Computational Fluency Workshop

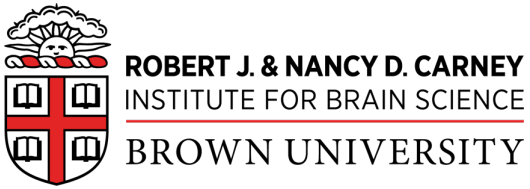
Day 1: Beginning with structure

https://github.com/brownritt/cfw2022

Jason Ritt

jason_ritt@brown.edu

Scientific Director of Quantitative Neuroscience



Expectations

> "Everybody is ignorant, only on different subjects."
>
> - Will Rogers

> "What makes coding uniquely difficult? Coding is hard on your memory. This book's theme is that writing good research code is about freeing your memory."
>
> - Patrick Mineault, *Good Research Code*

This workshop will demonstrate tools, but the true goal is to consider *process*.

You will already know some things, but probably not all the things. Don't be afraid to be wrong, as long as you are open to change. Ask for help when you want it. Help others when you can (*if* they want you to!).

Everyone uses StackExchange and Wikipedia; not everyone uses them well.

Expectations Part Deux

The reality is you will need to learn and do things your PIs and mentors do not, because the practice of science is changing faster than the people doing it.

I will present one way of doing things, but if you have reason to prefer an alternative process, great. Just be thoughtful about your choices, and develop a process that works for you (and your colleagues…).

I will focus on scientific computing and data science (distinct from, e.g. app development), but most principles will easily generalize.

In the course of four sessions, we cannot cover any one idea or tool comprehensively. The goal is to bolster your lifelong learning, refined by sustained practice, of the use of computation in scientific research.

Software installation: It takes time, so we start it first

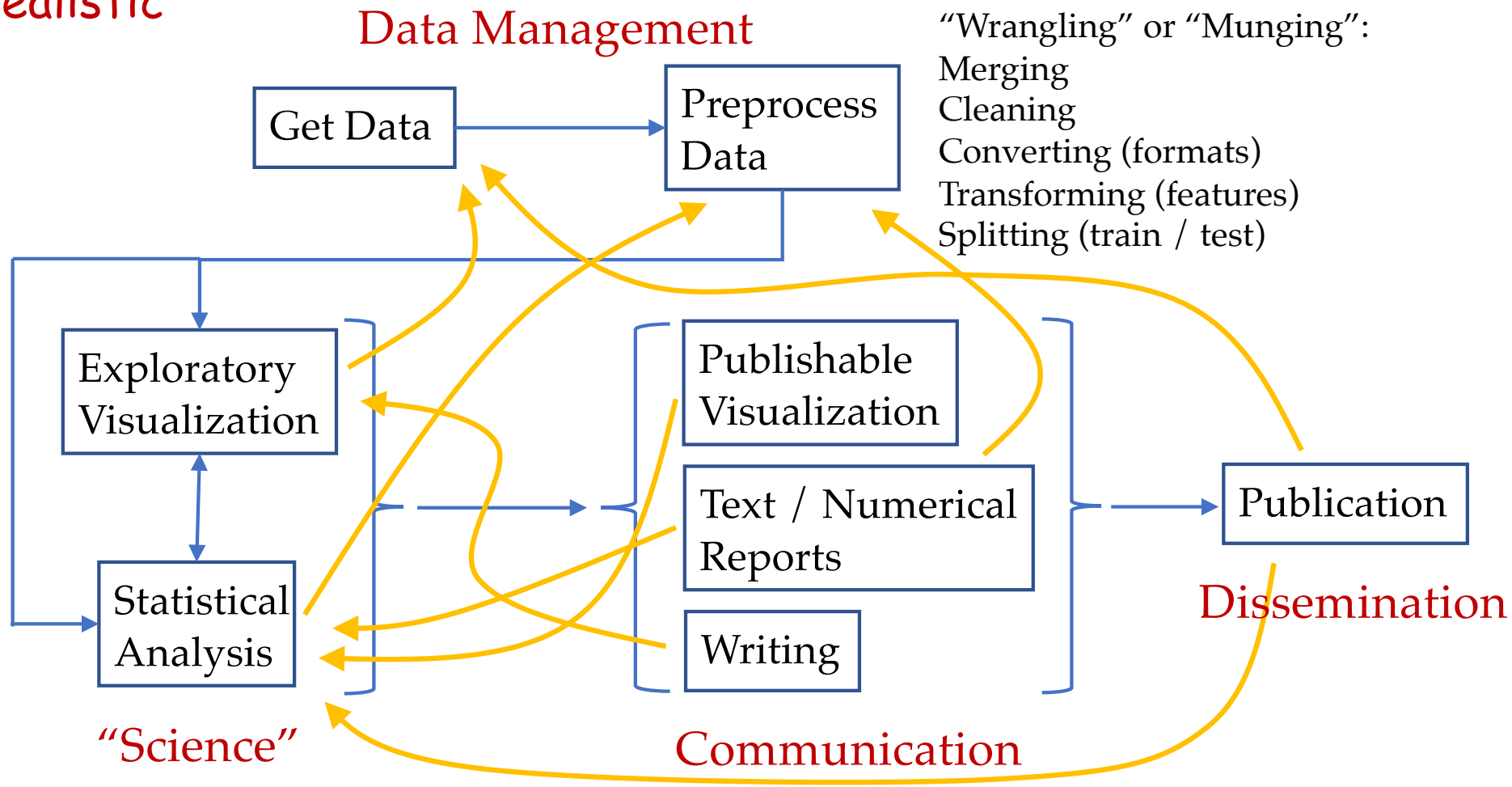
Because tools can take some time to install, we want to start that process now, so our computers can churn away in the background while we talk about what they are and how to use them.

Note: Adopting a good system often takes a lot of upfront effort, with a delayed pay off in terms of ease and rigor. It's completely normal for things to feel uphill for a while at the beginning.

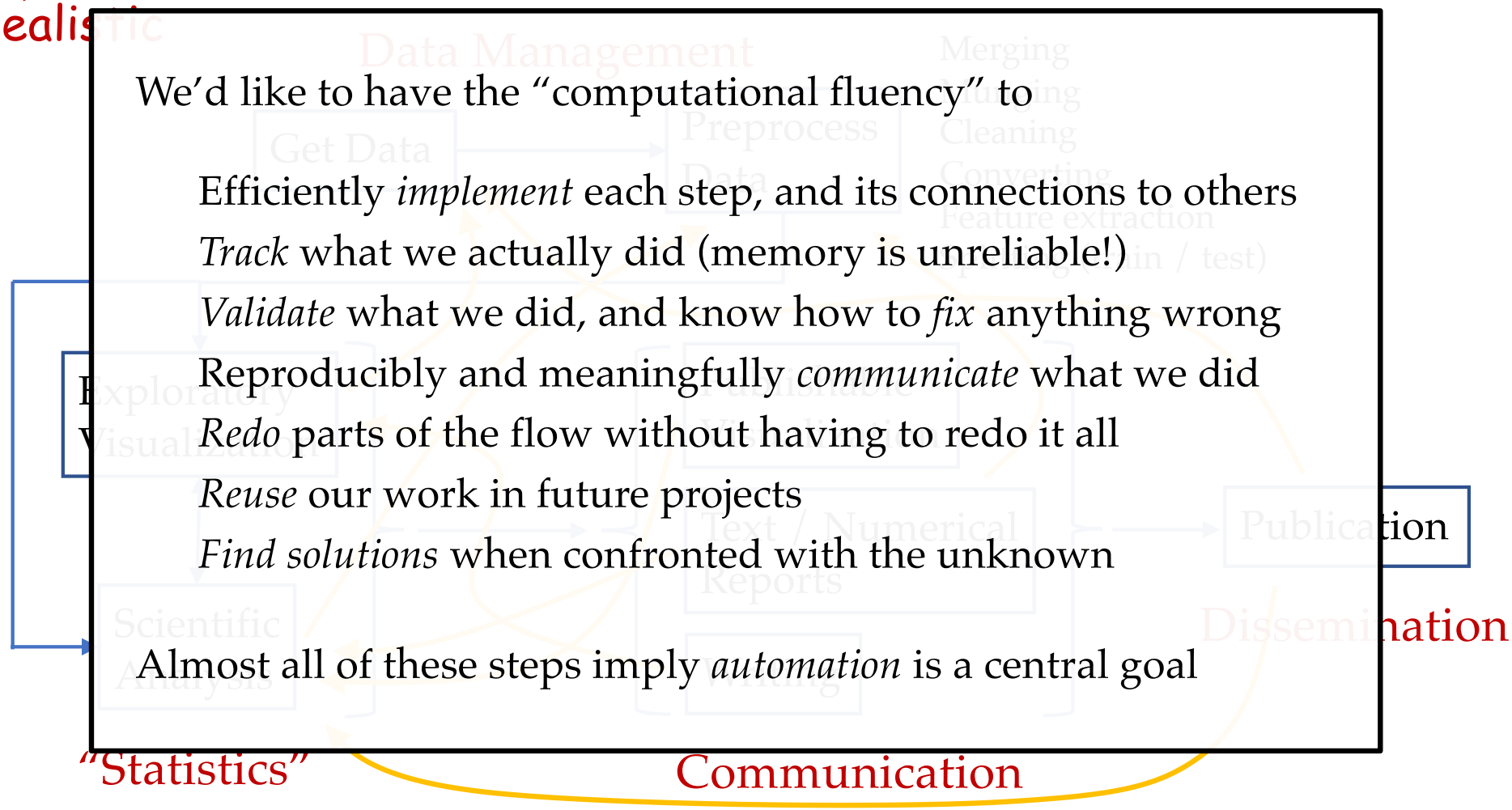
Today's targets:

- coding environment (Matlab or python)
- an IDE (Matlab or Spyder)
- `git`

An ~~ideal~~ realistic scientific analysis workflow (1000 ft view)



An ~~ideal~~ realistic scientific analysis workflow (1000 ft view)



Challenges that distinguish *research* computation

Vague specifications: It's often not clear exactly what the problem is, or what would count as a solution.

Iterative implementation: It's rare that a research computation project is well defined by end users (scientists) and delivered on a specific launch date. There will typically be many versions and a lot of back and forth while the science itself develops.

Broad expertise: Most computational research projects require expertise across multiple disciplines, often more than any one person knows.

Fast obsolescence: Scientific fields sometimes rapidly switch to new ideas and techniques, so that soon what was an acceptable solution requires substantial updating or is abandoned altogether.

Back to basics: What is a computer?

A central processor (**CPU**), and often auxillary processors (e.g. graphics processing units, **GPU**s)

Memory, for fast and transient work

Storage (hard drives, thumb drives, etc), organized as a ***File system***, for slow and permanent work

Devices (e.g. keyboard, screen, WiFi interface…) for interfacing with the world

Processes, many task-specific programs, interacting with each other

Every command invokes some set of the following questions:

Who am I? ***Accounts***
What am I allowed to do? ***Priviledges***
Where am I? ***Working directory***
Where is the file that I want to run or access? ***Paths***
What kind of thing is the file I want to run or access? ***File formats***

Live demos and exercises

Processes

Shells, command line interfaces, the REPL framework

IDEs over editors

Exercise: pupillometry analysis, setting up a new project, beginnings of versioning