cell lines, Baggerly and Coombes reproduced the published heat maps for six out of the seven drugs
4) for docetaxel, the prime example in Potti's paper, Potti's software only identified 31 of the 50 genes they reported. Some of these may have come from another paper, but not from Potti's data.
5) the leave-one-out cross-validation was not done correctly, but with training and test sets that were mixed. Their software does not maintain the independence of training and test sets, and the test data alter the regression model.
6) the combination of the incorrect gene list and the mixing of training and test sets gave results that were better-than-chance predictions, but in the wrong direction.
7) when Baggerly and Coombes corrected the problems and tried to run the method independently, their results were extremely poor, and they were not better than what one would have gotten taking a randomly selected set of cell lines.

You get stellar results, but they can't be reproduced. And in fact, the data you report, analyzed properly, is no better than doing things randomly by chance.

Baggerly and Coombes Disclaimer:
"We do not believe that any of the errors we found were intentional. We believe that the paper demonstrates a breakdown that results from the complexity of many bioinformatics analyses. This complexity requires extensive double checking and documentation to ensure both data validity and analysis reproducibility. We believe that this situation may be improved by an approach that allows a complete auditable trail of data handling and statistical analysis. We use Sweave, a package that allows analysts to combine source code, in R, and documentation, in LaTeX, in the same file. Our Sweave analyses are available [URL on their website]. Running them reproduces our results and generates figures, tables, and a complete PDF manuscript."

So Baggerly and Coombes actually had a **reproducible pipeline** that others could look at. Their push for reproducibility is reflected in a 149-page supplement to their one-page-long article that shows their calculations, their reasoning, their entire method in great and reproducible detail.

Potti and Nevins provided a (weak) rebuttal:

1) Baggerly and Coombes followed the methods in the paper and in its supplements and in some of the websites that were at Duke. But now Potti comes back and says that Baggerly and Coombes missed some other website (not in the paper) where the method was described.
2) Potti and Nevins admitted to the problems with the gene lists, but with no supporting data claim that it doesn't change the results. You use the wrong genes, but the results don't change. But everything's OK--doesn't make sense.
3) they dismissed the argument about mixing training and test data, and then say it doesn't matter and cite unpublished data. But unpublished data that no one has ever seen refutes an analysis where people have shown the method doesn't work.

Baggerly and Coombes were very concerned because a lot of these (Potti and Nevins) papers had obvious flaws, and yet the methods developed and described were being used in clinical trials. They were being used to treat patients.
Major scientific journals considered this dispute as scientific difference of opinion. And so they said with the publication of the point, counterpoint, the matter was closed.

Lisa McShane and Jeffrey Abrams at the NCI launched their own evaluation of one of the trials. And they themselves had trouble reproducing the predictors. And when they asked Potti and Nevins for the supporting data, Potti and Nevins couldn't produce that data.
Then a letter signed by 31 internationally known biostatisticians and bioinformatics experts that asked the NIH to suspend the trial and conduct an independent analysis because of concerns about reproducibility.

Harold Varmus, the director of the US National Cancer Institute, requested that the Institute of Medicine conduct an **independent analysis** of the omics-based test developed at Duke. The committee found problems with the process at Duke, with the hierarchical culture of science, where junior people couldn't necessarily report problems to more senior people, with the journals that shied away from even considering publishing responses to high-profile papers that were critical of those papers. But most importantly, the committee made recommendations for reproducible research.

For work to be **reproducible**, the authors should make available
1) full data underlying the results,

2) metadata or sample annotation for every sample in the collection,
3) fully described methods for the analysis,
4) freely available open source software used in the analysis
5) fully executable pipeline so people can fully recreate the analysis
6) clean, controlled, independent validation study that reaffirms your findings

3.4 Ioannidis 2009

"'tis better to be silent and thought a fool than to speak and remove all doubt."
-- Abraham Lincoln

Of the 18 tables and figures that they had selected from these 18 high-profile papers, none, not one, could be completely replicated. For many, the methods used for analysis were poorly specified or the software wasn't available or the parameters used in running that software were not specified.
Inability to reproduce the analyses was mostly due to unavailability of data and metadata. Two had no data at all, one had only summary data. For one, it was unclear which data were used. Another had data on a limited set of genes. For one, Ioannidis and his group could not even access the software. Another had problems with gene annotation. And in the last two, the lack of raw data and incomplete description of the analyses thwarted Ioannidis' effort. Of the eight that were partially reproducible, there were many issues that prevented the results from being fully validated.

For the papers that could be reproduced, the method sections that were published as online supplements included a clear description of the path that was taken in the analysis, the steps, the software choices with available code, the parameters and decisions that were made, the input and output files.

3.5 Case Studies Wrap up

"No one who cannot rejoice in the discovery of his own mistakes deserves to be called a scholar."
-- Donald Foster
Face it, we all make mistakes. What we want to do is try to avoid them.
Ensure that the data/metadata, methodology, analysis code and software that you use in your analysis are available, and hopefully reproducible.

Although it's hard to check what the samples are that people use, there's one simple annotation we can check, whether the sample comes from a male or female. After all, human males have Y chromosomes and females don't. So if we take Y chromosome genes, extract only gene expression data for those genes and do a simple principal components analysis. And then color that PCA plot by sex, we should see a clear separation between the males and the females.

Y Chromosome PCA Plot For Four Studies of Alzheimer's Disease
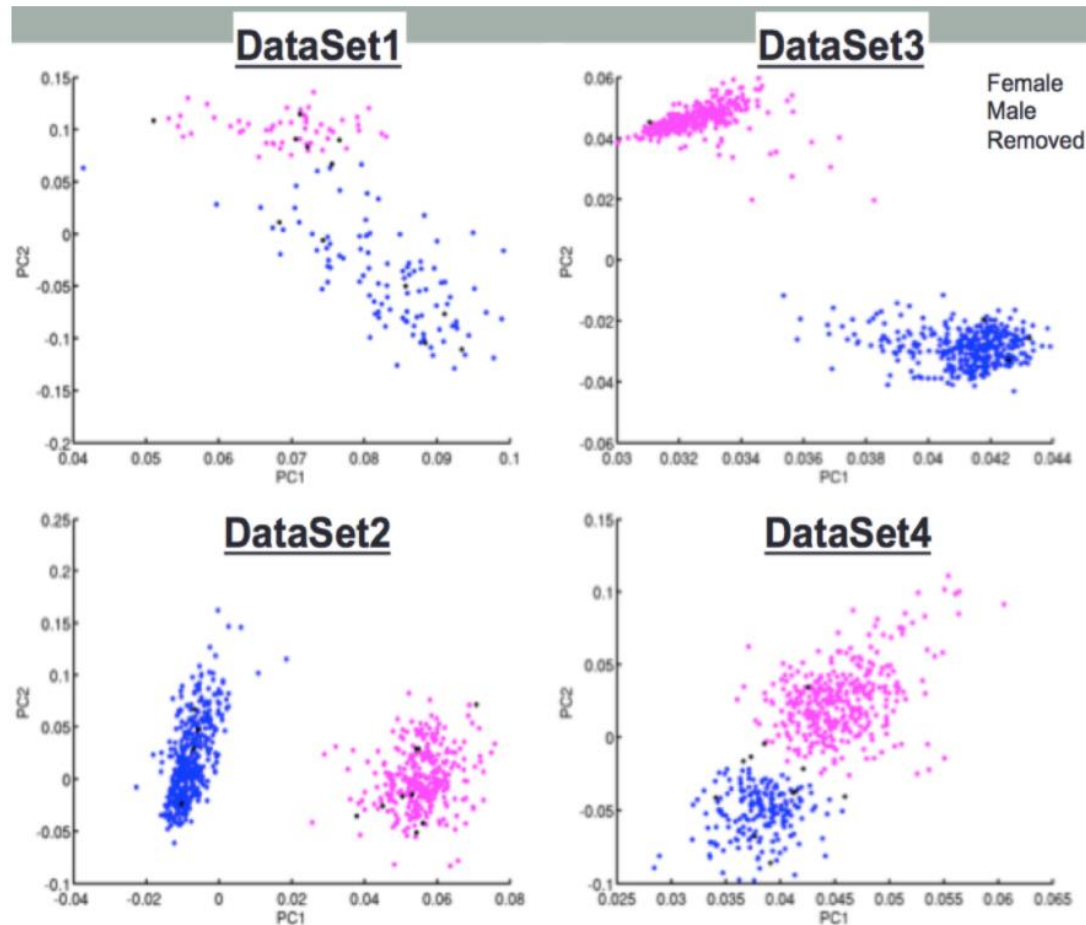
Misidentification of Colorectal Samples
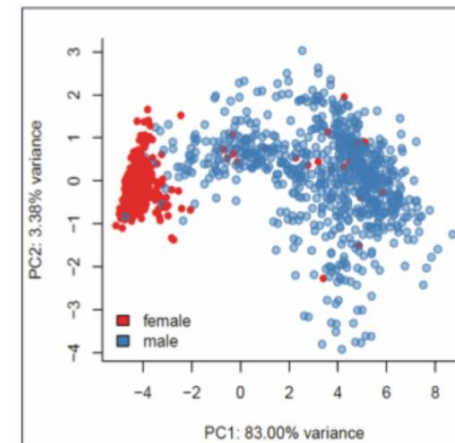
Camila Lopes-Ramos performed PCA on Y chromosome gene expression for pooled data on 1376 samples from five studies.

She excluded 85 samples, or 6%, as misidentified by sex.

Russian proverb: "trust but verify".

You trust what's in these public databases. There are published studies. But before you analyse them, you need to verify what's there. You want to assure that the data you analyse is correct. And the best way to do that is to assure that the data you publish is correct.

3.6.1 Introduction to Reproducible Reporting Science
"Those who agree with us may not be right, but we admire their astuteness."
-- Cullen Hightower

3.6.2 Journals and Reproducible Research
For a research to be reliable, it should be **reproducible**, **robust**, and **transparent**.

Principles to facilitate reproducibility
- encourage rigorous statistical analysis, have a mechanism to check the accuracy of those submissions, example p-values should be supported by a table (in the supplement to your paper) that includes the p-values you calculated, and to include the software and data that you used to calculate them. Can use tools like Sweave, Knitr, and Jupyter are so valuable as a means of documenting what you've done.
- transparency in reporting. There should be no limit or generous limits on the length of the method sections and papers, which could include extensive online supplements, while at the same time encouraging efficient and clear presentation of the methods to ensure that the reviewers had an opportunity to do a thorough examination of what the papers described.
- Use of checklists during the editorial processing to ensure the reporting of key methodological and analytical information is available to reviewers and readers. Example, method or analytical design and all of the software versions and parameters you use, as well as information about the availability of all the data and metadata. Also include an executable document as a supplement.
- Statistics to be fully reported in the paper, including the statistical test used, the exact value of n, the number of samples looked at, the definition of center, dispersion and precision measures such as mean, median, standard deviation, et cetera.
- to require that authors state whether the samples were randomized, and to specify the method of randomization, at a minimum, for all animal experiments.
- to require authors to state whether an appropriate sample size was computed when the study was being designed, data sets and to include the statistical method for computation. Should include some description of how the sample size was determined.

- to require authors to clearly state the inclusion and exclusion criteria that were used for exclusion of any data or research subjects, and to include any similar experimental results that were omitted from the reporting for any reason, especially if the results don't support the main finding of the studies.
- to adhere to data and material sharing policies, and specifically, that the data are available through publicly supported databases.
- to encourage the presentation of all data values in machine readable form in the paper or in its supplementary information, and suggested requiring material sharing after publication.
- to encourage the sharing of software and require at the minimum a statement in the manuscript describing if the software is available and how it can be obtained, but they didn't mandate the availability of open source tools.
- to consider establishing best practice guidelines for image based data. For example, screening for image manipulation.

What we need to do as scientists is to assure that our work is reliable, and can be reproduced by others.

3.6.3 NIH Guidance on Reproducible Research

National Institutes of Health, NIH, recognized they were investing significant amounts of money on projects for which the data generated were not available, the laboratory resources were not accessible, and from which the published results were not reproducible.

NIH has changed requirements for grant applications in January of 2016. These changes were designed to address the issue of reproducibility, starting with good experimental design and going through the reporting and sharing of resources.

The instructions and revise review criteria focus on four areas deemed important for enhancing rigor and transparencies:

- the scientific premise for the proposed research - to describe the general strengths and weaknesses of that prior research being cited as crucial to support the application. Consider discussing the rigor of the previous experimental designs, as well as the incorporation of relevant biological variable and authentication of key resources.
- rigorous experimental design - the strict application of the scientific method to ensure robust and unbiased experimental design, methodology, analysis, interpretation, and reporting of the results.

Emphasise how the experimental design and methods proposed will achieve robust and unbiased results.

- consideration of sex and other biological variables - including sex, age, weight, underlying health conditions, are often critical factors affecting health or disease. In particular, sex as a biological variable has been frequently ignored in animal study designs and analyses, leading to an incomplete understanding of potential sex-based differences in basic biological function, disease processes and treatment response.
- the authentication of key biological and/or chemical resources - include, but are not limited to, cell lines, specialty chemicals, antibodies, and other biologics. Data is a key biological resource. Provide a detailed description of the data resources, and provide a plan for how to validate those resources before starting any new analyses or methods development.

Chapter 4. Data Provenance

4.1.1 Introduction to Data Provenance
Provenance = (Latin) come forth = history
Data provenance = any data source, format, time and date stamp, and detailed history of handling.
- Past. Where did your data come from?
- Present. What did you do with it?
- Future. Where is it going to live after you publish it?

Data provenance can be assisted by electronic aids, such as computational project lay out, scientific workflow systems, electronic lab notebooks, and standards for file naming and recording.

Scope of this module:
- Requirements for journals and funders
- Repositories file storage standards, public databases, and documentation requirements to satisfy these data provenance guidelines.
- Data security (the need to ensure that only authorized users can access data) and privacy (the need to segregate sensitive data during research)

Maintaining **past** provenance, where your data came from?

- technical considerations: contact information, file integrity history
- high level concepts: understanding a data generating technologies strengths and weaknesses

Maintaining data provenance in the **present**
- maintaining electronic records indicating what you did, where your files are, and how each file relates to others. Example, summary statistics that reduces a big data set down to just a few numbers
- tracking data contents: metadata associated with file formats and histories,
- tracking analyses: what commands were run

Maintaining **future** provenance
- data and code sharing repositories. Many data type specific and type independent repositories exist that guarantee long-term availability of research data and products for future studies and replication.
- maintaining research software: future user support, bug fixes, new releases, recording and describing these changes so there's transparency for other researchers.

4.2.2 Experimental Design Stage I
Software development team:
40% time spent on specification and design, planning elements (background study, research hypothesis)
20% time spent on coding programming
25% time spent on testing and debugging
15% time spent on implementation and production support

Elements to consider:
- ensure that available data can answer your research question
- carrying out formal power calculations when possible
- ensuring that population structure or confounding won't reduce the effective number of data points you have available
- throwing out extra data that won't be useful in addressing your questions
- ensuring that you'll have one or more independent data sets available in the future for validation.

- If you have the luxury of designing experiments to generate completely new data, you might need to balance time or monetary budgets against the number of data points generated or conditions tested

Record everything about your sources of data--who carried out each experiment, at what date and time, how to contact them, and model and make information for any equipment use, including batch numbers of chemical reagents.
During project planning take an extra day or two, or a week or two, if you need it, and build an analysis infrastructure that will make future computational experiments easy and reproducible.
From the technical to the conceptual, pre-research planning can literally take days or weeks to establish, but for a data analysis that might take months or years of your career, it's worth it.

4.2.3 Experimental Design Stage II
During a data science research project, additional computational and conceptual steps can be taken to ensure reproducibility and reliable data provenance.

Implement provenance tracking infrastructure
Ensure that you have provenance tracking infrastructure for at least three different aspects of your research
- analysis commands that you run as part of your main workflow
- data that is produced by these or other commands
- a lab notebook or other informal provenance capture mechanism for everything else, the miscellaneous little tasks that crop up during research

Formal data provenance capture
- **transactional** provenance: Many database infrastructures will capture at least the basics of each query or update by default, including a **time and user stamp**
- **workflow-level** provenance: captures the conceptual **data transformations** we've mainly discussed, what you want to know as a reproducible data scientist
- **database-level** provenance: captures each **changed** individual data **records**, easier to automate
- Open Science Data Framework, or OSDF, provides an API that links files in a standard file system to a provenance schema in an accompanying database.

- myGrid, executes workflows in a predefined standardized language, and returns the resulting data products with provenance using web services.
- IBM Data Science Experience is a commercial environment that layers containerization technologies over a custom API that links provenance to underlying file system, database, or cloud services.

Choice of data provenance environment
- positive controls: Include at least one, if not several, small synthetic data sets for which you know the right answer. If possible, add an automated check that ensures the expected result is produced by the associated workflow step.
- negative controls: Generate random data, null data, or corner cases containing unusual input, and test that the associated analysis doesn't produce anything interesting. Statistical significance is only exciting when it applies to good data and not to anything at all.
- visual inspection: At some point during your analysis process, take a look at the data going into and coming out of each computational step. It's surprising how often an analysis process isn't doing quite what you expect, and it's easy to see when and why by just taking a look at a tractable data subset or snippet with your own eyes.
- use a standardized database which will assign a stable locator to your data set--a URL, accession number, Digital Object Identifier (DOI) or other data record.

Assessment
Q: Identify the following as either a positive or negative control.

This is a matching exercise. Drag the elements on the left into the ones on the right to create a match. You can match items with more than one other item, but you should choose the best match if you can. Click on the items for a better view if you need it.

Subject:
Workflow-level provenance
Database-level provenance
Transactional provenance

Example:
Capturing intermediate products and results in an iterative process to build a parsimonious phylogenetic tree
Data is stored in a relational database with two relations R1 and R2 such that data is handled at the tuple level of information
Recording user and timestamp information about when a query was processed

Show answer:
Explanation
A. Positive control, B. Negative control, C. Negative control

4.2.4 Tools and Standards
To try things out, less formal aspects of data provenance management and reproducibility
A good way to play around with data informally is using a literate programming environment such as Jupyter or knitr with RStudio.
Another even less formal alternative is to use an environmental capture process such as the terminal script command, a screen capture session, or even a plain old digital recorder if you're discussing an analysis out loud.

- be consistent in your own file system structures, file naming conventions, and documentation.
- Use informative consistent filenames.
- Extensively document your code.
- Make sure each script has a header indicating who wrote it, when and why.

- Notate any particularly complex piece of code with an explanation of what it does.
- provide at the least a README file for your main data inputs and outputs that describes what they contain, in what format, and why they're important.

4.2.5 Workflows
data transformation = scientific workflows
computational tools: a wide variety of services available that will help you systematically record exactly what data analysis steps you carry out on which data inputs and outputs.

a scientific workflow system captures the provenance of data transformations in a standardized format:
- some combination of the data products themselves;
- associated provenance metadata;
- and the transformations between them, which can include scripts;
- executable dependencies;
- cloud or web services;
- manual processes in graphical environments.

If you're playing around with data to figure out what's possible, but you want the process to be somewhat well-recorded and reproducible, use literate programming.
If you're assembling a production data science environment to reproducibly generate publication-ready products or outputs for sensitive application areas, like health care, use a more formal scientific workflow environment.

The basic rule is rather than running commands by hand on the command line, write them into a single driver script that you run instead. This driver script can call other scripts, command line executables, or move around data inputs or products. If you centralize this process, even informally, it can then easily be checked into a revision control repository and change-tracked as well.

A better step up, however, is to use a dependency driven framework as a minimal scientific workflow environment. These will not typically track data provenance explicitly, but they will track data

transformations, parameters, and make your life easier by automatically remembering analyses that have been running, allowing you to repeat them when needed and not repeating them when nothing has changed.

At the heaviest-weight end of the scale, formal scientific workflow environments typically provide a graphical interface for editing data transformation tasks; import and export of workflows and standardized formats, such as the common workflow language; and task dispatch to a variety of execution engines that might be either local, grid, or cloud based. Common examples in this family include Kepler, Taverna, Arvados, Cromwell, Ergatis, Galaxy, and, again, many others.

Planning for Reproducible Data Science
Before Analysis

- Research Questions / Hypotheses
  - Should have several in mind
  - Should relate to the knowledge gap are you trying to fill
  - Should be coherent and focused
- Data Acquisition
  - Choose an experimental design (if appropriate)
  - Consider population structure and confounding
  - Ensure you'll have data available for validation in the future
  - Balance time and monetary budgets against the number of data points generated or conditions tested
  - Carry out power calculations
  - If limited to a particular data set(s) that don't provide information needed to answer your research question, you may need to change your research question or hypothesis
- Data Provenance
  - Data Source(s)
    - Record who carried out each experiment, at what date and time, how to contact them
    - Record the model and make information for any equipment used
  - Data Storage and Management
    - If you expect your project to include big data, by any definition - volume, velocity, or variety - plan up front for how you'll manage it
    - If working with big data, consider a distributed file system and parallel processing, as well as streamed processing that can save progress, start, and stop as needed

- If your data change frequently, include an explicit annotation process, automated if possible, that captures where new files are coming from, at what time, and who's responsible for them
- If you're working with many different kinds of data, consider using a data store such as the Open Science Data Framework that provides detailed typing, metadata, and cross-file integration
  - Analysis Plan
    - Consider what could go wrong and how can you avoid it
    - Ensure you have the resources in place (necessary software, sufficient computing power) to carry out the whole project
    - Ensure re-running an entire analysis will not be difficult to carry out
  - Analysis Infrastructure
    - Choose a formal scientific workflow environment
    - Find and install software (text editor, development environment, task tracking, documentation, and communication tools) that will facilitate your day-to-day work
    - Set up a revision control repository for the project (GitHub, BitBucket, etc.)
    - Create a directory and folder structure
      - Store files in a consistent and intuitive manner
      - Include a README file that contains information about all other files in your repository

During Analysis

  - Provenance Tracking Infrastructure
    - Ensure you have infrastructure for at least the following three aspects of your research
      - Analysis commands you can run as part of your main workflow
      - Data that is produced by these, or other, commands
      - Lab notebook for everything else
    - Use a supplemental provenance annotation system and/or environment for formal data provenance
      - JSON, XML, W3C PROV, Open Provenance Model
      - Taverna, Open Science Data, MyGrid
  - Workflow Documentation
    - Practice literate programming and extensively document all code
    - Choose easily interpretable variable names
    - Make use of version control
      - GitHub, Google Docs, Etherpad, etc.
    - Make code and workflow as automated as possible
      - Use driver scripts
      - Use platforms like R Markdown, Jupyter notebook, etc.
    - Incorporate positive and negative controls throughout the analysis
    - Consider using a shared file service

- Dropbox, Google Drive, OneDrive, etc.
  - Data Quality Control
    - Document the actions performed to achieve a clean data set
      - Use workflow environments like Doit, Luigi, Taverna, Galaxy, etc.
  - Verify Analysis Methods
    - Avoid overfitting using a variety of methods
      - Cross-validation
      - Regularization
      - Bootstrapping

After Analysis

- Verify Analysis Results
  - Implement positive and negative results checks
  - Conduct simulation studies
  - Sensitivity analyses
- Implementation Checks
  - Ask at least one person to re-run your analysis and gauge how much effort it takes to implement
    - Include a README file
  - Ensure at least one person knows the basics of what data and methods are stored and how they are documented
  - If possible, re-run your analysis using new and/or different software
- Methods and Documentation for Publication
- Code Publication
  - Ensure your code is well documented and commented
  - Consider who will be maintaining your software in the future and how
  - Consider how your code will be available
    - GitHub, BitBucket, etc.
    - As part of the supplementary material for a journal article
    - Your own private web site
- Data Publication
  - Cite your data
  - Provide a detailed README file
  - If needed, anonymize or de-identify data
  - Consider who will have access to the data
  - Consider if your data will be available long-term and how (where it will be stored)If providing raw, anonymized or de-identified data is not possible (i.e. proprietary data), provide a synthetic data set that can be used in its place
    - GitHub, Dataverse, Figshare, etc.
    - As part of the supplementary material for a journal article

- ▪ Your own private web site

Summary of best practices

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
- 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
- Use version control
  - ○ Slow things down
  - ○ 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
- 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)
- 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 seed() command
  - ○ 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

DONT's:
- Do things by hand
  - Editing spreadsheets of data to "clean it up"Editing tables or figures (e.g. rounding, formatting)
    - Removing outliers
    - QA/QC
    - Validating
  - 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
- 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
- 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

4.3.1 Journals and Reporting
Journal requirement
accession numbers are included for all raw datasets associated with a publication pointing to public repositories
- Link to raw data in public repositories using the appropriate accession

- numbers, provided summary tables in a supplement,
- and host useful process data products that allow future work either locally or in general purpose public databases.

Open access data still require attribution for any future use
- can even be licensed as necessary to preserve intellectual property,
- allow mixed scientific and commercial use,
- track multiple data products as needed.

4.3.2 Mechanisms for Reporting
Mechanisms for ensuring reproducibility during data and, later, methods publishing
- in an article's supplementary files, simple formatted summary tables, excel file
- host data on your own private web site, need to maintain availability for many years, or store in cloud
- use version control repositories, such as GitHub or Bitbucket
- (best option) in public repositories, either data-type-specific or general, example Figshare
- University's library-supported long-term general data repositories
- public repositories example Figshare
- most journals do not currently support mechanisms for publishing consistently reproducible analysis methods

The best reproducible analyses will include a full pipeline that generates each dataset, value, and figure in a manuscript, version controlled, and made publicly available at the time of publication.

4.4.1 Data-type-specific Repositories
Data type specific repositories that support reproducible data sharing
- nucleotide sequencing: DNA and sometimes RNA
- Sequence Read Archive (SRA) maintained by the NCBI at the NIH
- European Nucleotide Archive (ENA)

- International Nucleotide Sequence Database Collaboration Network (INSDC) - minimal information about any sequence or MIxS: XML based information links studies, samples, experiments, runs, analyses, and information about the repositories submission process for each data point within a data set
- Metabolomics repositories
- Proteomics repositories
- ProteomeXChange
- Network Repository

A good practice for reproducible research is when you're ready to publish to search out the right one or few databases for your experiments and data types. It's often safe to assume that someone's already created one and you can check to make sure that it's stable, well-documented, and well-maintained before submission.

4.4.2 General Repositories
Data-type-specific standards
- Generalist data repositories – Dryad, Figshare, Harvard Dataverse, Database of Genotypes and Phenotypes (dbGaP)
- controlled access data (with private data such as age, gender, medical history)

In brief, data submitters to dbGaP must provide, in addition to their data, a description of how and under what circumstances it can be used. Applicants must provide a brief explanation of how their research will use the requested data. This controlled access sharing provides a flexible mechanism in which a highly secure infrastructure is already in place to publish a wide range of data types that might require extra security even after publication.
Data set specific journals, such as Nature's Scientific Data, often have convenient mechanisms that specifically integrate with data repositories.

4.4.3 Code

Code is an important data type, too, and that repositories that deal mainly with code can, under the right circumstances, be a good place to version and share data as well.

Repositories such as GitHub and Bitbucket are modern revision control systems, databases originally developed to track changes in source code made over time and among multiple developers

4.4.4 Documentation

Documents that you write describing your research, be they electronic lab notebooks, manuscripts, or simple readme(s) accompanying data sets, can be among the most important types of "metadata."

For different phases of a project, and for different types of documentation needs, avoiding conflicts between multiple editors, preventing multiple copies from being made (accidentally or intentionally):

- use of a code revision control repository for documentation such as GitHub or Bitbucket: ideal for long term documentation with versioning
- Collaborative editing environments, such as Google Docs or Etherpad: edited by multiple users simultaneously
- shared file service such as Dropbox, Google Drive, or OneDrive: for highly formatted documents or publication ready files

Assessment
Blank Advanced Problem

In the following scenario, identify the problem and propose a possible solution.

You are interested in investigating the association between age and developing toxemia during pregnancy for a group of mothers. After receiving the data set, you realize an indicator of toxemia was not recorded. However, the data set does include the variables that indicate the presence or absence of hypertension, diabetes and edema.

Show answer:
Explanation
The data do not provide the necessary information needed to answer the scientific question. A possible solution would be to come up with another, similar question that you can answer with the data available. For the maternal health example, you could use the information available to create a likelihood of toxemia score for each mother and study the association between that score and age of the mother.

4.5.1 Data Privacy and Security
Data privacy = sensitive and personal info that needed protection
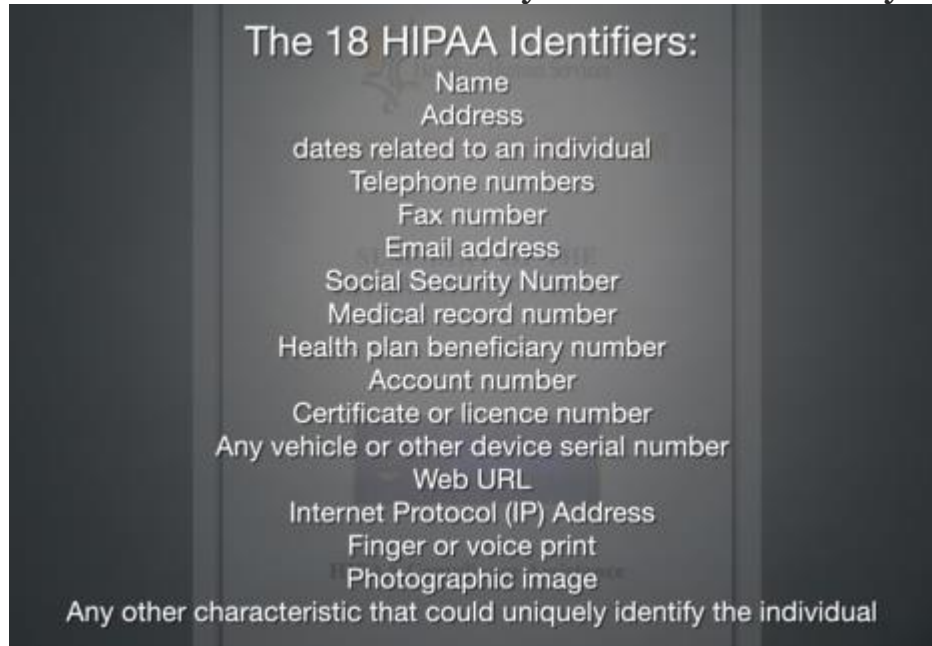Data security = mechanisms for protecting data

In the life sciences research, your data or metadata might both include **sensitive** material, such as genetic information or health records) or other **personal** information, such as physical location or social networks. Such data must typically be specially protected by the researcher and shared only with other users authorized for research purposes. The unintentional release of sensitive data to other parties is referred to as a **breach**, and under many circumstances must be reported to agencies such as university oversight committees or federal funders.

Private data = Protected Health Information (PHI). PHI is typically protected by the masking of any unique identifiers that could be traced back to an associated human subject, such as name, address, or social security numbers, in addition to less obvious risks like dates or URLs.
Anonymization of such records is one step that can protect them during research, referring to any method of manipulating the data set that masks these unique identifiers or prevents them from being correctly linked

to their associated medical records. **De-identification** is one of the most common forms of anonymization, in which all such identifiers are removed completely.

Health Insurance Portability and Accountability Act, HIPAA

The 18 HIPAA Identifiers:
- Name
- Address
- dates related to an individual
- Telephone numbers
- Fax number
- Email address
- Social Security Number
- Medical record number
- Health plan beneficiary number
- Account number
- Certificate or licence number
- Any vehicle or other device serial number
- Web URL
- Internet Protocol (IP) Address
- Finger or voice print
- Photographic image
- Any other characteristic that could uniquely identify the individual

Data that have been manipulated to protect such private information are referred to as **coded**, and it is typical for the original identified data set to be handled by an external party, such as a hospital clinical coordinator, and completely unavailable to researchers.

data security are the mechanisms for protecting them:

- Electronic: encryption, user account restrictions, strong passwords, or limiting a data set's network exposure completely.
- Physical: paper documents, limiting researchers or other individuals who have access to physical files or might see information on a screen, or in extreme cases, actively preventing social engineering attacks or intrusion attempts.

Private data and the security process should not impede reproducibility, although they certainly create additional challenges.

Many private data sets are intermediate, in that they cannot be usefully formatted to allow public sharing, but can be made available as controlled access data, anonymized but otherwise unmodified and available to any researcher who provides a valid research purpose and ensures that they will in turn protect the data appropriately. This type of controlled access sharing is facilitated by repositories such as dbGaP, the Database of Genotypes and Phenotypes.

Example:
Medical records collected at clinical sites within a hospital
De-identified anonymised subsets of the collected records released to researchers
Less sensitive info integrated with data, but is maintained in a reasonably secured location.
Publications might include aggregated info from the sensitive data
Non-sensitive data subsets are shared in a supplementary table or public repository, the more sensitive subsets might be distributed through dbGaP or another controlled access interface.

4.5.2 Privacy
Types of privacy management

- The data can be **completely removed** if possible without disrupting the research process
- **Anonymisation**, for example, to replace dates such as birthdays with a numerical value for age since the latter is far less unique than the former. Likewise, the date of a medical procedure might be replaced with the number of days since a study began. Unique identifiers such as names or Medical Record Numbers, MRNs, can be replaced with completely random code strings. Second-tier anonymized data sets still need to be secured and distributed only through controlled-access mechanisms.
- Sensitive information that must be maintained to carry out the desired research, but for which availability to the general public or other unintended recipients would cause material harm to human participants. The execution of such projects is typically **actively monitored** by an organization, such as an Institutional Review Board, or IRB, who will likely also monitor how outside researchers can interact with the data set for reproducibility purposes in such a case.

Data breach by de-anonymization

Netflix prize challenge in mid-2000, Netflix released a collection of several million movie ratings generated by thousands of real users. The challenge was to develop a machine learning algorithm to suggest new movies that a user would like, based on past rating preferences. The ratings were thus anonymized, but not de-identified, since the research question required the association of multiple records with individual specific users.

However, the shared data set was not entirely well-protected by comparing the anonymized ratings with those from a public source, the Internet Movie Database, or IMDB. These publicly available records were not anonymous. And in cases where an IMDB user had rated movie overlapping with those in the Netflix data set, their name could be quite accurately associated with the remainder of the Netflix ratings.

Unintentional release of names associated with psychiatric disorders, medical risks, sexual history, or other sensitive topics can risk an individual's livelihood or well-being.

If you reveal information about your genome, you're also unavoidably revealing information about your relatives. If you consent to share research information about, say, your genetic medical risks, does that mean your parents or siblings do too?
National Human Genome Research Institute, or NHGRI, provides specific guidelines for human DNA and other molecular data sharing, which is typically treated as requiring controlled access, even if there are no obvious immediate risks.

Assessment
Q: De-identify and clean the data contained in the table below.

Name	Date of Birth	Sex	Address	Weight	Height	Heart rate
John Adams	10/30/1735	Male	Quincy, MA	178	5'11"	85
Benjamin Franklin	01/17/1706	Male	Boston, MA	211	5'10"	90
John Jay	12/23/1745	Male	Bedford, New York	180	5'8"	81
Alexander Hamilton	01/11/1755	Male	New York, New York	179	6'0"	79
Thomas Jefferson	04/13/1743	Male	Charlottesville, Virginia	170	5'9"	92

James Madison	03/16/1751	Male	Orange, Virginia	175	6'1"	78
Abigail Adams	11/22/1744	Female	Quincy, MA	125	5'4"	83

Answer:

Patient	Age	Sex	Location	Weight	Height (inch)	Heart rate
ID0001	65	1	1	178	71	85
ID0002	94	1	1	211	70	90
ID0003	55	1	2	180	68	81
ID0004	45	1	2	179	72	79
ID0005	57	1	3	170	69	92
ID0006	49	1	3	175	73	78
ID0007	56	0	1	125	64	83

Change subject name to number.
Change date of birth to age.
Change Sex to be coded 0 & 1.
Change location to be coded: MA is 1, NY is 2, Virginia is 3.
Change height to inches.

4.5.4 Security

Physical security - Ensure that physical documents are locked as necessary, accessible only by appropriate research personnel, and that physical access to electronic infrastructure, such as server rooms, is limited.

Electronic security - apply to the primary copy of a sensitive data set, also to any (controlling

such multiple copies when necessary) copies of it made for collaborators, reproducibility, or even just backups. Along with mechanisms such as on disk encryption, file access control lists, authenticated network access, two factor authentication, and even limiting network access altogether.

Harvard University Information Security Policy

Tier one information is public, including any published documents or data sets, information derived from public sources, or data about harmless non-human research subjects, such as microorganisms, public statistics, weather, and so forth. There is no restrictions on tier one data storage or sharing.

Tier two information is protected essentially by choice. For example, research that is not sensitive but which is not yet published. Security measures for tier discretionary and generally just common sense. Keep files in an access controlled location, use reasonable passwords, and don't share them. Such information can be referred to as confidential, but not necessarily sensitive.

Tier three includes de-identified protected health information, financial records, or records that are personally identifiable but benign-- say employment history. Security requirements become fairly rigorous at this tier. It may only be stored in encrypted locations, copies may not be made unless there are equivalently secured, servers hosting tier three data must be professionally maintained and updated, an active review must be periodically carried out to ensure that the security is maintained. This is also typically the tier at which breaches are required to be reported.

Tier four information, for example, includes the types of personal sensitive information you associate with data breaches in the popular press-- credit card numbers, social security numbers, or in the case of research, protected health information, genetics, or identifiable financial records. Someone could be actively materially harmed if such data were released and mechanisms for ensuring their security include preventing them from ever being on local storage. They can only be carried on encrypted external media. All access of any type should be logged and associated paper records must be destroyed after use.

Tier five information is rare in any research, but extremely sensitive. Identifiable medical records including psychiatric or sexual information, legal documents, or history, or active financial transactions. Tier five data generally cannot be accessed online at all-- only on network disconnected machines. And even then, only under some type of external supervision.

But even for challenging situations, mechanisms do exist to ensure that other researchers can, under controlled circumstances, access data for future studies and replication.

Assessment
Blank Advanced Problem

This is a matching exercise. Drag the elements on the left into the ones on the right to create a match. You can match items with more than one other item, but you should choose the best match if you can. Click on the items for a better view if you need it.
A) Identifiable medical records that include age, sex and mental health details of all patients at Mass General Hospital
B) A data set including the date, time, duration, location and category of tornadoes in Oklahoma in 2011.
C) Identifiable financial records including credit card numbers and social security numbers of people living in Los Angeles, CA.
D) The de-identified employment history of all adults 18-70 in Massachusetts.
E) A data set detailing microorganisms used for not yet published research.

Answer:
A = Tier 5
B = Tier 1
C = Tier 4
D = Tier 3
E = Tier 2

Chapter 5. Statistical Methods for Reproducible Science
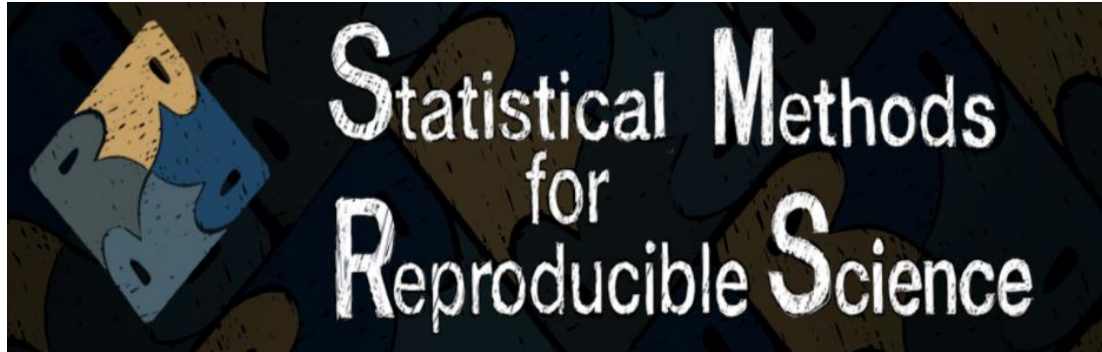
5.1.1 Introduction to Cross Study Validation of Prediction Models
Precision medicine is becoming more and more relevant for making decisions. As for example, we can predict, based on biomarkers, which patients will respond to specific treatments. Other prediction problem involve life expectancy and prediction allow us to discriminate long survival versus short survival.

Key questions:
- the performances of different machine learning tools I can use for doing prediction. For example, support vector machine, random forest, or other machine learning tools.
- to understand not only which one of them perform better when the training and the validation data set are identical
- to visualize differences in performances when there are heterogeneity that characterize different data systems can be used for validation
- examples of validation metrics, coefficient of determination, Brier Score, Area Under the receiver operating characteristics Curve (AUC), concordance in C-statistics in survival analysis

5.1.2 Statistical Methods Introduction
The goal is
To understand what are the best practices to evaluate prediction models.

To understand when can we say that a prediction model is reliable and that the results that we provide, its performance, which might be communicated on scientific journals or might be sufficient for important application

To understand how likely that the same levels of performances will persist in the future when experiments will be replicated and when the prediction model is going to be used in different applications or across different contexts.

5.2.1 Coefficient of determination
R square = coefficient of determination = 1 ➔ the model M is accurate in prediction

5.2.2 Brier Score
Brier Score = the simplest validation metric that we have when we want to predict binary outcomes

5.2.3 Area Under the Curve (AUC)
AUC = the area under the operating characteristic curve
= to measure the accuracy of prediction models with binary variables

5.2.4 Concordance in survival analysis
Definitions and estimation procedure and expression for concordance indices in survival analysis

5.2.5 Motivation for Cross Validation
Possible algorithms that are suggested by cross-validation and cross-study validation.
These different metrics suggest us that if we only look at cross-validation, once again, the conclusion can be **over-optimistic**, just because differences in the distribution of predictors and outcome of interests across studies, across populations, can decrease the performance of models that are trained on a single study. And when we use the same environment, the same study, for doing both the training and the validation, even if we avoid bias by using cross-validation, there still might be over-optimism, just by the fact that different environments come with different distributions of our x and y variables.

5.2.6 Cross Validation
To obtain variability estimates, and uncertainty summary using Bootstrap for validation summaries

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

In a prediction problem, a model is usually given a dataset of known data on which training is run (training data set), and a data set of unknown data against which the model is tested (testing data set). The goal of cross-validation is to define a dataset to "test" the model in the training phase (i.e., the validation data set), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, etc).

5.2.7 Bootstrap

To use Monte Carlo simulations to obtain variability estimates, and uncertainty summary using Bootstrap for validation summaries.

In statistics, bootstrapping is any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates.

Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution.

One standard choice for an approximating distribution is the empirical distribution function of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of resamples with replacement, of the observed dataset (and of equal size to the observed dataset).

5.3.1 Simulations

to evaluate possible algorithms to train prediction models

Start by first developing a simulation model-- just code that allows us to iteratively as many time as we want with a computer to simulate the entire library of data sets. All of them with their predictors and covariates. And the reason we want to iterate simulation is because this facilitates the evaluation of algorithm that deliver prediction models.

5.3.2 Clustering

We can use libraries of studies in order to obtain multiple prediction models, Each of them trained with a distinct study, and then validate them in heterogeneous studies. And that's how we obtain our matrix of validation summaries.

Statistical techniques that have been successfully used in meta analysis can be directly applied and modified in our setting, in which we try to interpret validation matrices. And we try to understand what are the best algorithms and machine learning tools in order to develop accurate prediction models.

The meta analytic approach not only indicate as best the approach, but it also allows us to quantify the level of uncertainty.

Chapter 6. Computational Tools for Reproducible Science

6.1.1 Introduction to Computational Tools for Reproducible Science
In this module, we are going to present computational tools that make reproducible science easier.
We are going to focus on:

(1) Reproducibility and code
- Can an editor / IDE be helpful?
- What is version control? (to keep track of all the changes and the edits and the collaborations that you have when you download code)
- How to keep track of bugs and issues?
- What is GitHub?

(2) Reproducibility and data
- What is a data repository?
- Can data be under version control too?
- What is Harvard Dataverse?
- How to use an API (Application Programming Interface) to upload/download/modify data on a remote repository?

(3) Reproducibility and document generation
- What is literate programming? (allows you to write together a document and the analysis that you are going to perform in the document. So you have together the text of the document, the code to generate the statistical analysis, and the plots. And you generate everything at once.)
- What are the benefits of using R Markdown? (tools such as knitr)
- Can R Markdown be used with Python?
- What are Notebooks? In particular, what is Jupyter?
- How can I use makefiles to automate tasks? (a combination of commands that allow you to put all the pieces of an analysis together)

6.1.2 Introduction to Contributors
- Erik Surface, Senior Computational Engineer, Department of Epidemiology, Harvard T.H. Chan School of Public Health
- Christopher Gandrud, Research Associate, Institute for Quantitative Social Science, Harvard University
- Mercè Crosas, Chief Data Science and Technology Officer, Institute for Quantitative Social Science, Harvard University
- Simon Adar, CEO, Code Ocean, New York City

6.2.1 Introduction to Editors
RStudio
IDE-- an integrated development environment

6.2.2 Introduction to R and Rstudio
Studio is a very useful IDE for R
but there are other choices. In particular, a very popular one is Emacs-- or sometimes Aquamacs on OS X-- which can be used with something called ESS, which means Emacs Speaks Statistics.
Another interesting choice is vi, or Vim, which is an editor available on pretty much any Unix platform.
Sublime is a popular choice, or Eclipse.

6.2.3 Introduction to Python
In the Python world, there are also many, many tools.
Emacs is also popular, Sublime, Vim, the same Eclipse.
Other IDE: Spyder, Atom
Invest some time to learn the features of the editor, because if you know your editor well, it's going to save you a lot of time, really a lot of time. So it's a very good investment to learn your tools very well.

6.2.4 Introduction to Git and Github
tools that make version control, that is keeping track of everything that happened to a file or a set of files, an easy part of your workflow.

6.2.5 Downloading and installing Git
Git is an open source software. It's a software which is meant to be used from the command line--that is, from a terminal. But there are several GUI clients, ie graphical user interfaces, that are available to get you started with Git if you don't like the command line or if you're not too familiar with the command line.

Once Git is set up on your machine, you can use Git by itself, but many, many programmers use additional tools, such as GitHub.

6.2.6 How to create a repository in GitHub
I logged in to GitHub with my username and my password.
And if I click on this plus icon, I have the possibility of creating a new repository.

6.2.7 GitHub Interface
- create branches - take a repository and we copy it with a different name so that we can work on fixing a bug or developing a new version. Example master branch, which is the main branch, and then there's the dev, or development, branch, where new features are added so that you can test new code without creating problems to your users.

- use the web interface to merge. Merging means that you have two different branches and you want to incorporate the changes of one branch into the other branch. When there's an explicit conflict between two files, say you edited the exact same file at the exact same place, then Git is going to ask you what you want to do.
- use the web interface to report bugs and issues. And every bug is going to have a number. And you can address the bug-- or, it's called an issue. You can address the issue by using its number.
- can also comment on an issue, and you can assign an issue to someone or set a milestone, say a deadline, to get the issue fixed.

6.2.8 Questions on Git
Multiple Choice

From a terminal, what are the order of steps you take to create a new git repository?
- ○ Initialize git project, change directory to where your project is
- ● Change directory to where your project is, initialize git project
- ○ Change directory to where your project it, push project files
- ○ Initialize git project, push project files

learn more here: https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

Which git command shows you the previous events and changes of a git repository?
- ● git log
- ○ git record
- ○ git commits
- ○ git history

learn more here: https://git-scm.com/docs/git-log

What is the meaning of a "pull" of a git repository?
- ○ To make a repository unavailable to access
- ● To get the newest version of a git repository
- ○ To merge new changes onto a git repository

○ To make a repository available to access
learn more here: https://git-scm.com/docs/git-pull

How do you start tracking a new file with a git command?
○ git track ⟨filename⟩
○ git ⟨filename⟩
◉ git add ⟨filename⟩
○ git push ⟨filename⟩
learn more here: https://git-scm.com/docs/git-add

Which of the following is **NOT** a command to get help on a git verb?
○ git help ⟨verb⟩
○ git ⟨verb⟩ --help
○ man git-⟨verb⟩
◉ man ⟨verb⟩
learn more here: https://git-scm.com/book/en/v2/Getting-Started-Getting-Help

6.2.9 Video A conversation with Erik Surface
The version control systems I've used include Git, Mercurial, and Subversion.
Git is my preferred version control system, because it provides de-centralized access to code and is mostly used by Mac and Linux programmers.

Git is a program that is usually run on the terminal that provides a version control to anyone writing code. GitHub is a web platform that allows users to access the same types of commands as the Git program, but in a web interface, and to collaborate much more easily with other users.

Git can be used to store not only code, but also files to track like documentation, or small datasets are also useful in there.

Distributed workflow with Git means that the repository, or your actual code base, is shared among many people. And you may have a copy on your system, but there's also a copy on a server and a copy that a contributor has downloaded.

And as each person works on that code and makes changes to that code, it's their responsibility on their end to merge any changes that you've made before pushing it back up to the main server. And that way, changes to the code are shared in a repeatable manner.

It's a good way to **make changes to code incrementally** so that you can continue your work, and also allows for you to go backwards in time to run different versions of the code.

So if you need other people to test your beta code, you can have them get the code from the dev branch, and then your regular users get code from the master branch. And in that way, you're not breaking everybody else's environment or system with your new feature set change.

Advice to use the tools around for reproducibility, because when you come back to this code, you want to be able to run it easily in a way you remember. And both things like Git, good comments, labeling, and making notes in a Git environment can help you do that.

6.3.1 A conversation with Merce Crosas
And a lot of times, in spite of how many researchers work with data, they underestimate, or they are not aware about data management and importance of data sharing.
When you're working in physics or astrophysics, you're basically building your own data analysis tools, your own data management, and sometimes, your own data infrastructure. Build own tools, build own software for analysis.
To build data systems that would help tracking the data in a lab, like a Lab Information Management System, or LIMS, or building some other database that would help to store and access the data easily by a group of researchers.
Data management is about and how you can make a difference by thinking about that ahead of time and taking it into account as part of your research.

6.3.2 A Conversation with Merce Crosas

Data management consideration

- what data, what type of data do I need to collect to get a data set that will help you address the research questions?
- How to do the data collection, using which instrument?
- what are the values? What are the different (column) attributes or variables?
- what is the structure of the data or metadata? (sometimes the structure of the data comes already with the instrument that you collect that data)
- most researchers that work with data will tell you that they spend about 80% of the time cleaning and processing data sets
- a tabular data set or a quantitative data set, will need to be organized
- what format you will have your data in? what is my community using? (eg. astronomical observations)
- what format-- CSV format, or an R data frame format, or some other format that will make it easier to do that analysis
- where will you store the data?
- how will I share that data set with others while we're working on it?
- what considerations I need to do for sensitive data?

And you don't need to necessarily spend a lot of time to think about those, but just by being aware of it in the beginning of planning your research helps you then work with the data so much better.

And the data-management process is by itself not a goal, but what it helps you with is that, then, after a few months, after you've done your research, you and others in your team will be able to reuse that data.

6.3.3 Documenting Your Data

Best practices for documenting your data

- how well you document your data.
- Machine readable
- tools like Markdown, or Notebooks, data format, type of variable, what transformation, what they signify, the process, the analysis, etc

- fair data principles (findable, accessible, interoperable, and reusable data)
- as standardized as possible, in the way that your discipline or research community uses and can understand.
- to map or be able to relate how that type of terminology or tools that you're using mapped to other disciplines, because that will make your data more interoperable.

6.3.4 Data Sharing
Data sharing = making data accessible
that goes with the code and the documentation associated with that data.

6.3.5 Ten Simple Rules for the Care and Feeding of Scientific Data
Published: April 24, 2014 https://doi.org/10.1371/journal.pcbi.1003542
1. Love your data and help others love it, too.
2. share your data online, with a persistent or permanent identifier (data citation).
3. conduct science with a particular level of reuse in mind. In notebook
4. Publish work flow as context, connects to also documenting your data and the workflow and the transformations (provenance) that you've got with your data
5. link your data to your publications as often as possible
6. publish your code, even the small bits, a chance again to make the data open, others can help you even improve your code
7. state how you want to get credit
8. foster and use data repositories. domain-specific repositories. There are interdisciplinary repositories. There are institutional data repositories.
9. reward colleagues who share the data properly
10. Be a booster for data science

6.3.6 Merce Crosas on the Challenges of Big Data
Harvard University Privacy Tools Project: https://privacytools.seas.harvard.edu/
DataTags: https://datatags.org/http://datatags.org/
Repositories that exist now are meant to be for a small data sets, I mean megabytes, a few gigabyte data sets, but not a terabyte or petabytes.

So what is needed to make that work is the combination of data repositories with computing computational resources and technologies.

integrate the computational workflow (working in a cloud computing environment or in a research computing environment) with repositories that comes with provenance and version control, so that data accessible to other team members.

That will help us to support big data, large data sets, streaming data sets that require intense computational resources with the repository itself.

If we want to accommodate data sharing for large data sets and also streaming data sets, or data sets that get frequently updated, they will need intensive computational resources to interpret those data sets and derive results.

What happens when you start having large data sets or streaming data sets is that, often, you will not download your data in your local environment to do the analysis. So then what you want is that the repository facilitates access to the computing of that data.

And you bring computation software close to the data so that you don't need to be doing large transfers of data that would be slow for this type of large data sets.

6.3.7 Big Data Privacy Concerns
- building **security environments** that help to share that data in a way that standardizes it and makes it easier to make parts of that data accessible. Or at least expose what you can to the public, but still follow the requirements for that part of the data set that should be restricted and follow security requirements, and access requirements.
- building **tools** that would help you provide summaries of the data, analysis of the data, while preserving the privacy of the data set.

- privacy tools research project collaboration that it's part of well different groups at Harvard and MIT.
- blue level, data is not sensitive at all to crimson, which is you need to have high security

- six levels to standardize how data is accessed in a secure way
- system standardization is very important so that you can do across different environments, different repositories, across different institutions in different countries.
- financial tool to run some summaries statistics or analysis of a data set in a way that you are able to preserve the privacy of that data set that you have access the raw data.
- other tools like the one called robot lawyers the collaboration is working on that helps you also generate data use agreements and licenses for data sets.

6.3.8 A conversation with Merce Crosas 8

there are requirements that come from journals that ask for the data when you publish an article, and also just a realization that we're publishing so much science, so much research in articles across all disciplines. All the data and the code that went into the research findings should be available so that they could be verified over time. And also they could be reused.
So we were building science on top of the science for the develop and build by others. So we can solve problems faster. We can improve the quality from previous results.

6.3.9 Document Your Data

When you're thinking about reproducibility, the bare minimum that you need is document your data, just document what you are doing, your data transformations, your data analysis. And that documentation can be as simple as a document text that just says what you've done step by step, or with tools like Markdown, or Jupyter Notebooks.

6.3.10 Merce Crosas Reasons to Share Data

It's a reusable just helps you for yourself to be reused.
If you do all the steps to make that useful just for yourself, eventually you'll make it useful for others.

It is important to establish some standards or systematic ways and protocols of how you're going to collect that data so that it is easier to share later so there are no issues related to privacy or it's not done in a way that then you feel protective of that data because you've done it.

6.3.11 A conversation with Sonia Barbosa

The project Dataverse is an open source web application to share, preserve, cite, explore, and analyze research data. It facilitates making data available to others, and it allows you to replicate others work more easily.

A dataverse repository is the software installation, which then hosts multiple dataverses. Each dataverse contains data sets and each data set contains descriptive metadata and data files, including documentation and code that accompany the data.

Dataverse the software is open to researchers, journals, institutions, and developers. Data is more discoverable to the research community and satisfies data management plans.

Developers get to contribute code extensions, documentation, testing and/or standards of integrating, research analysis, visualization, and exploration tools with dataverse.

People wanted control of their data, and they wanted to retain copyright of their data, and they wanted to know everything you were doing with their data. Now, you have a digital platform that actually allows you to do that. As the author, you maintain control. You decide what you put up, and you decide who gets to access it, and it also makes my work a lot easier because now I don't have to actually clean other people's data. They clean their own data with my guidance.

If you answer the questions using the tools that we give you, your users are going to have a great picture of your overall data collection, from beginning to the end.
Why did you collect the study?
What measures did you use?
What was the sample?
How were your variables computed?
What program was the original dataset in?
What did you convert it to?
Do I have to use a particular software, statistical software, to analyze the data?

6.3.12 Dataverse Sonia Barbosa

To look at some examples of the features that we have available for the Dataverse tool.
project website is dataverse.org
We have a large community, not only of users of our data but of developers that are interested in helping us to develop the code and become part of our Dataverse community.
Dataverse is an open-source software that allows you to create a repository for data storage.
Available to Researchers, journals, institutions, and developers.

6.3.13 Dataverse Sonia Barbosa P2

on the dataverse tool, once you publish something, you cannot unpublish it.
You have to offer retractions, and in the case of data, you offer what's called versioning.
if you have any questions, you can contact us at support@dataverse.org

6.3.14 API Access

How to interact with the Dataverse in a programmatic way through a mechanism called API, application programming interface
To get the token, I logged in to the Harvard Dataverse. And I go to the menu, API token. And it's my token, so it's something that identifies my account, and it should not be shared.
There is a package called Dataverse developed by Thomas Leeper, which allows to connect to the Dataverse.
The package is available on GitHub. I'm going to install it using DevTools.
Then load it to R. And I have to set two variables: Dataverse key (token), Dataverse server.

6.4.1 Introduction to Dynamic Report Generation

Reproducibility: use the right tools (eg. R Markdown, Jupyter Notebook) for your own papers, you use the same code, you use the same data, and you get the exact same results.
Literate programming: the possibility of combining together, in the same document, code and analysis.

Reproducible Research with R and R Studio, Second Edition Christopher Gandrud

6.4.2 A conversation with Christopher Gandrud

Reproducibility-- when I started my PhD in 2008 was kind of an amorphous concept that people had a vague sense that they should, when they submitted an article for a journal, have basically their data and some of

maybe their source code included with their submission to their publication. But it wasn't very systematic, tools are not available, and very few people actually did it.

The tools are becoming even easier to use, and the incentives are really there, especially for graduate students and young researchers, to use these tools.

The overarching principle of reproducible research is document everything, and what specifically that means and what tools you use for that is going to be dependent a little bit on your project-- so how big your data set is, what type of software you're using for your analysis. But really, the key is to document as many of the steps that you took as possible to reach your research finding.

No matter how your data is formatted, assuming it's not a gigantic data set, you want to be storing the analysis code and the data in plain text files. And this is because plain text files allow you to totally document your research in a way that can be opened by other researchers. It's not dependent on a specific piece of software, and it's not dependent on software that may not exist in the future.

6.4.3 knitr and Rmarkdown with Christopher Gandrud

R Markdown document allows you to tie your analysis, your data gathering, and your presentation documents together in one place.

A new RStudio session, and we can see four panes.
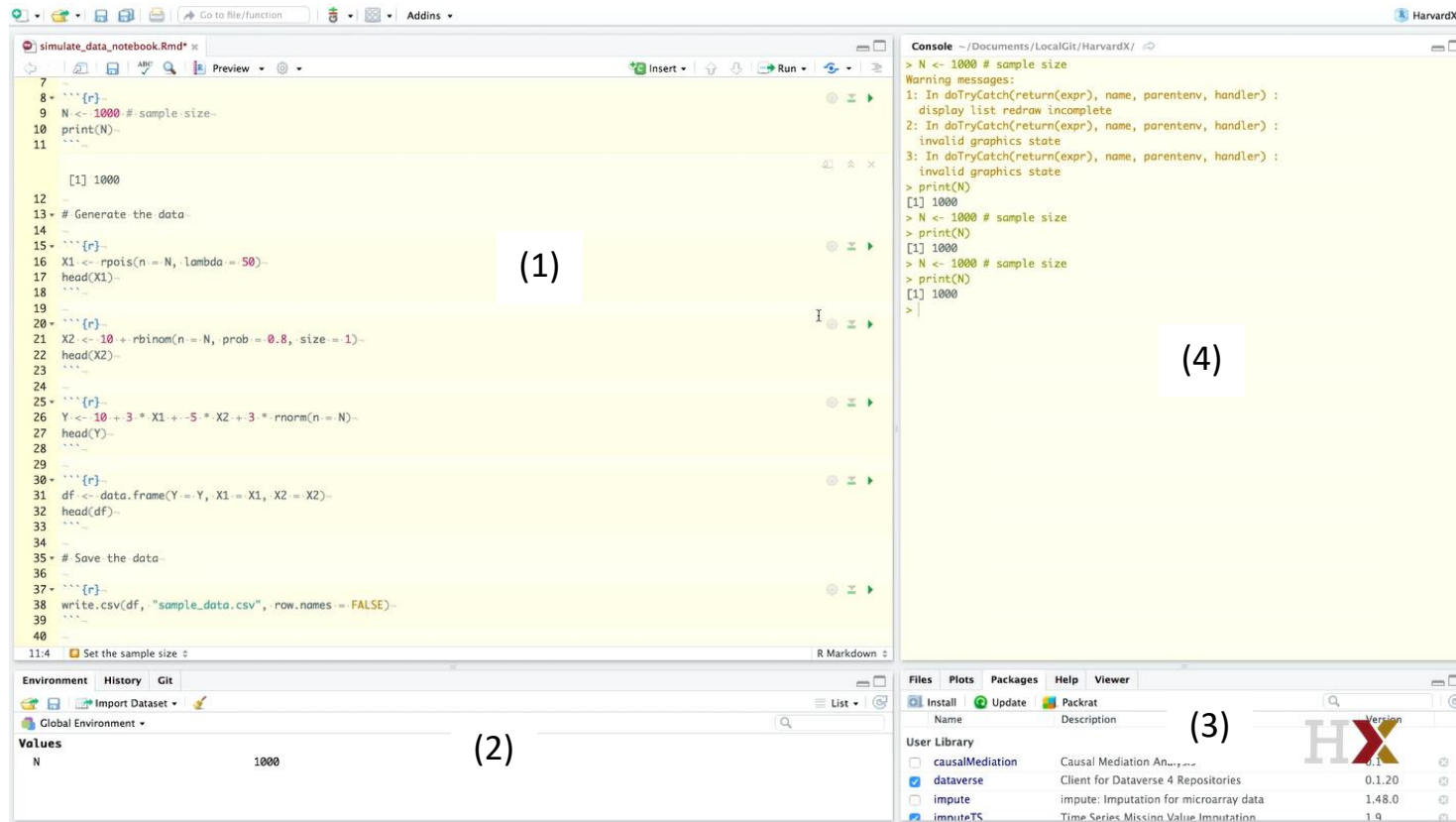
(1) The first one here on the top is our R source code file. It also be where our R Markdown document will be.
(2) The one on the bottom, the lower left, is the R console. So we can type in any R function that we'd like to do down here and run it interactively.
(3) On the lower right, we have a pane that includes a number of different options. One is file viewer and also a plot viewer and a way to see your packages and help files.
(4) And then on the top right, there's the environment and history pane, which will show you all of the objects that are in your environment and also your function history.

6.4.4 Questions on Knitr

Q1. Which of the following can you control within the chunk options? Choose all that apply.

☑ The specific lines of code that get evaluated

☑ The size of the output figures

☐ The speed at which the code gets evaluated

☑ If the code stops when it encounters an error

Explanation Within the chunk options, one can specify which lines get evaluated with the "eval" option, the size of the figures with the "fig.width" and "fig.height" option, and if the code should stop if it encounters an error with the error option.

Q2. What kind(s) of output files can a Rmarkdown document generate? Choose all that apply.

☑ .pdf

☑ .html

☑ .docx

☐ .ppt

Explanation Rmarkdown can produce pdf, word docx, opendocument odt, rich text document rtf, and markdown md documents.

Q3. What would tidy=T do to the a code chunk? Choose all that apply.

☐ It automatically truncates the error outputs if the code generates an error

☐ It does not do anything

☐ It takes the results from the chunk and formats it in a legible way

☑ It takes the code within the chunk and formats it in a legible way

Explanation tidy uses the tidy_source() function from the formatR package to format the source code in a legible matter for display in the knitted output.

Q4. With which chunk option can one adjust whether or not the output gets saved for loading for the future (after, say, one restarts R/Rstudio)?

☐ future

☑ cache

☐ save

☐ load

Explanation With the cache option, one can control if a code chunk gets saved so that it does not need to be rerun on the future occasion. By default, the objects created by the code in these chunks are loaded from saved databases stored in the same folder that the .Rmd file is stored in.

Q5. Can an .Rmd document evaluate code chunks that are not written in R?

☐ No, code chunks can run R code only

☐ Yes, it will detect the language automatically and switch dynamically

☐ Yes, but only Python code

☑ Yes, only if the language is specified in the code chunk

Explanation Code engines are specified in the code chunk with the "engine" option. Specific versions of the code engine can be specified with the option "engine.path" pointing to the directory where the programming language has been installed.

6.4.5 Notebooks R Notebook
To go through an example in which we're going to use an R notebook to simulate data, and a Python Jupyter notebook to make a graph.

6.4.6 Notebooks Compiling in an R File
The work flow was I generate the data in an R markdown file. I create a plot in a Python notebook, and then I estimate in an R file.

6.4.7 Notebooks Jupyter
To use a Jupyter Notebook to generate a graph of the simulated data using Python.

6.5.1 Makefiles
The last computational tool that we're going to introduce is Make. So the command Make and the associated Make files. So very often when you have to perform an analysis, you have to retrieve data, you have to clean it, you have to process it. Then you have to perform statistical analysis, write a document, create graphs, et cetera, et cetera. And it's possible that you use different tools, that people in your team are using different tools and you can use Make to make these tools communicate together to create a final analysis.

6.5.3 A Conversation with Simon Adar Introduction (part 1)

If I'm using somebody else's code in my computer, I have a different set of dependencies and versions which are different from the author's computing environment. So in many cases, the code will simply not work because of those differences.

To have a global system, a centralized system where everyone can deposit their code, publish it in a freeze environment so that the computational environment doesn't change.

6.5.4 A conversation with Simon Adar On Docker (part 2)

One of the most important building blocks of our system is Docker. Docker is a sandbox container that allows you a lightweight virtual machine that allows you to set up a computing environment.

You give it a recipe, the Docker file. You say what you want in that computing environment, what kind of version of operating system you want, what kind of dependencies would you like to have, and then you can generate an image, which is the computing environment. The codes can run in that Docker container environment, for all those different authors, they have different computing environments, different programming languages, different versions, different dependencies.

6.5.6 A conversation with Simon Adar part 4

Code Ocean platform

Every published algorithm on a platform has a DOI that you can cite. And you can cite the exact state of the code, including the whole computational environment. So there's a freeze. Once you publish, there's a freeze of the Docker container, all the packages that you added, all the versions, and the code itself. So everything is frozen, including the metadata. And if you have any versions later on that you want to add to that, it's going to be a version of the same entity.

Planning to have a GitHub sync mechanism, where you not just get things from GitHub, but we also can push the versions to GitHub. And then if you're doing it right, you actually can run the code locally, run it on Code Ocean, and you have the same version all the time.

6.5.7 A conversation with Simon Adar part 5

Code Ocean is living in the clouds. It's a natural place for Code Ocean to use the power of the cloud. We can scale very rapidly.

We can use lots of cores for any task.

We can use lots of memory inside a machine.

We can host big data.

We implemented as part of the architecture we have a special storage for big data that is very close to the computing servers. So you can access those big data files in a very efficient manner.

It's not replacing development platforms today. Because when you develop you need actually a debugger, and Code Ocean is not a debugging system.

6.5.8 A conversation with Simon Adar part 6

Code Ocean partnered with IEEE. Every author that submits an article to IEEE will be able to link that article with the code that generated the figures in the article or demonstrated the algorithm, the analysis, the benchmark.

But we also have a widget, which is very similar to a YouTube widget, and that will be at the point of the article that you can read the article in the publisher's website. You'll be able to see the code-- download code for free. If you want to run, you have to be a signed in user.

Code Ocean has APIs that different platforms can integrate with. We are integrating with the publishers-- we're integrating with repositories. One of them is Dataverse.

6.5.9 A conversation with Simon Adar On Jupyter Notebooks (part 7)

Jupyter Notebooks are beautiful, are great. It comprises prose and code together in the same place.

Every programming language that is being supported by Jupyter, let's say Python, let's say R, there is a subset of that community that is using Jupyter Notebooks.

Jupyter Notebooks is a place where it's one file. Sometimes, projects are complex. They have hundreds of files, hundreds of Python files behind the scenes. And you build in the model form, so if you're still going outside of the Jupyter Notebook and opening the old style of an editor to change the code under the hood.

Jupyter Notebook itself doesn't provide you any tools to preserve the programming versioning, the packages, the operating system. And I think for them to move into that, they would have to maybe use a combination of Jupyter Notebook and a Docker.

If somebody wants to use Docker, and he's still left with a problem that his users will have to use Docker and will have to learn how to use it, and there's a certain level of complexity that I think most researchers would

not like to use it in its current way. So in Code Ocean, we build a platform that hides the complexity of Docker, and make it more accessible for the common researcher.

6.5.10 A conversation with Simon Adar part 8
Code Ocean platform is open for everyone to sign in and use it for free.
If they're publishing on a platform, they get the pro account for free for six months.

6.5.11 A conversation with Simon Adar part 9
Everything that you have in GitHub that is free is free on Code Ocean.
And you can do more than that for free, which is run algorithms, for free, every month.

6.5.12 A conversation with Simon Adar part 10 on reproducibility
It starts with try to think from a user point of view, not yourself. If you were given somebody else's code, what would you need to do to make it work?
Start with the file that you actually want to run first, and give it a meaningful name, like main or run or something that points out and screens out, this is the file that you look at first.
And the second thing is to provide some kind of data with the code, because people don't know, if you get a code, and they don't know what to do with it, and how to connect that code with data that they might have.
And it's so useful to see a sample code, synthetic data, or anonymize data.
The last thing is to provide described list of dependencies that this thing needs to live in a computing environment, because it's really frustrating to run code, and then you get errors, you don't know where it's coming from. And you see a missing library or package, and then you try to install that. But then you are missing something else.
In CodeOcean, we're integrating everything in the same platform. So the computing environment has everything that it needs. The user does not need to know about what is missing, because everything works. it has the code, the data, and a working environment with everything that it has.

6.5.13 Video Demonstration of Code Ocean with Simon Adar
To start from the point of view of a user that is going to see an article published in a scientific journal.
Click that link, it will take you to the Code Ocean website and you'll be able to see the code and the data that is associated with that algorithm.
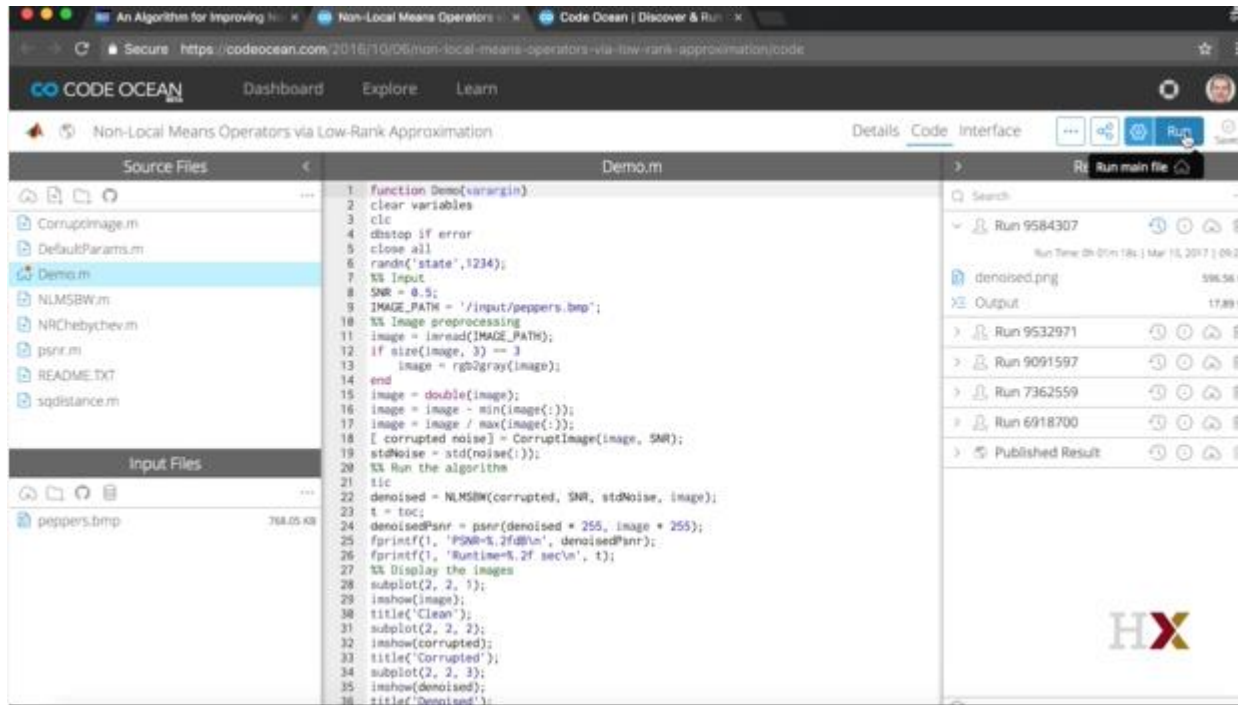
If we press that Run button, what happens behind the scenes is that we take the code. We associate it with a write computing infrastructure. We run everything in the cloud. And then when the results are finished, we are bringing it back here. And the user is able to visualize the results.

Every algorithm has, in addition to the Code tab, has a Details tab. Here, you have a description, some information about the algorithm, like a DOI Digital Object Identifier.

6.5.14 Simon Code Ocean Conclusion

If you need some help, you can always use the Learn tab and find out a little bit more with a very quick, short animated GIFs that shows you how to do different tasks on Code Ocean.

6.5.15 Creating a New Algorithm in Code Ocean

Creating a new algorithm in Code Ocean is very easy. You go to the dashboard, press New Algorithm, choose the programming language you want. And you immediately will get a template which is a working template. If you press Run, you run the code in the cloud, and you'll be able to see the results.

6.6 Conclusion

some tools have been around for a long time, for example, Make or Vi or Emacs. Some tools are more recent, for example, R Markdown or Jupyter. Some tools might be successful. Some tools might change. Make sure that there is enough community to support these tools.

In general, try to document as much as possible. So document your code. Document your data. Document your documents, which sounds a bit redundant. But document everything you can. And try to stay up-to-date when it comes to best practices in reproducible research.

Final Project

Course Final Project

This project serves as an opportunity to apply the techniques presented in the course videos to attempt to reproduce the findings of a recently published journal article:

[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.

In 1995, MIDUS survey data were collected from a total of 7,108 participants. The baseline sample was comprised of individuals from four subsamples: (1) a national RDD (random digit dialing) sample (n=3,487); (2) oversamples from five metropolitan areas in the U.S. (n=757); (3) siblings of individuals from the RDD sample (n=950); and (4) a national RDD sample of twin pairs (n=1,914). All eligible participants were non-institutionalized, English-speaking adults in the contiguous United States, aged 25 to 74. All respondents were invited to participate in a phone interview of approximately 30 minutes in length and complete 2 self-administered questionnaires (SAQs), each of approximately 45 pages in length. In addition, the twin subsample was administered a short screener to assess zygosity and other twin-specific information. With funding provided by the National Institute on Aging, a longitudinal follow-up of MIDUS I began in 2004. Every attempt was made to contact all original respondents and invite them to participate in a second wave of data collection. Of the 7,108 participants in MIDUS I, 4,963 were successfully contacted to participate in another phone interview of about 30 minutes in length. MIDUS II also included two self-administered questionnaires (SAQs), each of about 55 pages in length, which were mailed to participants. The overall response rate for the SAQs was 81%. Over 1,000 journal articles have been written using MIDUS I and II data since 1995.

You will attempt to reproduce the findings of [1] and critique the reproducibility of the article. This particular article focuses only on MIDUS II data, including biomarker data, and investigates the relationship between optimism and lipids. You can download the

MIDUS II data and supporting codebook and other documents [here](). You can download the data in multiple formats. You can download the biomarker data [here]().

The project must be submitted in the form of a Jupyter notebook or RMarkdown file, and include an introduction, methods, results, and discussion/critique section with commented code interspersed. Visuals and schematics should be included if appropriate. All project documents must also be uploaded to the course [GitHub repository]() in a folder. The folder must also include a README file describing the contents of the folder and how to reproduce all results. You should keep in mind the file and folder structure covered in the videos and make the process as automated as possible.

Specific tasks:

- Reproduce all figures and tables

- Reproduce all analyses

- Critique the reproducibility of the paper

    1. Is the data publicly available?

    2. Is the data easy/intuitive to access?

    3. Is there a codebook and/or instructions about how the data and documentation is organized?

    4. Are the file names intuitive?

    5. Are the variable names intuitive?

    6. Is the software used for analysis publicly available?

    7. If the software is available, is it well commented?

    8. Is there a toy example provided?

    9. Are you able to reproduce the figures, tables and results presented in the paper?

    10. Was there anything you think should have been made clearer, or explained in a different way?

    11. Did you find any faults in the methods used in this paper? Would you have used more or different methods?

Chapter 7. Conclusion to the Course

The goal is this course is to help ensure that reproducibility is integrated into data science processes from beginning to end. The combination of concepts taught can hopefully be at the back of your head during all stages of data science, from planning through publications and data dissemination. With support from software tools, your own best practices can be supported by computational aids to make the process seamless.

Course completed!

Recap topics covered:
- Introduction – concepts and motivations
- Fundamentals – terms and best practices
- Case studies – historical incidents
- Data provenance – methods for data tracking
- Statistical methods – mathematical theory behind best practices
- Computational tools – concepts implemented through software environments

Happy learning!

GitHub Classroom

You're ready to go!

You accepted the assignment, **HarvardX Final Project**.

Your assignment repository has been created:

https://github.com/HarvardX-PH527X/harvardx-final-project-JNYH

We've configured the repository associated with this assignment (update).

Your assignment was due on **May 3, 2018, 06:40 SST**

Principles, Statistical and Computational Tools for Reproducible Data Science

HarvardX - PH527x
Started - Dec 31, 2020

Resume Course

Course   Progress   Dates   Syllabus   Schedule   Discussion   Glossary   Repository Resources   More... ▾

View my courses

Congratulations!

You have completed your course. Share your success on social media or email.

in 🐦 f ✉