Links to content

Notepad (this doc): https://bit.ly/2024_UNSW_R_intermediate

- Use this for shared notes on each section. This will be gradually migrated into the github repository (below)

Github repository: https://github.com/nicercode/2024_R_intermediate_UNSW/

- We will build this up as we go

Slidedeck (view only):

- https://docs.google.com/presentation/d/1zALE1QyV-8OryxPk80ahpL1UcTGCXISs03NRBXEN-Ls/edit#slide=id.g2edc875595f_0_54

Schedule

- 08:30-09:00: pre-class installation issues
- 09:00-10:30: Session 1 - Welcome, Challenges to Reproducible Research
- 10:00-10:30: Morning tea
- 11:00-12:30: Session 2 - Collaboration and Version control with Git & Github
- 12:30-13:15: Lunch
- 13:15-14:45: Session 3 - Techniques for Reproducible Research (part 1)
- 15:00-16:00: Session 4 - Techniques for Reproducible Research (part 2)

Session 1 - Reproducible research

What is ideal reproducible research?

- Transparent methods
- Accessible data and code
- Can be replicated
- Interoperable across operating systems and contexts

Historically, science was secretive but there has been a recent push to make science more open.

Many aspects of science can be made transparent, open, replicable. This course is about computation reproducibility.

Goal as an open scientist: Be able to reproduce the results at a later time, better if collaborator can do so, someone from research community.
There are processes to handle sensitive data (keep the science part open but the sensitive parts, private)

Activity: Identify challenges/problems/barriers for reproducible analyses and collaboration

Software environment

- **Packages** breaking, versioning, software out of data [+1] **GIMME** ABBY AND NEVE AND ALEXIS AND SAM **NO ONE TOUCH**
- Different **methods of recording code**

What is the problem?

- Getting different results when running code before and after updating the software
- ?help on the console
    - Very technical verbiage especially for people who don't know what 'objects' 'arrays' etc mean
- Some functions just don't work
- Last time a custom function was updated was in the early 2010s and never updated since
- Local computer can't handle the package processing - timing out or crashing

What is the solution?

- Emailing package authors, reading the user manual
- Finding someone else's data on figshare and running the code on THEIR data just in case my data was inherently funky to begin with (ISOLATE the problem)
    - Sample data that comes with the package
- Watching YouTube videos, using ChatGPT

- Sticking to file types I already know - knowing how to excel better so staying in that platform despite it taking longer
- Talking to the IT staff on campus if the packages aren't able to be downloaded within R-studio (katana version)

Workflow - Will, Mohammad, Shaina, Jigmidmaa

- **Project management** directory set up
    - R projects; using standard/intuitive file structures
    - Naming conventions
    - Data management plan - naming conventions, pipelines
- **Use of proprietary tech** and don't know how things are done (black box)
    - Make clear the software used and process you undertake
    - Might be able to cite a manual that explains the process
    - Try to engineer similar process in R - difficult.
- **Errors in code**
    - Troubleshooting (stack overflow; chat GPT; collaboration - have someone look at your code)
- **Cross-platform**, manual, non programmatic processing
    - Don't touch the raw data!
    - Automate as much as possible.

Coding conventions - madhu, craig, yannick

- Varied Coding style/stats preferences
- Naming conventions (e.g. for variables) - identify these in the documentation for the code or the metadata
- Adopting consensus conventions (e.g. Tidy R approach to coding and data management)
- Encountering issues midway through the analysis, which might tempt you or require you to change the data structure or naming conventions
- Collaborating with people who are at different stages of their coding experience or have particular views about how things should be done

Versions Lyra, Annabel, and Xiancheng

- **version** of data, cleaning practices

Problem:

- Package do not run in different versions of R
- Lose information when cleaning data or inconsistant data cleaning
- Collaboration can cause conflict

Solution:

- Use R pacakge and R markdown
- Keep raw data and cleaned data in sperate file instead of overwriting

- Standard naming convention and documentation
- Github
  Data cleaning/copy-paste from the website (vegetation species) - keep notes, preprocessing (R/Excel - clear format)
- **version** of R/packages - machines, HPC
- Install again (cran, github, settings), ask experts (who have done, NCI team)

Documentation – Green table (Erin, Courtney, Bina and Amelia))

- Is the package actually achieving what we want?
- **Social/cultural** aspects of data collection, how to manage while keeping reproducible
- Lack of instructions and guidance
- **Metadata** [+1]
  Language differences

Solution:

- Annotations used only when necessary, using formal language, omitting slang/colloquialisms
- Read.me files in a standard language explaining context of data collection
- Use standard formats, comments not in different languages & don't use slang.
- Removing old code that doesn't work

Data management - Steph, Ruby, Josh, Phoebe

- Edits on data, out of sync with code/analysis
- Naming conventions of **project files**
- Lack of Data management plans

PROBLEM:

- Syntax the researcher uses may not be coherent across code documents
- Code may be out of logical order due to issues during analysis
- Editing data outside of Rstudio

SOLUTION:

➔ Establish naming/data management system before starting (and stick with it)
  - ◆ Rstudio projects, GitHub repositories

➔ Clear workflow in script
  - ◆ Document every step of the process
    - Readme text files associated with code

➔ Rerun analysis before publishing files, troubleshoot

➔ Clear communication between collaborators

Big data

- how can I share large project files
- How to break this down to share
- Open in High Power Computer / Cluster Compute

[Feel free to add your own notes here! 😀]

Git with it

Please:
Sign up for a Github account
Install Github desktop

Notes on this section

- Txt files are best
- Git most difficult when people edit the same line/the same file

```
library(tidyverse)
data <- starwars
data %>% filter(height > 200)
```