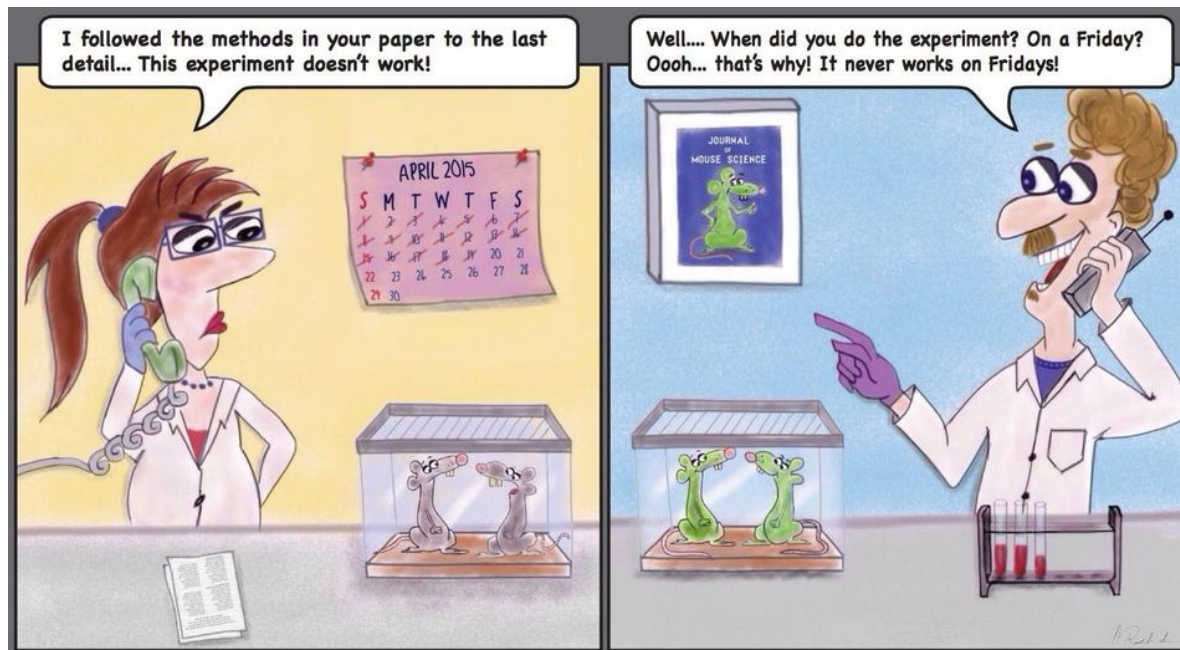


Introduction to Data Management, Reproducibility, and Integrity



Jiajia Li
12th, March 2024
PopGen in R Workshop at Kioloa NSW



Australian
National
University

Have you ever had problems reproducing your own or someone else's research?

#1 Yes, my own research.

#2 Yes, someone else's research.

#3 Yes, both.

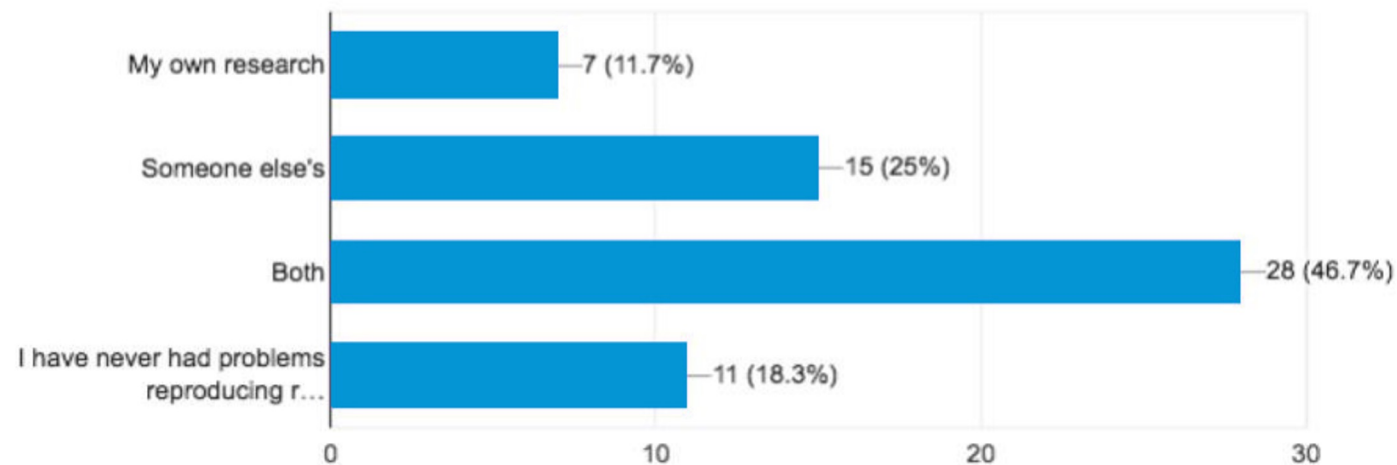
#4 No, I have never had problems reproducing published results.

[@repro4everyone](#) | repro4everyone.org



Have you ever had problems reproducing your own or someone else's research?

60 responses



[@repro4everyone](#) | repro4everyone.org



Goals and Objectives



The information lifecycle



Data management plan



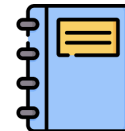
Naming your files



Structuring your project



Data and back-up



Electronic notebooks



Reproducible code



Version control



Bad data may lead to...



The information lifecycle



The information lifecycle illustrates how **data** (in all its various forms and derivatives, including *data points, datasets, databases, data files, visualisations, and code*) conceptually flows through its lifecycle of usefulness.

While data lifecycles are helpful frameworks to discuss **appropriate actions** taken at **different stages**, it's important to remember that for most data the path is not linear and some actions may not occur at all.



The information lifecycle



Think about ...

What data will you collect or create?

What documentation and metadata will accompany the data?

How will the data be stored and backed up during the research?

What analyses are you planning to do?

How will you share the data?

<https://data.uq.edu.au/data-essentials/information-lifecycle>

<https://data.wisc.edu/data-literacy/lifecycle/>

<https://dmptool.org/>





Data collection, experiment design

Information may be obtained through several means including manual data entry and automatic capture via devices or systems. **At the time information is acquired, key metadata should be recorded.**

Starting off with good data is much easier than remediating erroneous data down the line. After designing your collection method, reassess whether it worked as intended.

<https://data.uq.edu.au/data-essentials/information-lifecycle>

<https://online.hbs.edu/blog/post/what-is-data-integrity>





Metadata – “data that provides information about other data”

Why do we need metadata?

Good metadata can make up for human shortcomings.

People forget and misplace things, and leave research projects, taking their knowledge of the research methodology and the data with them.

Metadata ensure that data are **FAIR: Findable, Accessible, Interoperable, and Re-usable**.

<https://guides.lib.unc.edu/metadata/importance>





Metadata – “data that provides information about other data”

How to write metadata?

Stanford Libraries – [Create metadata for your research project](#)

The Carpentries Incubator – [Introduction to metadata](#)

ARDC – [Metadata guide](#)

<https://data.uq.edu.au/data-essentials/information-lifecycle>





Where to store your files?

How to back up your data?



Where to store your files?

Here I'll introduce some widely used data storage platforms in universities. To see what's available for your university, you can Google "Research data management Uni_Name".

- **OneDrive.** You can sync OneDrive to your desktop File Explorer/Finder for easy access and navigation.
- **Teams/SharePoint.** Teams/SharePoint storage can be used for storing shared documents for collaboration and team use. You can add shortcuts to OneDrive for easy access and navigation.
- **Network (Shared) Drives.** These drives normally provided by the university and can be seen by all staff members in a work group who have access to the shared folders.



Where to store your files?

- **Digital Research Notebooks.** [LabArchives](#) is a secure, cloud based collaborative research data management and electronic laboratory notebook platform (with version control).
- **HPCs.** University high performance computers and NCI.
- **Offline Storages.** Local device storage (e.g. laptop/computer), USB flash drives, portable hard drives, SSDs.



Data Tips

- Treat your raw data as read-only.
- Once the data is collected it should not be edited.
- List the changes you made with each new file in a MS Word/Markdown document.
- Use Git or other version control system.
- Always keep raw data.
- Always back up data.

@repro4everyone | repro4everyone.org

https://bookdown.org/arnold_c/repro-research/2-2-structuring-a-project.html



Do you know any data backup strategies?



3-2-1 Backup Rule

- Keep at least **three (3)** copies of data. The primary production data and two backup copies.
- Store **two (2)** backup copies on different storage media. (e.g. laptop, computer cluster, personal drive, network drive)
- Store **one (1)** backup copy offsite. (e.g. OneDrive, other cloud services)

<https://www.nakivo.com/blog/3-2-1-backup-rule-efficient-data-protection-strategy/>



3-2-1 Backup Rule

This rule is a universally accepted strategy within the IT industry and beyond.

By applying this rule, you ensure that data can be recovered in almost any failure scenario.

Keeping 3 copies of data is the **bare minimum** required to ensure that you can recover in any failure scenario, keep recovery objectives low and avoid a single point of failure.

The more backup copies you have, the less likely it is that you would lose all at once.

<https://www.nakivo.com/blog/3-2-1-backup-rule-efficient-data-protection-strategy/>





How to name your files?

How to structure your project folder?

How will you handle versioning?



Which type of person are you?



**Naming a file
relevant to the
project**

Jsjkdajkfk



Are you familiar with these feelings?

I cannot find this file!

Where is my file??

What version was it??



What did I call it again??

Was it the wild type picture or the mutant one?

Where is my RAW data???

[@repro4everyone | repro4everyone.org](https://repro4everyone.org)





File naming conventions

Avoid **extra long folder names** and **complex hierarchical structures** but use **information-rich** filenames instead.

➤ Don'ts:

- Z:\Production\Quality Control\Assembly Line7\Work Orders\Clients\Suzuki Motors\LOT3688_July-25-2009.xls
- Z:\Publications\Articles\eXadox\File-Naming-Conventions_V03.doc

➤ Dos

- Z:\Prod\QC\AssL7_WO_Suzuki_L3688_20090725.xls
- Z:\Pubs\Article_eXadox_File-Naming-Conventions_V03.doc

<https://exadox.com/en/articles/file-naming-convention-ten-rules-best-practice>





File naming conventions

Put **sufficient elements** in the structure for easy retrieval and identification but **do not overdo** it.

➤ Don'ts

- NOVALEC_INVOICE.pdf
- FUJITSU_S1500_SPEC_Black_Desktop_Scanner_ModelReplacesS510.pdf

➤ Dos

- NOVALEC_37507_INVOICE_20090703.pdf
- FUJITSU_S1500_SPEC_Scanner.pdf

<https://exadox.com/en/articles/file-naming-convention-ten-rules-best-practice>





File naming conventions

Use the **underscore** (_) as element delimiter. Do not use spaces or other characters such as !
\$ % ^ & * () + , . ; :

➤ Don'ts

- SMITH-J AXA 7654-6 POLICY 20120915.pdf
- FUJITSU \$S1500\$ SPEC\$Scanner.pdf

➤ Dos

- SMITH-J_AXA_7654-6_POLICY_20120915.pdf
- FUJITSU_S1500_SPEC_Scanner.pdf

<https://exadox.com/en/articles/file-naming-convention-ten-rules-best-practice>





File naming conventions

Use the **hyphen** (-) to delimit words within an element or **capitalise** the first letter of each word within an element.

➤ Don'ts

- Smith John AIG 7654 6 POLICY 2009 09 15.pdf
- White Paper Structure file naming strategy.doc

➤ Dos

- Smith-John_AIG_7654-6_POLICY_2009-09-15.pdf
- WhitePaper_StructureFileNameStrategy.doc

<https://exadox.com/en/articles/file-naming-convention-ten-rules-best-practice>





File naming conventions

Dates should be ordered: Year, Month, Day. (e.g. YYYYMMDD, YYYYMM). Time should be ordered: Hours, Minutes, Seconds (HHMMSS).

➤ Don'ts

- RFQ375_Cables-Unlimited_BID_10152009-1655.pdf
- Nov-20-2009_AMATProj_Phase1_Report.doc

➤ Dos

- RFQ375_Cables-Unlimited+BID_20091015-1655.pdf
- 2009-11-20_AMATProj_Phase1_Report.doc

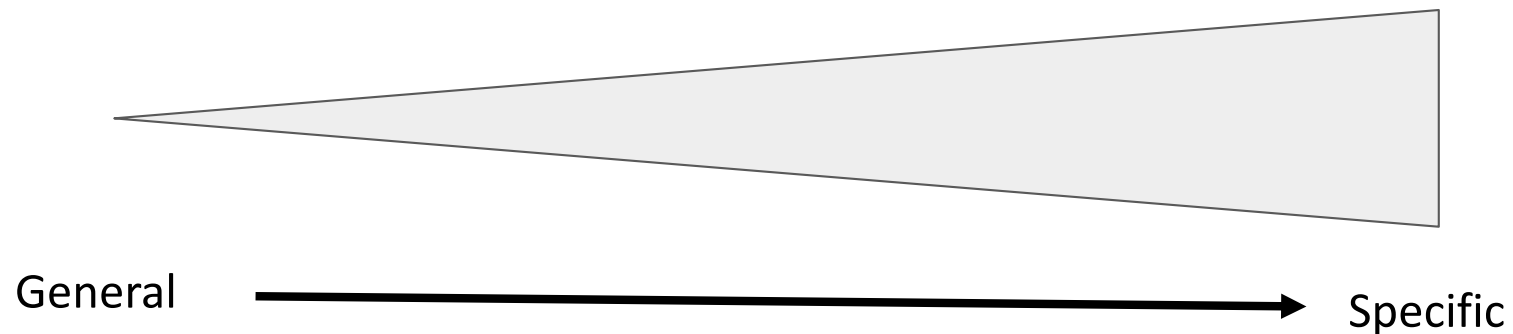
<https://exadox.com/en/articles/file-naming-convention-ten-rules-best-practice>



File naming conventions - Example

20130825_DOEProject_Ex1Test1_Data_Gonzalez_V3-03.xlsx

Project **Experiment** **Type** **ID** **Version**



<https://guides.lib.purdue.edu/c.php?g=353013&p=2378293>

@repro4everyone | repro4everyone.org





Structuring a project

The first step in creating reproducible research is creating **self-contained** projects.

Everything that goes in to, and comes out of, the project, should be contained within **a single folder** (directory).

I would recommend that you create a folder called **repos/** that your project folders live in, e.g. **C:/Users/owner/Documents/repos/proj_name/** . This way your project folders are neatly separated from other files.

https://bookdown.org/arnold_c/repro-research/2-2-structuring-a-project.html



Structuring a project

```
C:/
└─ Documents/
    └─ repos/
        └─ proj/
            ├── data/
            ├── docs/
            ├── figs/
            ├── funs/
            ├── out/
            ├── cleaning.R
            └─ analysis.R
```

This is an example structure of a project folder.

You may want to find a variation on it that works for you, but the basic premise of keeping repositories **self-contained** should remain.

https://bookdown.org/arnold_c/repro-research/2-2-structuring-a-project.html






Version control

- List the changes you made with each new file in a MS Word/Markdown document.
- OneDrive and LabArchives both have version control on your files.
- Version control with Git or other version control software. We will spend some time to learn Git later.





Share and
reuse

Share and reuse

Who are you sharing with? Are there any restrictions on sharing?

How to share large files?

- Within your institution – Network drives, portable hard drives, school server etc.
- With external collaborators – NCI (National Computational Infrastructure), Dropbox etc.

Put data into a public repository. e.g. NCBI, EBI.

Find a public data repository – Nature, [Data Repository Guidance](#)





Retain and archive

Information should be retained while required and archived in line with any relevant record retention periods (varies with projects).

Pay attention to **data retention policies of computer clusters**. e.g. NCI Gadi, university computing infrastructure.

Normally they only allow data to be stored on the platform for a certain period of time.





Dispose or destroy

Information should be destroyed in an appropriate manner at the end of its useful life, ensuring that records are destroyed or transferred to the appropriate owner.

End of project, end of PhD, be sure to organise your data and related documentation/metadata for future use.





Plan and design

“**Plan and Design**” sits in the centre of the Information Lifecycle, as information management should be carefully planned, with activities designed to meet university needs and compliance requirements throughout the lifecycle.

Write a data management plan.



What a data management plan should include?

- a back-up strategy
- an existing data survey
- outline of data to be created
- file format guidance
- metadata instructions
- ownership, access and security information
- data organisation and naming conventions
- information on managing data transfers and synchronisation between machines
- guidelines for collaborative writing with colleagues

<https://ardc.edu.au/resource/data-management-plans/>



What a data management plan should include?

- version control
- data storage locations
- hardware budget and management information
- bibliography management tools
- data sharing, publishing, archiving and licensing instructions
- data destruction rules
- responsibility allocations
- a budget for the overall DMP.

These covered most of the aspects and actions in the information lifecycle.

<https://ardc.edu.au/resource/data-management-plans/>





The [DMP Tool](#) is a free, open-source, application that helps researchers create data management plans (DMPs). These plans are required by many funding agencies as part of the grant proposal submission process.

The DMP Tool provides a click-through wizard for creating a DMP that complies with funder requirements.

It also has direct links to funder websites, help text for answering questions, and data management best practices resources.



Reproducibility and Replicability in Science

Reproducibility is obtaining consistent results using the same input data; computational steps, methods, and code; and conditions of analysis.

Replicability is obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data.

<https://www.ncbi.nlm.nih.gov/books/NBK547531/>



Factors decreasing reproducibility

Technical factors

- Contaminated cell lines
- Unvalidated antibodies and kits
- Natural variability
- Batch effects

Human factors

- Insufficient detail about methods, materials
- Poor sharing of reagents, data, and code
- Lack of version control
- Mistakes
- Publication bias

- Design flaws
- Misunderstanding statistics
- Confirmation bias
- Selective reporting

- Rewards for significant results, publication in high impact journals
- Lack of incentives for responsible research practices
- Hyper competitiveness
- Fraud

Study design & statistics

Rewards & incentives

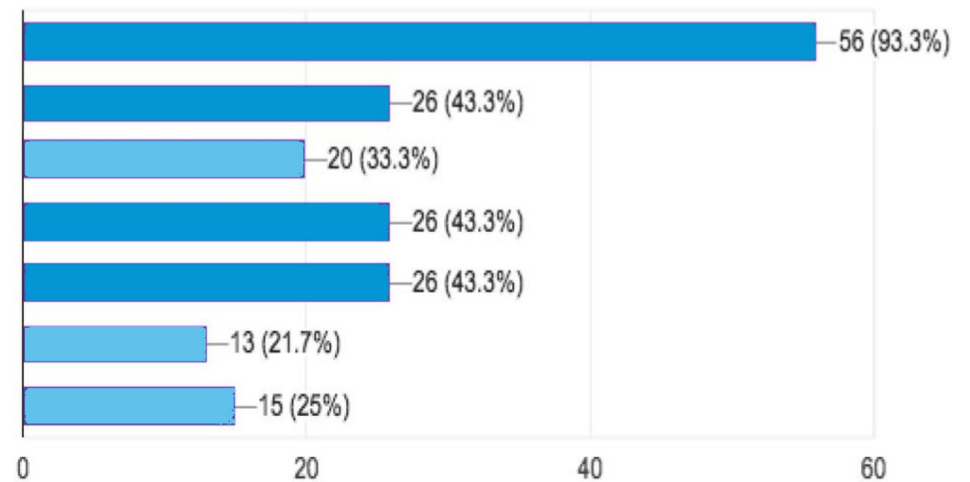
[@repro4everyone](https://repro4everyone.org) | repro4everyone.org



What are the major improvements that could lead to more reproducibility?

More detailed methods, analysis and record keeping
More publicly available data including meta-data

Fewer incentives to be first rather than right
Better reagent sharing e.g. plasmids, antibodies...



@repro4everyone | repro4everyone.org



What advantages does reproducible research have?

Higher citations rates

Happy future self

Money saved

Less reinventing the wheel

Happy funding agencies

Better reputation

Increased reuse of data

More funding?!?

New discoveries due to meta analysis

[@repro4everyone](https://twitter.com/repro4everyone) | repro4everyone.org



Fun exercise (10 mins) – Allegory of biology method drawing

1. Use a pen and a piece of paper
2. Imagine a beautiful owl. Everyone knows how an owl looks like. Draw the owl. Don't share it. Label it with your name and #1.
3. On a separate piece of paper, write detailed instructions and label it with your name and #2.
4. Hand in your sheet #2. Keep sheet #1.
5. Receive a sheet #2. Draw the owl according to the instructions. Label it with #3 and the name of the instructor.
6. Compare and share with the group.



How to draw an owl

1.



2.



1. Draw some circles

2. Draw the rest of the owl



Wet lab protocol sharing

Write as much detail as possible:

- duration/time per step
- reagent amount
- vendor name
- catalogue number
- expected result
- software package
- secret sauce

Use physical or electronic notebook to document your experiments as you go.

SciNote – [How to write a lab protocol](#)

Nature – [Five keys to writing a reproducible lab protocol](#)



[@repro4everyone](#) | [repro4everyone.org](#)



Electric Lab Notebooks



LabArchives is a web-based application designed for scientists to organise and share their laboratory data with their colleagues, friends, students, or anyone across the world.

Users may create an **unlimited** number of notebooks and customise the name and folder structure of each notebook.

Most of the universities in Australia use LabArchives as their official electronic lab notebook service. You can also sign up for free.

<https://help.labarchives.com/hc/en-us/articles/11723701830676-LabArchives-Introduction>



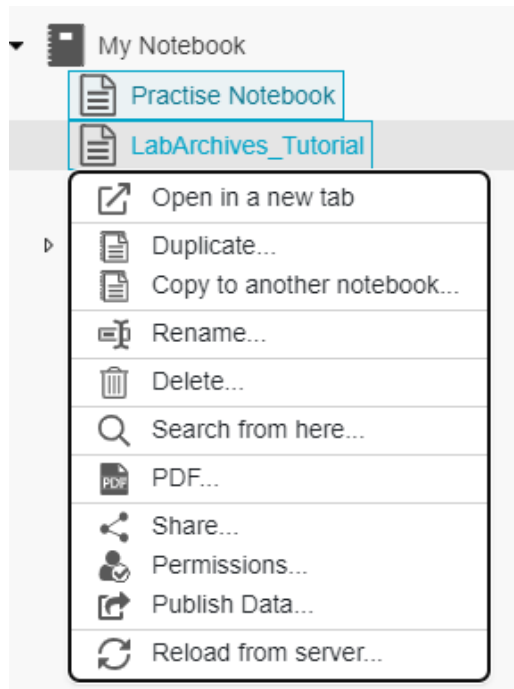
- Sign up or login with LabArchives [here](#).
- A place to store data permanently. 4GB maximum for a single file. (Insert attachment to add your data)
- The LabArchives team has created templates for different experiments, you can refer to it when creating your own notebooks. (Notebooks -> Lab Builder)
- It can be used on mobile devices such as IOS and Android phones and tablets. You can access your notebooks anywhere and anytime.
- Try to create a page, and add some sections and text, attach a file, insert a table.



- LabArchives stores every version of each entry and page that you make. Each revision is stored with an exact date and time stamp of when the entry was changed. You can always revert to a previous version of your data or undelete an item.

Date and Time	Entry version #	Revised by	Revised by ip	Revision Action	Data Type	Change	Revert Revision
06 Mar 2024, 1:26 pm AEDT	1	Jiajia Li	130.56.34.166	added	text entry	1.9 KB	
06 Mar 2024, 1:24 pm AEDT	1	Jiajia Li	130.56.34.166	added	heading	8 Bytes	revert to this version
06 Mar 2024, 12:51 pm AEDT	2	Jiajia Li	130.56.34.166	edited	plain text entry	162 Bytes	revert to this version
06 Mar 2024, 12:49 pm AEDT	1	Jiajia Li	130.56.34.166	added	plain text entry	151 Bytes	revert to this version
06 Mar 2024, 12:47 pm AEDT	1	Jiajia Li	130.56.34.166	added	heading	8 Bytes	revert to this version
06 Mar 2024, 12:47 pm AEDT	1	Jiajia Li	130.56.34.166	added	plain text entry	43 Bytes	revert to this version





- To share a notebook, folder, or page, **right click** on the item and click **Share**.
- To invite people to collaborate on the notebook, **right click** on the item and click **Permissions**.
- You can also export the notebook to PDF file.



Computational Reproducibility





Computational Reproducibility

Why is it so difficult to re-run other people's code?





Computational Reproducibility

When results are produced by **complex** computational processes using **large** volumes of data, the methods section of a scientific paper is insufficient to convey the necessary information for others to reproduce the results.

Additional information related to data, code, models, and computational analysis is needed for others to reproduce the results.

Researchers should convey **clear**, **specific**, and **complete information** about any computational methods and data products that support their published results in order to enable other researchers to repeat the analysis.

<https://www.ncbi.nlm.nih.gov/books/NBK547531/>





Information to ensure computational reproducibility

Data: the input data used in the study, as well as intermediate results and output data for steps that are non-deterministic.

Study methods: a detailed description of the study methods (ideally in executable form) together with its computational steps and associated parameters.

Computational environment: information about the computational environment where the study was originally executed, such as operating system, hardware architecture, and library dependencies. (or use containers)










<https://www.ncbi.nlm.nih.gov/books/NBK547531/>





Writing reproducible code

General principles of writing reproducible code

-  1. Work on your code iteratively
-  2. Prioritize readability over cleverness.
-  3. DRY up your code.
-  4. Don't be afraid to delete and refresh.
-  5. Use code comments effectively.
-  6. Use informative variable names.
-  7. Follow a code style.
-  8. Organize the layout of your code
-  9. Set the seed if your analysis has randomness involved.

https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/writing-durable-code.html#general-principles



Reproducible Analysis with Quarto



Quarto is a multi-language (R, Python, Julia), next-generation version of R Markdown from Posit and includes dozens of new features and capabilities while at the same being able to render most existing R markdown files without modification.





Rendering

- Open **hello.qmd** file in Posit Cloud, or download [hello.qmd](#) and open it using R studio. Let's try rendering this file.
- When rendering, Quarto generates a new file that contains selected text, code, and results from the **.qmd** file. Please compare the rendered result and the code. (render in new window or preview)
- The new file can be an HTML, PDF, MS Word document, presentation, website, book, interactive document, or [other format](#). (Try changing the output format, PDF/MS Word, find the name for MS word)

<https://quarto.org/docs/get-started/hello/rstudio.html#rendering>





Source vs Visual editor

- Visual editor will help you generate the source code for qmd file.
- In the source editor, you need to write the code yourself.
- Explore the visual editor and source editor. Try add some titles, formatted text, and a few line of code.
- If you don't know how to write Markdown text, we will introduce it later.





Quarto document (.qmd)

A Quarto document contains three types of content:

- A YAML header
- Code chunks
- Markdown text

<https://quarto.org/docs/get-started/hello/rstudio.html#authoring>





YAML header

```
---  
title: "Hello, Quarto"  
format: html  
editor: visual  
---
```

- An (optional) YAML header demarcated by three dashes (---) on either end.
- By writing these dashes, you tell the Quarto interpreter that this is a header.
- Between these dashes, the YAML syntax uses key-value pairs in the format **key: value**.

<https://quarto.org/docs/get-started/hello/rstudio.html#yaml-header>





YAML header

```
---  
title: "Hello, Quarto"  
format: html  
editor: visual  
---
```

- When rendered, the title, **"Hello, Quarto"**, will appear at the top of the rendered document with a larger font size than the rest of the document.
- The other two YAML fields denote that the output should be in **HTML format** and the document should open in the **visual editor** by default.

<https://quarto.org/docs/get-started/hello/rstudio.html#yaml-header>





YAML header

```
---  
title: "Hello, Quarto"  
format: html  
editor: visual  
---
```

- Other YAML fields commonly found in headers include metadata like **author**, **subtitle**, **date** as well as customisation options like **theme**, **fontcolor**, **fig-width**, etc.
- You can find out about all available YAML fields for [HTML](#), [PDF](#), and [MS Word](#) in the links.
- Try adding subtitle and changing fontcolor.

<https://quarto.org/docs/get-started/hello/rstudio.html#yaml-header>





Code chunks

```
```${r}
#| label: load-packages
#| include: false

library(tidyverse)
library(palmerpenguins)
```
```

- R code chunks identified with `{r}` with (optional) chunk options, in YAML style, identified by `#|` at the beginning of the line.
- In this case, the **label** of the code chunk is **load-packages**.
- And we set **include** to false to indicate that we don't want the chunk itself or any of its outputs in the rendered documents. (change it to true)

<https://quarto.org/docs/get-started/hello/rstudio.html#code-chunks>





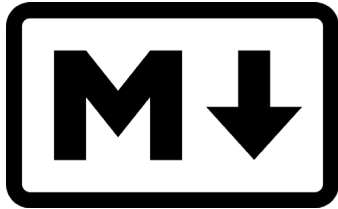
Code chunks

```
{r}  
#| label: plot-penguins  
#| warning: false  
#| echo: false
```

- What do these chunk options mean?

<https://quarto.org/docs/get-started/hello/rstudio.html#code-chunks>





Markdown

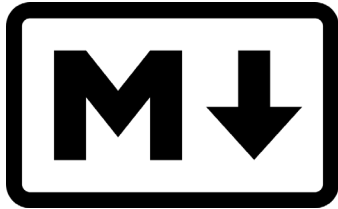
Markdown is a lightweight markup language for creating formatted text using a plain-text editor. It is intended to be easy to read in its source code form.

Markdown is widely used for blogging and instant messaging, and also used elsewhere in online forums, collaborative software, documentation pages, and readme files.

Markdown [cheat sheet](#).

<https://en.wikipedia.org/wiki/Markdown>





Markdown

You can upload Markdown files to GitHub directly and it will interpret it without rendering. See [here](#).

[VS Code](#) is a good editor for writing Markdown files, you can preview it as you write.

Personally I find it is very easy to create formatted documents once I got used to writing Markdown text.





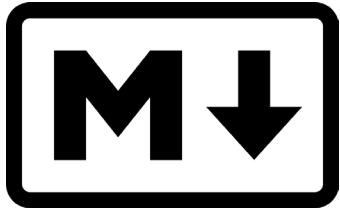
Additional resources

[Tutorial: Computations](#) – Learn how to tailor the behaviour and output of executable code blocks.

[Tutorial: Authoring](#) – Learn more about output formats and technical writing features like citations, crossrefs, and advanced layout.

<https://quarto.org/docs/get-started/hello/rstudio.html#code-chunks>





Markdown Exercise – 10 mins

<https://www.markdowntutorial.com/> is a good interactive lesson to learn Markdown, it will only cost you 10 minutes!





Learn Git version control – 20 minutes

<https://learngitbranching.js.org/> is a good website with interactive command line and anime to help you learn Git.



Exercise 30 mins – Write a Data Management Plan

1. Go to dmptool.org and sign up with your email address.
2. After logging in, go to your Dashboard and create a plan.
3. Based on your own research topic, fill in all the necessary information.
4. Download your plan and exchange it with another participant, pretending you are a funder and provide feedback to each other.
5. Compare your plan with other published plan on DMP tool.
https://dmptool.org/public_plans





Data integrity refers to the accuracy, consistency, completeness, and reliability of data throughout its entire lifecycle.

Good, detailed data management ensures data integrity.



Additional resources

- Callum Arnold – [A Guide to Reproducible Research](#)
- Swiss Institute of Bioinformatics – [Bioinformatics training materials](#)
- Candace Savonen - [Intro to Reproducibility in Cancer Informatics](#)
- Australia Research Data Commons – [Resource Hub](#)
- The Carpentries Incubator – [FAIR in \(biological\) practice](#)



Australian
National
University

Conclusion

Data management is important and vital because one of the responsibilities of researchers in the **Australian Code for the Responsible Conduct of Research** is to:

“Retain clear, accurate, secure and complete records of all research including research data and primary materials. Where possible and appropriate, allow access and reference to these by interested parties.” (R22)

Data management is a broad topic, and there is much more knowledge to learn. I hope this lesson has given you some new insights on it!!





Australian
National
University

THANK YOU

Contact Us

Research School of Biology | Biological Data Science Institute
ANU College of Science
The Australian National University

46 Sullivan's Creek Road, Acton ACT 2601

jiajia.li1@anu.edu.au



Australian
National
University