INTRO TO REPRODUCIBLE RESEARCH

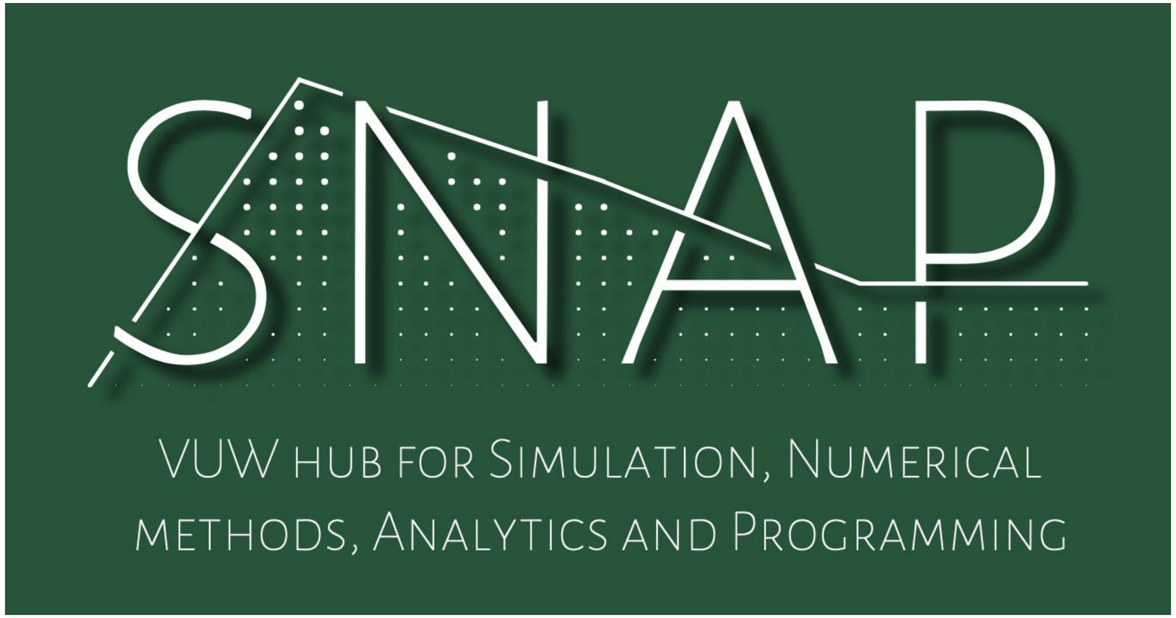
SNAP

VUW hub for Simulation, Numerical methods, Analytics and Programming

SNAP workshop
Feb 20th 2025

daniel.wrench@vuw.ac.nz

WHAT IS SNAP?



VUW hub for Simulation, Numerical methods, Analytics and Programming

- Interdisciplinary community of researchers who code, simulate, use supercomputers (e.g. Rāpoi), etc.
- Sharing expertise across subjects
- **Looking for student rep**
    - Helping connect with postgrad student community
    - Monthly meetings
    - Good for networking and CV

SUMMARY OF WORKSHOP

- **What is reproducible research?**
- **Creating a reproducible data pipeline**, as one would regularly encounter in scientific analysis or data science
    - **Good code repository structure**
    - **Using Git and GitHub**
    - **Virtual environments**
    - **Sharing your codes**
- Won’t cover:
    - Containers
    - Branches, pull requests and other intermediate/advanced aspects of Git (unless we have time and interest)
    - Object-oriented programming
    - Testing, `__init__.py` files, and other aspects of creating a piece of "software"

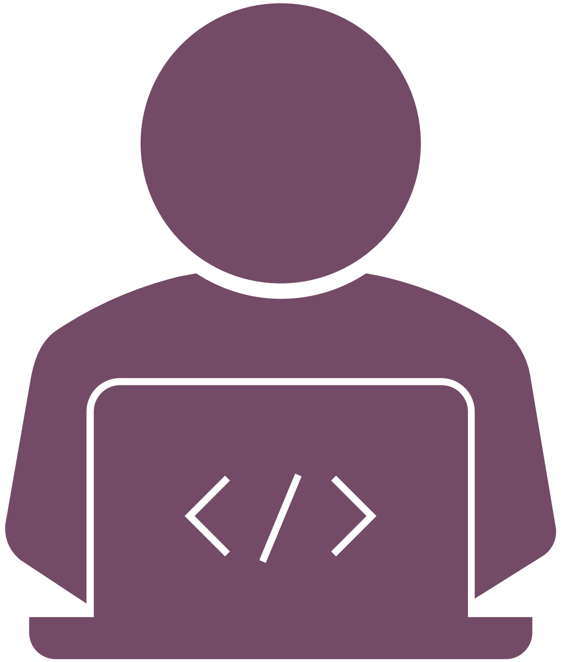
YOU WILL NEED

- The following programs installed
  - Git: check you are configured with `git config --list`
  - Python
- An account on GitHub.com
- A terminal or IDE of your choice (I'll be working in the terminal and VS Code)
- **Ask if you need help getting set up**

WHAT IS REPRODUCIBLE RESEARCH?

NICE PAPER: GOOD ENOUGH PRACTICES IN SCIENTIFIC COMPUTING (WILSON ET AL., 2017)

- AKA open science, sustainable research software
- **What do these terms mean to you?**
- Set of principles and practices in scientific computing (version control, documentation, virtual environments, etc.) that ensure
  - Collaboration
  - Longevity
  - Transparency
- Allows your work to be understood and trusted by
  - Yourself, 6 months from now
  - Your supervisor and colleagues
  - The scientific community at large: **the more open work is, the more widely it is cited and re-used**



TASK 1: **GET STARTED WITH A GIT REPO**

1. Understand Clone vs. Fork vs. Use template
2. Go to github.com, search **snap research template**
3.

-> Create a new repository
4. Give it your own name
5. Wonder at your perfectly-structured creation!
6. Code -> Local -> copy HTTPs URL
7. Open a terminal (in VS Code, Git Bash, whatever) and navigate to where you want to work
8. **`git clone paste_url_here`**
9. **`cd your-repo-name`** Check out structure, requirements.txt

OR START WITH AN EXISTING FOLDER ON YOUR COMPUTER

```
git init
(git remote add origin github_url.git)
```

TASK 2: **SET UP A VIRTUAL ENVIRONMENT**

1. **What is a virtual environment?**
2. Follow steps in README
3. In step 3, first **`pip install`** the following packages
   - `pandas`
   - `requests`
   - `matplotlib`
4. Make your first commit! (requirements.txt). Note changes in git status.

```
a.  git status
b.  git add requirements.txt
c.  git commit –m “concise but informative description of changes”
d.  git status
e.  git push
f.  git status
```

OR OTHER OPTIONS

```
Conda: environment.yml
R: renv
```

EXTRA FOR EXPERTS

Look at calmcode guide to pip-tools compile

TASK 3: **DOWNLOAD DATA**

1. Run the code in `scripts/` from the terminal or VS Code play button
2. **Can/should we commit this file?**
3. Delete from the terminal
4. Correct the `output_path` in the script
5. Re-run
6. Note **`git status`**
7. **What if you only wanted to ignore the raw data, and share the processed stuff?**
8. Pull up changes made to the script
9. Commit this change

TASK 4: **PLOT DATA AND UNDO MISTAKES**

1. **Make a plot of this data.** Up to you how to do it: doesn't need to be fancy. Think about where the code and output should go, a good name for it. Make sure can run from terminal.
2. Commit this file
3. Undoing saved (but **uncommitted** changes)
    1. Delete a bunch of the file and save it
    2. Check with **`git status`**
    3. Undo the change: **`git restore file_name`**
4. Undoing **committed** changes
    1. Change it again, commit, push
    2. Undo the latest commit: **`git revert HEAD`** ←

PLOTTING TIPS: MAY WANT TO INCLUDE

```
plt.xticks(rotation=45)
plt.tight_layout()
```

GO BACK TO A SPECIFIC COMMIT BY REPLACING HEAD WITH THE COMMIT HASH

DEMO OF USING GIT WITH HPC CLUSTER

SHARING YOUR REPO

- **Want reproducible analysis not just for ourselves or our colleagues, but for the whole scientific community.** Share data and software in your papers!
- **How could we do this?**
- All too common: *The data (and maybe poorly documented code) are available on reasonable request*
- Better: *Here's the link to my GitHub repo*
- Best: *The code is available on GitHub (link),* ***can be used under this license*** *and* ***is archived in Zenodo (citation with DOI)***
- Zenodo-GitHub integration -> CITATION.cff file in repo -> easy copy-and-paste BibTex citation
- **Finally, ensure you have good documentation for when they get the codes!** At minimum, a comprehensive README and metadata (explanation of the data and where it came from)

FINAL THOUGHTS

- Feel free to use my template however much you want – just remember the "Use this template" button
- This is all extra work, but it's worth it: for you, for your colleagues, and for science
- It's also not the whole picture: need tidy, readable, documented, modular code as well!
- ChatGPT and other LLMs are an invaluable tool – as long as you're not blindly copy-pasting!
  - For VS Code users, highly recommend installing Copilot. Limited version free to everyone, unlimited if you sign up to a GitHub student account
- Great resources:
  - Git Software Carpentry tutorial, including Chapter 10 on Open Science
  - Good enough practices in scientific computing (Wilson et al., 2017)
- Any volunteers for SNAP student rep?

ANY QUESTIONS?

INTRO TO REPRODUCIBLE RESEARCH

SNAP

VUW hub for Simulation, Numerical methods, Analytics and Programming

SNAP workshop
Feb 20th 2025

daniel.wrench@vuw.ac.nz