DOING REPRODUCIBLE SCIENCE

AN OPINIONATED INTRODUCTION

OPEN SCIENCE STUDENT SUPPORT GROUP

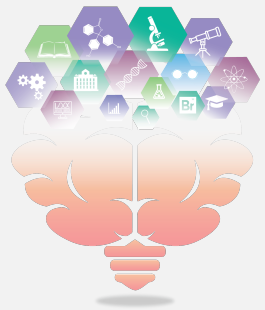
JANUARY 29, 2021

MICHAEL MCCARTHY

MICHAEL MCCARTHY

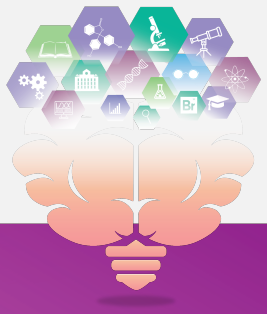
Twitter: @mccarthymg
GitHub: mccarthy-m-g
Personal Website

- Brain and Cognitive Science student working in Andrea Protzner's Brain Dynamics Lab
- In the spirit of Thomas Kuhn, I am a scientific revolutionary who wants to make open science normal science
- Developing workflows, practices, and tools to do reproducible science is one way I hope to make normal science more open
- That's my dog Thor, he's passionate about open science too



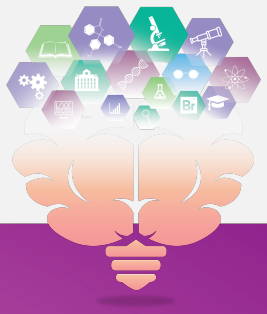
REPRODUCIBLE SCIENCE

- A scientific pipeline whose steps, processes, procedures, and results can be reproduced by other scientists (or future you)
- A separate concept from replicable science
  - The robustness of a given scientific finding as determined by the degree to which it can be repeatedly obtained
- Reproducible science makes it easier for other scientists and yourself to:
  - Verify the veracity of your findings
  - Replicate your research
    - Your findings are more likely to replicate when they are informed by Open Theory



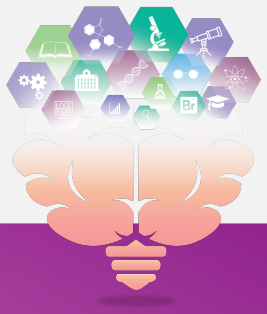
IMPORTANCE

- Scientists are untrustworthy. Some are:
  - Careerists interested in fame, money, or cultural capital over good science (e.g., Sigmund Freud)
  - Frauds running citation rings, forging data, p-hacking, or self-plagiarizing (e.g., Daryl Bem, Hans Eysenck, Mark Griffiths, etc.)
- Many are:
  - Humans making basic errors or using heuristics to guide their decision-making (e.g., You and Me)
    - More than 50% of papers report impossible statistics



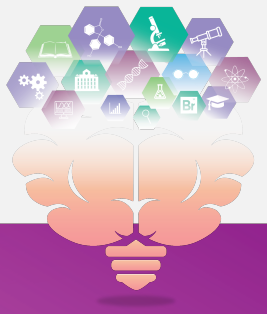
IMPORTANCE CONT'

- Thus, scientific findings should be treated as *possible but untrustworthy anecdotes* unless they can be verified by other scientists
- By making science reproducible we allow our results to be verified, increasing their trustworthiness
  - Trustworthy != True
  - False positives, undetectable data forgery, etc., are still possible with open data and materials
- Reproducibility also makes science more efficient by reducing redundant labour



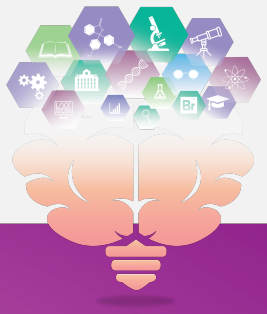
HOW DOES IT WORK?

- Science can be made reproducible by:
  - Sharing materials, data, etc.
  - Documenting your scientific pipeline
- Use free open-source software wherever possible in your scientific pipeline. This ensures your work is accessible to:
  - Lower-income scientists
  - Yourself once you lose access to all the licenses the university is paying for you
- Cite all software and packages you use ☺



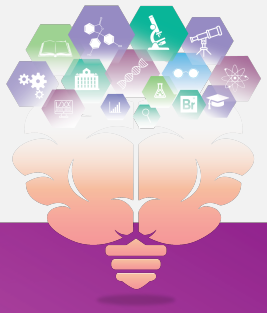
HOW DOES IT WORK? CONT'

- Catalog the scientific pipeline used to obtain your results. Methods of reproducibility include:
  - Written descriptions
  - Photographic and Video guides
  - Software along with instructions on how results were obtained (using text, pictures, videos)
  - Reproducible code
  - Packaging code and data
  - Continuous Integration, Continuous Deployment, Unit Testing
  - Machine-readable Hypothesis Testing



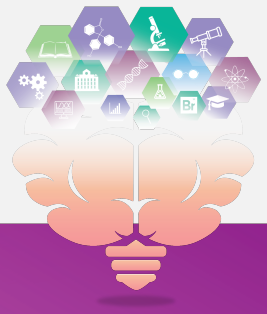
TIPS

- **Guiding principle: Keep it simple for yourself and other scientists**
- Use OS independent software that works on Mac, Windows, and Linux
- Name project folders and files in their order of operation (e.g., 01_data, 02_cleaning, etc.)
    - All code/syntax should run linearly
- Make your code/syntax human readable, follow a style guide:
    - Google Style Guides (for Python, R, and many other languages)
    - Tidyverse Style Guide (for R/tidyverse)
    - Comment your code/syntax liberally, explain what it's doing *and why*



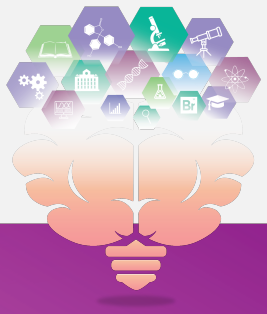
TIPS CONT'

- Make your data, code, and instructions machine readable (i.e., processable by computers):
  - Never take screenshots of data or code in place of sharing in a machine-readable format, seriously
  - .csv is the gold-standard for data, .json has uses too
  - .txt or .md are ideal for plain-text
  - Native file formats for any programming language are best for code or reproducible manuscripts
- Do not write scripts that install packages or change settings on someone else's computer, it's rude and disruptive



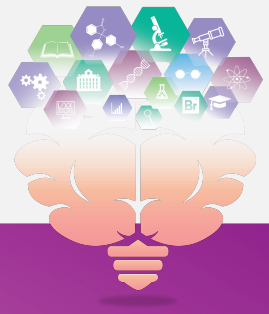
OTHER TIPS

- Do not hard code information that may change in the future or whose computation should be checked for reproducibility. This includes:
  - Citation styles
  - Statistics and other numbers
  - Tables and Figures
- Write your work as a reproducible manuscript to avoid hard coding information
- Make templates for yourself to make project management easier in the future
- Use GitHub Releases and/or OSF Registrations to take snapshots of your project at different stages



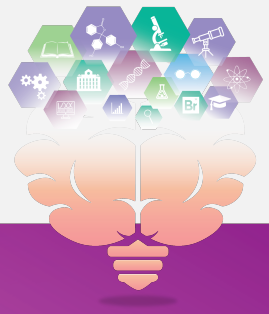
OTHER TIPS CONT'

- Set a seed before running any code/syntax that relies on a randomization function
- Use Internet Archive URLs or save webpage data if you are web mining



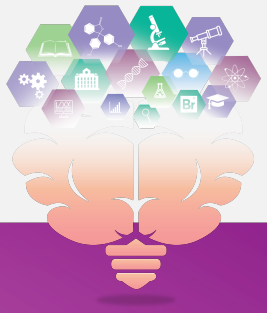
BENEFITS

- Your work will be more trustworthy
  - There's proof you actually did what you said you did
- The chances of errors in your work being identified will increase
  - (especially if you have a nemesis who wants to disprove your ideas)
- Other researchers (and future you) can repurpose your scientific pipeline for their own projects
- Collaboration will be easier
- You will learn and apply skills that will help land you a well-paying job



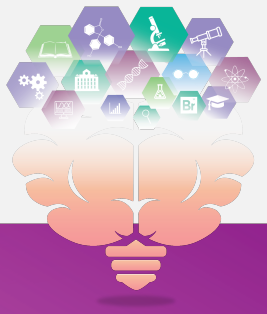
BENEFITS CONT'

- You can automate the least creative tasks of the scientific process, leaving you more time for theorizing
  - Citations can be automagically generated to different formats (APA, MLA, etc.) using CSL files
  - Statistics, tables, and plots can be automagically generated to reflect changes in your data
  - You can create living scientific documents that are automagically published to the web



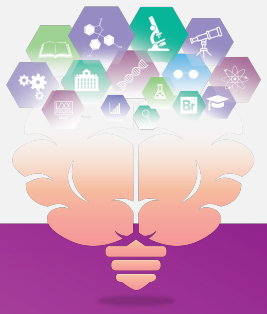
BARRIERS

- Reproducibility requires data sharing, and not all data can be shared
  - Solution: Share synthetic data that has similar statistical properties to your closed data
- Making your science reproducible may require learning new software or APIs
  - This can be difficult working around a busy schedule, but the payoff is worth it
  - Collaborators might not be willing to switch to or learn these either
  - Solution: Thoroughly documenting your scientific pipeline in a software agnostic way is a good practice regardless, so do this in the meantime



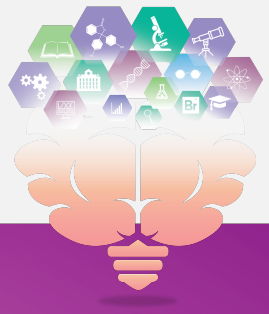
BARRIERS CONT'

- Less robust reproducibility methods may lead to irreproducible results in the future
  - Certain methods in software can break or disappear after updates
  - Solution: Use virtual environments, package version control, etc., in your projects
- More robust reproducibility methods may be less accessible to scientists with less technical ability than you
  - Solution: Make it so things "just work" without requiring the user to troubleshoot APIs they are unfamiliar with



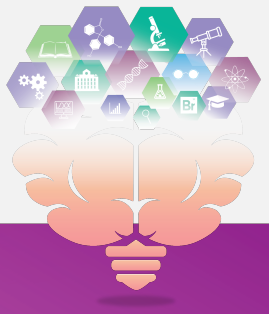
DEMONSTRATION

- If you are viewing these slides after the fact, please see the recorded presentation for the demonstration
- Ephemeral demo link: https://osssg-demo.netlify.app



REPRODUCIBILITY CHECKLIST

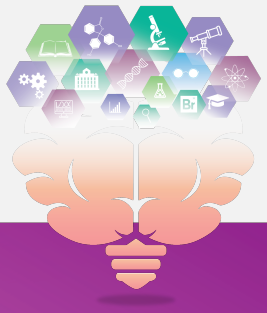
- **Are your results based on a quantitative analysis?**
  - If yes, please work through this checklist
- **Does your analysis use code?**
  - If no, does the software you're using output code? (Most GUI statistics software does)
  - Do you provide code and other documentation sufficient to reproduce all your results?
  - Do you reference the version of all hardware, software, and code used for analysis in your manuscript?
  - Is your code and other documentation version controlled? (Git)
  - Is your code and other documentation deposited in a standard code hosting repository? (GitHub, OSF)
  - Is your code and other documentation in a human and machine-readable format? (written as plain text)
  - Do you use package version control for each of the programming languages in your project?
  - Do you provide a self-contained code execution environment? (Binder, Docker, etc.)



This checklist is adapted from the TOP guidelines for reproducibility and eLife's reviewer checklist

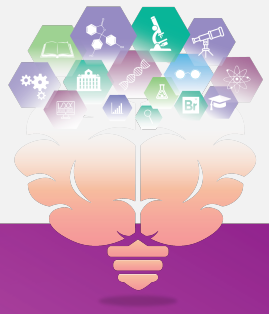
REPRODUCIBILITY CHECKLIST

- **Do your results rely on a series of pre-processing steps?**
  - Do you provide code for preprocessing?
    - If no, do you clearly describe all steps, provide a flow-chart of the steps, or refer to a pre-existing publicly available workflow?
- **Do your results rely on a series of analysis steps?**
  - Do you provide code for each of these steps?
    - If no, do you clearly describe all steps, provide a flow-chart of the steps, or refer to a pre-existing publicly available workflow?
- **Do you provide code for automated execution of each step in your scientific pipeline?**
  - If no, do you provide detailed descriptions for manual execution of your scientific pipeline?
- **Has your code been peer-reviewed for reproducibility and accuracy?**
  - If no, are you using Continuous Integration and/or Unit Testing to check for reproducibility?



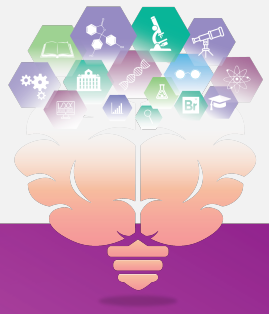
REPRODUCIBILITY CHECKLIST

- **Do your results include analysis of data collected by yourself or your collaborators?**
  - Is your data available at a digital repository?
  - Do the authors include all meta-data to reproduce the analysis?
  - Do you provide a full account of the procedures used to collect, preprocess, clean, or generate the data?
  - Do you provide research materials and description of procedures necessary to conduct an independent replication of the research?
- **Are there any results based on data or materials that cannot be shared for legal or ethical reasons?**
  - Do you provide synthetic data with similar statistical properties instead?
  - Do you explain the restrictions on the dataset or materials and how they preclude public access?
  - Do you provide a public description of the steps others should follow to request access to the data or materials?
- **Do you provide access to all data and materials for which the constraints do not apply?**



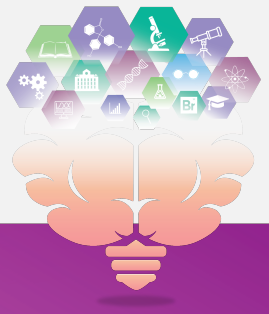
REPRODUCIBILITY CHECKLIST

- **Is your manuscript written in a reproducible format? (R Markdown, Jupyter Notebook)**
  - Do you report statistics and other numbers using inline code?
  - Do you generate, report, and embed tables and figures using code?
  - Is all your project code called within your manuscript?
    - If yes, consider splitting project code into separate script files, then calling those within the manuscript for readability
  - Is your manuscript output suitable for the forum you plan to share it in? (Preprint, Journal, Talk, Website)
    - If no, do you document the additional steps needed to finish formatting your manuscript?
- **Have you gotten anyone to try to reproduce your results?**
  - Were they successful?



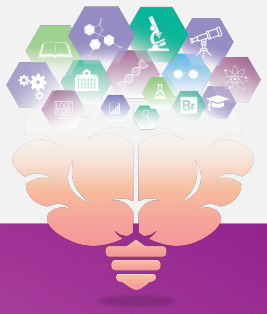
CHALLENGES

- Learn more about it!
  - Work through the The Turing Way, an open source community-driven guide to reproducible, ethical, inclusive and collaborative data science
  - Listen to one of the reproducible science podcasts linked to at the end of this presentation
- Talk about it!
  - Talk to your collaborators about how you can introduce reproducible workflows into your own projects
- Try it out!
  - Attend our Writing Reproducible Manuscripts workshop in two weeks
  - Try to reproduce the results of the first analysis you ever did
  - See how well your current project fairs against our Reproducibility Checklist
- Implement it!
  - Write your thesis project as a reproducible manuscript
  - Set aside time to check out the coding and reproducibility resources linked to at the end of this presentation
  - Pick one item on the reproducibility checklist and implement it in your next project



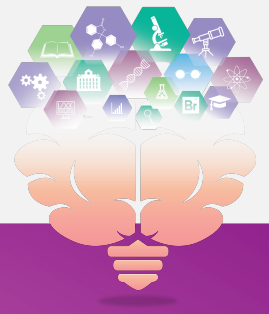
THANK YOU!

COMMENTS, QUESTIONS?



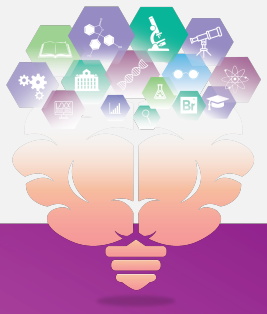
OPEN-SOURCE ALTERNATIVES

- Mendeley/Endnote alternative:
  - Zotero plus Zotero Connector
  - Import from Mendeley or Endnote
- Useful Zotero plugins:
  - scite
  - pubpeer
  - Better BibTeX
  - zotfile
  - Sci-hub Downloader
- SPSS alternatives with GUI interface:
  - Jamovi
  - JASP
- Code-based SPSS alternatives:
  - R and RStudio
  - Python and RStudio v1.4+
  - Julia
- E-Prime/Presentation/Qualtrics/etc. alternatives:
  - PsychoPy
  - jsPsych
  - Formr



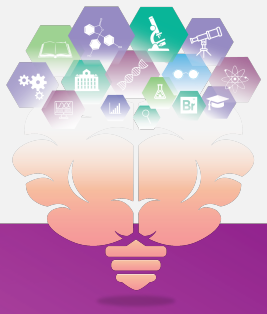
REPRODUCIBILITY IN GENERAL

- Version control:
  - Git
- Data and code distribution, collaboration:
  - GitHub and GitHub Desktop
  - OSF and osfr
- Data repositories:
  - UCalgary Library Guide
  - Nature Recommended Data Repositories
- Virtual environments:
  - Docker
  - Code Ocean
- Continuous Integration:
  - GitHub Actions
- Web hosting:
  - Netlify



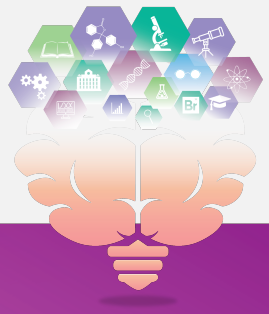
REPRODUCIBILITY IN R

- Use RStudio Projects
- Use inline R code to report statistics
- Package version control:
  - {renv} (works at project level; use from the start)
  - {groundhog} (works at script level; use to recover a reproducible environment in scripts without one)
  - {holepunch} (easy Docker sessions)
- Codebooks:
  - {codebook}
- Reproducible workflows:
  - {worcs}
  - {targets}
  - {breakerofchains}
  - Rocker
- Reproducibility w/ interactive data:
  - {shinymeta}
- Learn R Markdown:
  - R Markdown Cookbook
  - R Markdown: The Definitive Guide
- Manuscript writing:
  - {rmarkdown}
  - {bookdown}
  - {distill}
  - {rticles}
  - {papaja}
  - {officeverse}
- Machine-readable hypothesis testing:
  - {scienceverse}
- Unit testing:
  - {testthat}
- GitHub Actions templates:
  - r-lib/actions
- Zotero connector:
  - rbbt
  - citr



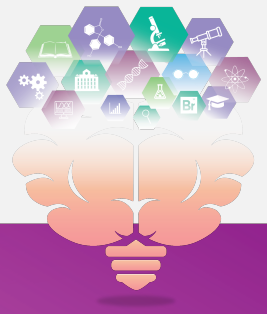
REPRODUCIBILITY IN PYTHON

- Use Python Projects
- Use inline python code to report statistics
- The Turing Way has more python reproducibility information
- Package version control:
  - {virtualenv}, {venv} (python virtual environments)
  - {recipy}
  - {sumatra}
- Manuscript writing:
  - Jupyter Notebooks
  - Alternatively, you can use any of the R packages for manuscript writing from the previous slide and run Python code within them using the {reticulate} R package
- GitHub Actions guide:
  - Documentation



RESOURCES

- Learn Git and GitHub:
  - Happy Git and GitHub for the useR
  - GitHub Learning Lab
  - Resources
  - GitHub Desktop Documentation
- Learn Jamovi:
  - Documentation
  - Textbook
- Learn JASP:
  - Textbook
- Learn R:
  - Online Books
  - Learn R, in R (Swirl package)
- Learn Python:
  - Python for Data Analysis
  - Automate the Boring Stuff With Python
- Learn Docker:
  - Documentation
  - Hands-on Tutorials



OPEN SCIENCE PODCASTS

- RIOT Science Club:
  - Is there a reproducibility crisis in science?
  - Improve your workflow for reproducible science
  - Five selfish reasons for working reproducibly
  - Synthetic data: A primer
  - Reproducibility in neuroimaging: Problems and solutions
  - JASP and Jamovi
  - Reproducibility in psychiatric genetics
- Reproducibilitea:
  - Reproducibility now
- Everything Hertz:
  - Predicting the replicability of research
  - Large-scale collaborative science
  - Academic hipsters
  - A manifesto for reproducible science
  - Data Sharing
  - Software and coding