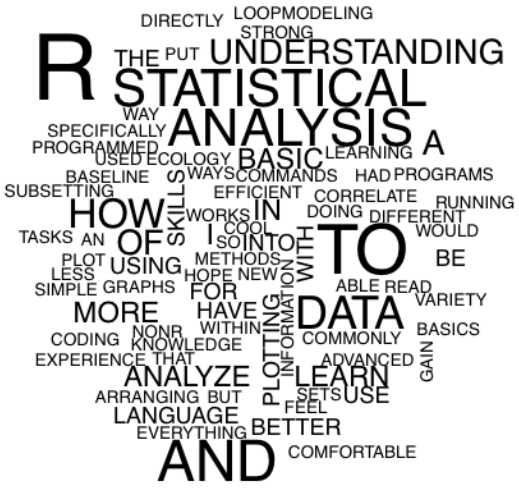


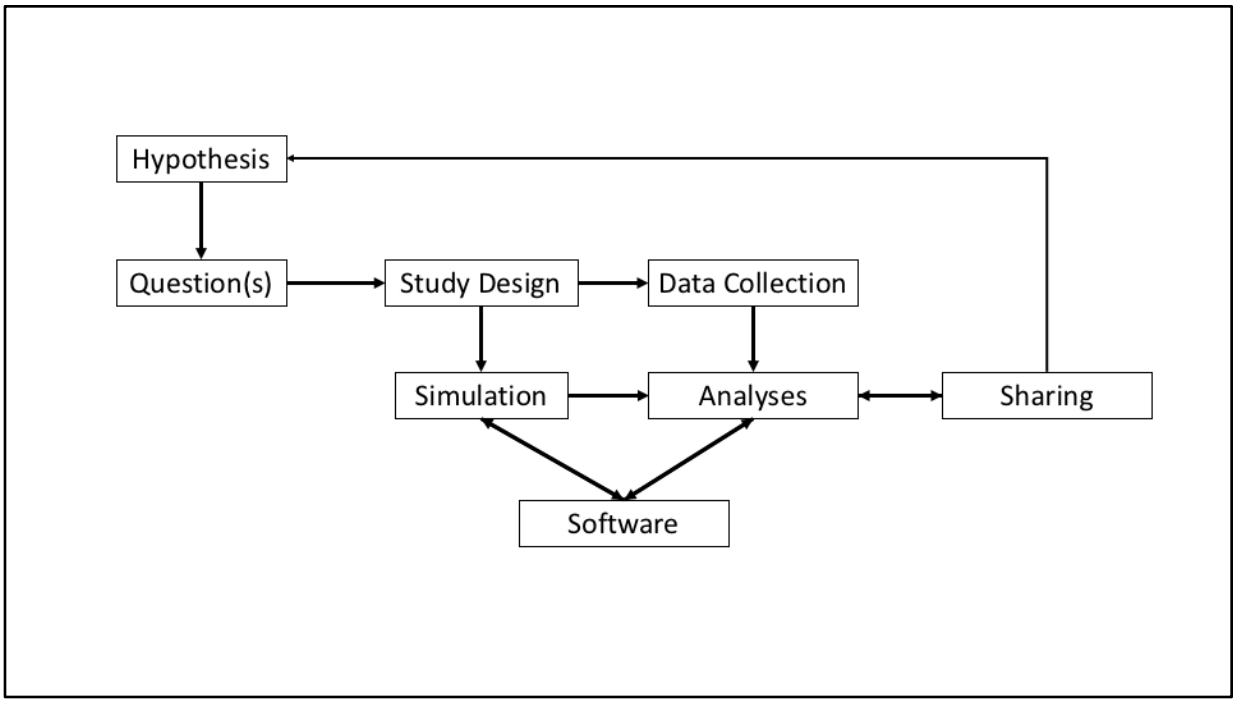
- Introduce youreself = Matt, external fellow, former REU, Originally from LA, PhD NAU, main research is community genetics of ecological networks
- Close computers, no cell phones
- Wordcloud = we're all here to learn something (BASICS, PLOTTING, MORE COMFORTABLE)
- STATISTICAL ANALYSIS = Not teaching stats, but R facilitates learning both through the language itself and through the open-source community

- Focus = R is for Reproducibility

Your moment of zen. Reproducibility is a goal not a state.

- Science is driven by the exchange of information and knowledge.
- Currently, there is a lot that we can do
- A recent study (Stodden *et al.* 2018) demonstrated that only 26% of studies published in *Science* could be reproduced. This was even more striking given that the study was conducted after the *Science* had instituted it's open data policy.
- Luckily, advances in open-source computer languages, such as **R** and **python**, provide a way to produce computations that can more easily document scientific research in a transparent, easily shared way.
- In this course, we will cover how to conduct **reproducible** scien- tific research using the **R** programming language and supporting software that will enable researchers to more clearly and easily docu- ment projects.
- Goal = coding in **R** using the *RStudio* IDE and the *git* version control system.

P(reproduce) = docs * data * software-system information


Hypothesis
Question(s)
Study Design
Data Collection
Simulation
Analyses
Sharing
Software

A Tool for Working Smarter (Transparency): code is a part of science and should be shared

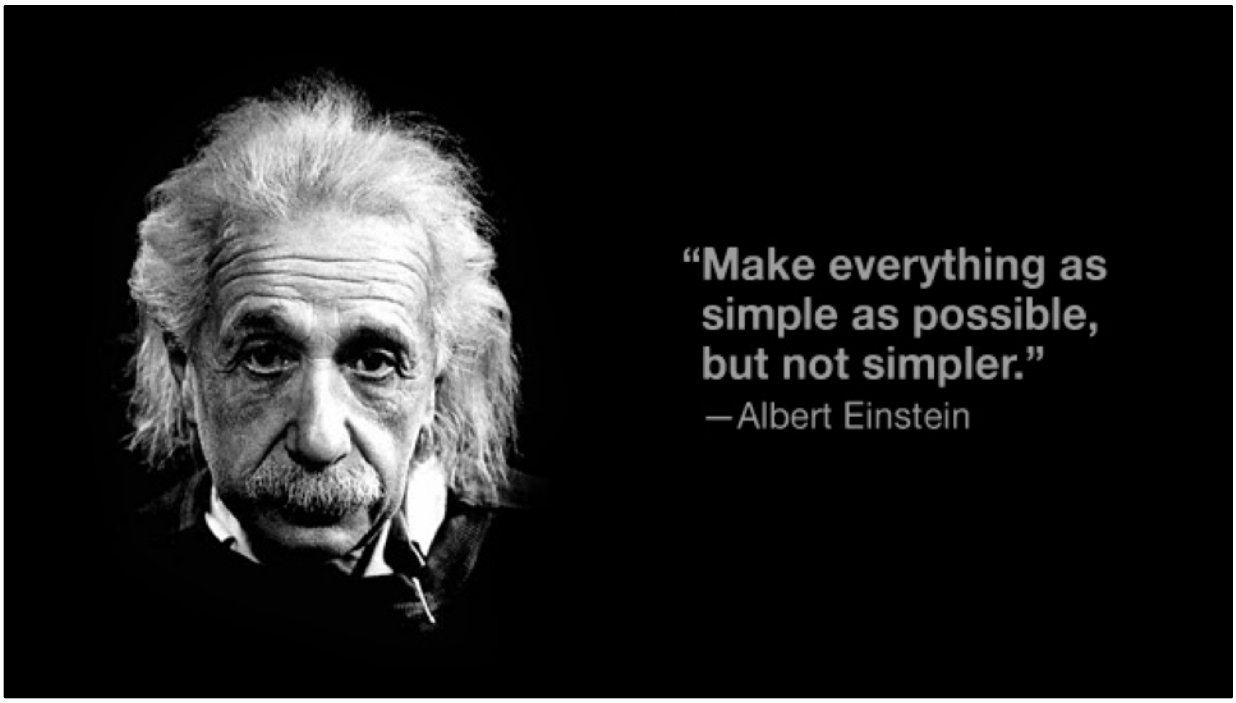
1. Setup your project so that there is a clear architecture (**RStudio**)
2. Work so that your computation from initial data to finished results will be coded (wherever possible), including data cleaning and processing steps (**R**)
3. Keep track of versions of your code (**git**)
4. Make initial data available (whenever possible, **Github**)
5. Keep track of software dependencies (***packrat***)
6. Be organized, succinct in style, coding and documentation (***formatR***, ***Rmarkdown***)

- Philosophy: Reproducibility & Benefaction
  - *Science* 44% w/ artefacts, 24% reproducible
  - FAIR
- Experiment – Design and Observation
- Data – Digitizing and Storage
- Data – Wrangling (IO, assign, object, index, type, paths)
- Check & Test (conditionals)
- Computation – Analysis, Modeling, etc.
- Results
  - Plotting
- Best Practices
  - Notebooks (data + code + annotation)
  - Style (formatR)
  - Data and code = Open Access
    - Github, Dataverse, Figshare, Dryad, EDI, NCBI
- Preservation
  - Capsules (ReproZip)
  - VMs (system independence)
  - Provenance (provR)

    - Code cleaning (cleanR)
    - Continuous integration (Travis)
- Engagement (Websites, Blogs, Twitter, etc.)
- Life-long learning – ROpenSci, stackoverflow, Tidyverse

- Spreadsheets and databases for data entry, not analysis. If “Data wrangling” and analyses are coded, they are more transparent and easier to redo.

- Scientific ”Scripts” are goal oriented, usually “one-offs”, leading to complicated “brittle machines”
- Aiming for shareable code is not the main goal, but it is important
- We want to turn on the light now, but making it easier for others to turn on the light is more beneficial


“Make everything as simple as possible, but not simpler.”
—Albert Einstein

Style

R is like an iceburg. This class will go as deep as we can.