**OSSSG sixth session: Doing Reproducible Science**

**Date:** January 29, 2021

**Location:** Zoom

**Presenter:** Michael McCarthy

**Session description:** What is reproducibility and why is it important that your scientific works are reproducible? Reproducibility refers to a scientific pipeline whose steps, processes, procedures, and results can be reproduced by other scientists (or future you), ensuring that results can be verified, and that the decisions that led to those results can be understood. In our first session we will be joined by MSc student Michael McCarthy, who will share with you an opinionated introduction on doing reproducible science. We’ll discuss general principles and procedures, share some neat tricks to make your work better and easier to do, and go over resources you can use to make your science reproducible too.

**Link to video recording of presentation:**

**Link to slides:** <https://osf.io/xcn5g/>

**Challenges:**

Learn more about it!

* Work through [The Turing Way](https://the-turing-way.netlify.app/welcome.html), 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

**Resources:**

* Learn Git and GitHub:
  + [Happy Git and GitHub for the useR](https://happygitwithr.com)
  + [GitHub Learning Lab](https://lab.github.com)
  + [Resources](https://try.github.io)
  + [GitHub Desktop Documentation](https://docs.github.com/en/desktop)
* Learn Jamovi:
  + [Documentation](https://www.jamovi.org/user-manual.html)
  + [Textbook](https://www.learnstatswithjamovi.com)
* Learn JASP:
  + [Textbook](https://learnstatswithjasp.com)
* Learn R:
  + [Online Books](https://michaelmccarthy.netlify.app/post/books-for-learning-r/)
  + [Learn R, in R (Swirl package)](https://swirlstats.com/)
* Learn Python:
  + [Python for Data Analysis](https://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/Python-for-Data-Analysis.pdf)
  + [Automate the Boring Stuff With Python](https://automatetheboringstuff.com)
* Learn Docker:
  + [Documentation](https://docs.docker.com)
  + [Hands-on Tutorials](https://www.docker.com/play-with-docker)

Open science podcasts

* RIOT Science Club:
  + [Is there a reproducibility crisis in science?](https://youtu.be/AHSCyC6V-2g)
  + [Improve your workflow for reproducible science](https://youtu.be/fwZqVvHaA0M)
  + [Five selfish reasons for working reproducibly](https://youtu.be/S8bU1CyEkRM)
  + [Synthetic data: A primer](https://youtu.be/0fAR_oro1NY)
  + [Reproducibility in neuroimaging: Problems and solutions](https://youtu.be/dF0bKztTdFk)
  + [JASP and Jamovi](https://youtu.be/o5u_9qcc3U4)
  + [Reproducibility in psychiatric genetics](https://youtu.be/gWe9bnnU89A)
* Reproducibilitea:
  + [Reproducibility now](https://m.soundcloud.com/reproducibilitea/episode-4-reproducibility-now)
* Everything Hertz:
  + [Predicting the replicability of research](https://everythinghertz.com/94)
  + [Large-scale collaborative science](https://everythinghertz.com/78)
  + [Academic hipsters](https://everythinghertz.com/39)
  + [A manifesto for reproducible science](https://everythinghertz.com/35)
  + [Data Sharing](https://everythinghertz.com/18)
  + [Software and coding](https://everythinghertz.com/15)

Open-source alternatives

* Mendeley/Endnote alternative:
  + [Zotero](https://www.zotero.org) plus [Zotero Connector](https://www.zotero.org/download/connectors)
  + Import from [Mendeley](https://www.zotero.org/support/kb/mendeley_import) or [Endnote](https://www.zotero.org/support/kb/endnote_import)
* Useful Zotero plugins:
  + [scite](https://github.com/scitedotai/scite-zotero-plugin)
  + [pubpeer](https://github.com/PubPeerFoundation/pubpeer_zotero_plugin/releases/tag/v0.0.6)
  + [Better BibTeX](https://retorque.re/zotero-better-bibtex/)
  + [zotfile](http://zotfile.com)
  + [Sci-hub Downloader](https://medium.com/@gagarine/use-sci-hub-with-zotero-as-a-fall-back-pdf-resolver-cf139eb2cea7)
  + SPSS alternatives with GUI interface:
  + [Jamovi](https://www.jamovi.org)
  + [JASP](https://jasp-stats.org)
* Code-based SPSS alternatives:
  + [R](https://www.r-project.org) and [RStudio](https://rstudio.com)
  + [Python](https://www.python.org) and [RStudio v1.4+](https://blog.rstudio.com/2020/10/07/rstudio-v1-4-preview-python-support/)
  + [Julia](https://julialang.org)
* E-Prime/Presentation/Qualtrics/etc. alternatives:
  + [PsychoPy](https://www.psychopy.org)
  + [jsPsych](https://www.jspsych.org)
  + [Formr](https://formr.org)

Reproducibility in general

* Version control:
  + [Git](https://git-scm.com)
* Data and code distribution, collaboration:
  + [GitHub](https://github.com/) and [GitHub Desktop](https://desktop.github.com)
  + [OSF](https://osf.io/) and [osfr](https://github.com/ropensci/osfr)
* Data repositories:
  + [UCalgary Library Guide](https://library.ucalgary.ca/c.php?g=395022&p=5066153)
  + [Nature Recommended Data Repositories](https://www.nature.com/sdata/policies/repositories)
* Virtual environments:
  + [Docker](https://www.docker.com)
  + [Code Ocean](https://codeocean.com)
* Continuous Integration:
  + [GitHub Actions](https://docs.github.com/en/actions)
* Web hosting:
  + [Netlify](https://www.netlify.com/)

Reproducibility in R

* Use [RStudio Projects](https://r4ds.had.co.nz/workflow-projects.html)
* Use [inline R code](https://rmarkdown.rstudio.com/lesson-4.html) to report statistics
* Package version control:
  + [{renv}](https://github.com/rstudio/renv) (works at project level; use from the start)
  + [{groundhog}](https://github.com/CredibilityLab/groundhog) (works at script level; use to recover a reproducible environment in scripts without one)
  + [{holepunch}](https://github.com/karthik/holepunch) (easy Docker sessions)
* Codebooks:
  + [{codebook}](https://github.com/rubenarslan/codebook)
* Reproducible workflows:
  + [{worcs}](https://github.com/cjvanlissa/worcs)
  + [{targets}](https://github.com/ropensci/targets)
  + [{breakerofchains}](https://github.com/MilesMcBain/breakerofchains)
  + [Rocker](https://github.com/rocker-org/rocker)
* Reproducibility w/ interactive data:
  + [{shinymeta}](https://github.com/rstudio/shinymeta)
* Learn R Markdown:
  + [R Markdown Cookbook](https://bookdown.org/yihui/rmarkdown-cookbook/)
  + [R Markdown: The Definitive Guide](https://bookdown.org/yihui/rmarkdown/)
* Manuscript writing:
  + [{rmarkdown}](https://github.com/rstudio/rmarkdown)
  + [{bookdown}](https://github.com/rstudio/bookdown)
  + [{distill}](https://github.com/rstudio/distill)
  + [{rticles}](https://github.com/rstudio/rticles)
  + [{papaja}](https://github.com/crsh/papaja)
  + [{officeverse}](https://ardata-fr.github.io/officeverse/index.html)
* Machine-readable hypothesis testing:
  + [{scienceverse}](https://github.com/scienceverse/scienceverse)
* Unit testing:
  + [{testthat}](https://github.com/r-lib/testthat)
* GitHub Actions templates:
  + [r-lib/actions](https://github.com/r-lib/actions)
* Zotero connector:
  + [rbbt](https://github.com/paleolimbot/rbbt)
  + [citr](https://github.com/crsh/citr)

Reproducibility in Python

* Use Python Projects
* Use inline python code to report statistics
* [The Turing Way](https://the-turing-way.netlify.com/welcome.html) has more python reproducibility information
* Package version control:
  + [{virtualenv}](https://pypi.org/project/virtualenv/), [{venv}](https://docs.python.org/3/library/venv.html?highlight=projects) (python virtual environments)
  + [{recipy}](https://pypi.org/project/recipy/)
  + [{sumatra}](https://pypi.org/project/Sumatra/)
* Manuscript writing:
  + [Jupyter Notebooks](https://jupyter.org)
  + 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}](https://github.com/rstudio/reticulate) R package
* GitHub Actions guide:
  + [Documentation](https://docs.github.com/en/actions/guides/building-and-testing-python)