A Beginner's Guide to Conducting Reproducible Research

Jesse M. Alston[1,2] and Jessica A. Rick[1,3]

[1]*Program in Ecology, University of Wyoming, 1000 E University Dr., Laramie, WY 82072 USA*

[2]*Department of Zoology and Physiology, University of Wyoming, 1000 E University Dr., Laramie, WY 82072 USA*

[3]*Department of Botany, University of Wyoming, 1000 E University Dr., Laramie, WY 82072 USA*

Abstract

Reproducible research is widely acknowledged as an important tool for improving science and reducing harm from the "replication crisis", yet research in most fields within biology remains largely irreproducible. In this article, we make the case for why all research should be reproducible, explain why research is often not reproducible, and offer a simple framework that researchers can use to make their research more reproducible. Researchers can increase the reproducibility of their work by improving data management practices, writing more readable code, and increasing use of the many available platforms for sharing data and code. While reproducible research is often associated with a set of advanced tools for sharing data and code, reproducibility is just as much about maintaining work habits that are already widely acknowledged as best practices for research. Increasing reproducibility will increase rigor, trustworthiness, and transparency while benefiting both practitioners of reproducible research and their fellow researchers.

Key words: data management, data repository, software, open science, replication

Introduction

Replication is a fundamental tenet of science, but there is increasing fear among scientists that too few scientific studies can be replicated. This has been termed the "replication crisis" (Ioannidis, 2005; Schooler, 2014). Scientific papers often include inadequate detail to enable reproduction (Haddaway and Verhoeven, 2015; Archmiller et al., 2020), many attempted replications of well-known scientific studies have failed in a wide variety of disciplines (Bohannon, 2015; Hewitt, 2012; Moonesinghe et al., 2007; Open Science Collaboration, 2015), and rates of paper retractions are increasing (Cokol et al., 2008; Steen et al., 2013). Because of this, researchers are working to develop new ways for researchers, research institutions, research funders, and journals to overcome this problem (Peng, 2011; Sandve et al., 2013; Stodden et al., 2013; Fiedler et al., 2012).

Because replicating studies with new independent data is expensive, rarely published in high-impact journals, and sometimes even methodologically impossible, “reproducible research” is often suggested as a method for increasing our ability to assess the validity and rigor of scientific results (Peng, 2011). Research is reproducible when others can reproduce scientific results given only the original data, code, and documentation (Essawy et al., 2020). This commentary describes basic requirements for such reproducibility in biological research. In it, we make the case for why all research should be reproducible, explain why research is often not reproducible, and present a simple three-part framework all researchers can use to make their research more reproducible. These principles are applicable to researchers working in all types of biological research with data sets of all sizes and levels of complexity.

Why Do Reproducible Research?

Reproducible research benefits those who do it

Reproducible research is a by-product of careful attention to detail throughout the research process, and allows researchers to ensure that they can repeat the same analysis multiple times with the same results, at any point in that process. Because of this, researchers who conduct reproducible research are the primary beneficiaries of this practice.

First, reproducible research helps researchers remember how and why they performed specific analyses during the course of a project. This enables easier explanation of work to collaborators, supervisors, and reviewers, and it allows collaborators to conduct supplementary analyses more quickly and more efficiently.

Second, reproducible research enables researchers to quickly and simply alter analyses and figures. This is often requested by supervisors, collaborators, and reviewers across all stages of a research project, and expediting this process saves substantial amounts of time. When analyses are reproducible, creating a new figure may be as easy as changing one value in a line of code and re-running a script, rather than spending hours recreating a figure from scratch.

Third, reproducible research enables quick reconfiguration of previously conducted research tasks so that new projects that require similar tasks become much simpler and easier. Science is an iterative process, and many of the same tasks are performed over and over. Conducting research reproducibly enables researchers to re-use earlier materials (e.g., analysis code, file organization systems) to execute these common research tasks more efficiently in subsequent iterations.

Fourth, conducting reproducible research is a strong indicator to fellow researchers of rigor, trustworthiness, and transparency in scientific research. This can increase the quality and speed of peer review, because reviewers can directly access the analytical process described in a manuscript. Peer reviewers' work becomes easier and they may be able to answer methodological questions without asking the authors. It also protects researchers from accusations of research misconduct due to analytical errors, because it is unlikely that researchers would openly share fraudulent code and data with the rest of the research community. In addition, reviewers can check whether code matches with methods described in the text of a manuscript, to make sure that authors correctly performed the analyses as described. Finally, it increases the probability that errors are caught during the peer-review process, decreasing the likelihood of corrections or retractions after publication.

Finally, reproducible research increases paper citation rates (Piwowar et al., 2007; McKiernan et al., 2016) and allows other researchers to cite code and data in addition to publications. This enables a given research project to have more impact than it would if the data or methods were hidden from the public. For example, researchers can re-use code from a paper with similar methods and organize their data in the same manner as the original paper, then cite code from the original paper in their manuscript. Another researcher may conduct a meta-analysis on the phenomenon described in the two research papers, and thus use and cite both the two papers and the data from those papers in their meta-analysis. Papers are more likely to be cited in these re-use cases if full information about data and analyses are available (Whitlock, 2011; Culina et al., 2018).

Reproducible research benefits the research community

Reproducible research also benefits others in the scientific community. Sharing data, code, and detailed research methods and results leads to faster progress in methodological development and innovation because research is more accessible to more scientists (Mislan et al., 2016; Parr and Cummings, 2005; Roche et al., 2015).

First, reproducible research allows others to learn from your work. Scientific research has a steep learning curve, and allowing others to access data and code gives them a head start on performing similar analyses. For example, junior researchers can use code shared with the research community by more senior researchers to learn how to perform advanced analyses. This allows junior researchers to conduct research that is more rigorous from the outset, rather than having to spend months or years trying to figure out “best practices” through trial and error. Modifying existing resources can also save time and effort for experienced researchers—even experienced coders can modify existing code much faster than they can write code from scratch. Sharing code thus allows experienced researchers to perform similar analyses more quickly.

Second, reproducible research allows others to understand and reproduce a researcher’s work. Allowing others to access data and code makes it easier for other scientists to perform follow-up studies to increase the strength of evidence for the phenomenon of interest. It also increases the likelihood that similar studies are compatible with one another, and that all of these studies can provide evidence in support of or in opposition to a concept. In addition, sharing data and code increases the utility of these studies for meta-analyses that are important for generalizing and contextualizing the findings of studies on a topic. Meta-analyses in ecology and evolutionary biology are often hindered by incompatibility of data between studies, or lack of documentation for how those data were obtained (Stewart, 2010; Culina et al., 2018). Well-documented, reproducible findings enhance the likelihood that data can be used in future meta-analyses (Gerstner et al., 2017).

Third, reproducible research allows others to protect themselves from your mistakes. Mistakes happen in science. Allowing others to access data and code gives them a better chance to critically analyze the work, which can lead to coauthors or reviewers discovering mistakes during the revision process, or other scientists discovering mistakes after publication. This prevents mistakes from compounding over time and provides protection for collaborators, research institutions, funding organizations, journals, and others who may be affected when such mistakes happen.

Barriers to Reproducible Research

There are a number of reasons that most research is not reproducible. Rapidly developing technologies and analytical tools, novel interdisciplinary approaches, unique ecological study systems, and increasingly complex data sets and research questions hinder reproducibility, as does pressure on scientists to publish novel research quickly. This multitude of barriers can be simplified into four primary themes: (1) complexity, (2) technological change, (3) human error, and (4) concerns over intellectual property rights. Each of these concerns can contribute to making research less reproducible and can be valid in some scenarios. However, each of these factors can also be addressed easily via well-developed tools, protocols, and institutional norms concerning reproducible research.

Complexity. — Science is difficult, and scientific research requires specialized (and often proprietary) knowledge and tools that may not be available to everyone who would like to reproduce research. For example, analyses of genomic data require researchers to possess a vast base of knowledge about statistical methodologies and the molecular architecture of DNA, and genomic analyses are therefore difficult to reproduce for those with limited knowledge of the subject. Some analyses may require high-performance computing clusters that use several different programming languages and software packages, or that are designed for specific hardware configurations. Other analyses may be performed using proprietary software programs such as SAS statistical software (SAS Institute Inc., Cary, NC, USA) or ArcGIS (Esri, Redlands, CA, USA) that require expensive software licenses. Lack of knowledge, lack of institutional infrastructure, and lack of funding all make research less reproducible. However, most of these issues can be mitigated fairly easily. Researchers can cite primers on complex subjects or analyses to reduce knowledge barriers. They can also thoroughly annotate analytical code with comments explaining each step in an analysis, or provide extensive documentation on research software. Using open software (when possible) makes research more accessible for other researchers as well.

Technological change. — Hardware and software both change over time, and they often change quickly. When old tools become obsolete, research becomes less reproducible. For example, reproducing research performed in 1960 using that era's computational tools would require a completely new set of tools today. Even research performed just a few years ago may have been conducted using software that is no longer available or is incompatible with other software that has since been updated. One minor update in a piece of software used in one minor analysis in an analytical workflow can render an entire project less reproducible. However, this too can be mitigated by using established tools in reproducible research. Careful documentation of versions of software used in analyses is a baseline requirement that anyone can meet. There are also more advanced tools that can help overcome such challenges in making research reproducible, including software containers, which are described in further detail below.

Human error. — Though fraudulent research is often cited as reason to make research more reproducible (e.g., Ioannidis 2005; Laine et al. 2007; Crocker and Cooper 2011), many more innocent reasons exist as to why research is often difficult to reproduce (e.g., Elliott 2014). People forget small details of how they performed analyses. They fail to describe data collection protocols or analyses completely despite their best efforts and multiple reviewers checking their work. They perform sloppy analyses because they just want to be done with a project that feels like it is taking forever to complete. Science is performed by fallible humans, and a wide variety of common events can render research less reproducible.

While not all of these challenges can be avoided by performing research reproducibly, a well-documented research process can guard against small errors and sloppy analyses. For example, carefully recording details such as when and where data were collected, what decisions were made during data collection, and what labeling conventions were used can make a huge difference in making sure that those data can later be used appropriately or re-purposed. Unintentional errors often occur during the data wrangling stage of a project, and these can be mitigated by keeping multiple copies of data to prevent data loss, carefully documenting the process for converting raw data into clean data, and double-checking a small test set of data before manipulating the data set as a whole.

Intellectual property rights. — Researchers often hesitate to share data and code because doing so may allow other researchers to use data and code incorrectly or unethically. Other researchers may use publicly available data without notifying authors, leading to incorrect assumptions about the data that result in invalid analyses. Researchers may use publicly available data or code without citing the original data owners or code writers, who then do not receive proper credit for gathering expensive data or writing time-consuming code. Researchers may want to conceal data from others so that they can perform new analyses on those data in the future without worrying about others scooping them using the shared data. Rational self-interest can lead to hesitation to share data and code via many pathways. However, new tools for sharing data and code are making it easier for researchers to receive credit for doing so and to prevent others from using their data during an embargo period.

A Three-Step Framework for Conducting Reproducible Research

Conducting reproducible research is not exceedingly difficult, nor does it require encyclopedic knowledge of esoteric research tools and protocols. Whether they know it or not, most researchers already perform much of the work required to make research reproducible. To clarify this point, we outline below some basic steps toward making research more reproducible in three stages of a research project: (1) before data analysis, (2) during analysis, and (3) after analysis. We discuss practical tips that anyone can use, as well as more advanced tools for those who would like to move beyond basic requirements (Table 1). Most readers will recognize that reproducible research largely consists of widely accepted best practices for scientific research, and that striving to meet a reasonable benchmark of reproducibility is both more valuable and more attainable than researchers may think.

Before data analysis: data storage and organization

Reproducibility starts in the planning stage, with sound data management practices. It does not arise simply from sharing data and code online after a project is done. It is difficult to reproduce research when data are disorganized or missing, or when it is impossible to determine where or how data originated.

First, data should be backed up at every stage of the research process and stored in multiple locations. This includes raw data (e.g., physical data sheets or initial spreadsheets), clean analysis-ready data (i.e., final data sets), and steps in between. Because it is entirely possible that researchers unintentionally alter or corrupt data while cleaning it up, raw data should always be kept as a back up. It is good practice to scan and save data sheets or lab notebook pages associated with a data set to ensure that these are kept paired with the digital data set. Ideally, different copies should be stored in different locations and using different storage media (e.g., paper copies *and* an external hard drive *and* cloud storage) to minimize risk of data loss from any single cause. Computers crash, hard drives are misplaced and stolen, and servers are hacked—researchers should not leave themselves vulnerable to those events.

Digital data files should be stored in useful, flexible, portable, non-proprietary formats. Storing data digitally in a "flat" file format is almost always a good idea. Flat file formats are those that store data as plain text with one record per line (e.g., `.csv` or `.txt` files) and are the most portable formats across platforms, as they can be opened by anyone without proprietary software programs. For more complex data types, multi-dimensional relational formats such as `json`, `hdf5`, or other discipline-specific formats (e.g., `biom` and `EML`) may be appropriate.

However, the complexity of these formats makes them difficult for many researchers to access and use appropriately, so it is best to stick with simpler file formats when possible.

It is often useful to transform data into a 'tidy' format (Wickham, 2014) when cleaning up and standardizing raw data. Tidy data are in long format (i.e., variables in columns, observations in rows), have consistent data structure (e.g., character data are not mixed with numeric data for a single variable), and have informative and appropriately formatted headers (e.g., reasonably short variable names that do not include problematic characters like spaces, commas, and parentheses). Data in this format are easy to manipulate, model, and visualize during analysis.

Metadata explaining what was done to clean up the data and what each of the variables means should be stored along with the data. Data are useless unless they can be interpreted (Roche et al., 2015); metadata is how we maximize data interpretability across potential users. At a minimum, all data sets should include informative metadata that explains how and why data were collected, what variable names mean, whether a variable consists of raw or transformed data, and how observations are coded. Metadata should be placed in a sensible location that pairs it with the data set it describes. A few rows of metadata above a table of observations within the same file may work in some cases, or a paired text file can be included in the same directory as the data if the metadata must be more detailed. In the latter case, it is best to stick with a simple `.txt` file for metadata to maximize portability.

Finally, researchers should organize files in a sensible, user-friendly structure and make sure that all files have informative names. It should be easy to tell what is in a file or directory from its name, and a consistent naming protocol (e.g., ending the filename with the date created or version number) provides even more information when searching through files in a directory. A consistent naming protocol for both directories and files also makes coding simpler by placing data, analyses, and products in logical locations with logical names. It is often more useful to organize files in small blocks of similar files, rather than having one large directory full of hundreds of files. For example, Noble (2009) suggests organizing computational projects within a main directory for each project, with sub-directories for the manuscript (`doc/`), data files (`data/`), analyses (`scripts/` or `src/`), and analysis products (`results/`) within that directory. While this specific organization scheme may differ for other types of research, keeping all of the research products and documentation for a given project organized in this way makes it much easier to find everything at all stages of the research process, and to archive it or share it with others once the project is finished.

Throughout the research process, from data acquisition to publication, version control can be used to record a project's history and provide a log of changes that have occurred over the life of a project or research group. Version control systems record changes to a file or set of files over time so that you can recall specific versions later, compare differences between versions of files, and even revert files back to previous states in the event of mistakes. Many researchers use version control systems to track changes in code and documents over time. The most popular version control system is `Git`, which is often used via hosting services such as `GitHub`, `GitLab`, and `BitBucket` (Table 1). These systems are relatively easy to set up and use, and they systematically store snapshots of data, code, and accompanying files throughout the duration of a project. Version control also enables a specific snapshot of data or code to be easily shared, so that code used for analyses at a specific point in time (e.g., when a manuscript is submitted) can be documented, even if that code is later updated.

During analysis: best coding practices

When possible, all data wrangling and analysis should be performed using coding scripts—as opposed to using interactive or point-and-click tools—so that every step is documented and repeatable by yourself and others. Code both performs operations on data and serves as a log of analytical activities. Because of this second function, code (unlike point-and-click programs) is inherently reproducible. Most errors are unintentional mistakes made during data wrangling or analysis, so having a record of these steps ensures that analyses can be checked for errors and are repeatable on future data sets. If operations are not possible to script, then they should be well-documented in a log file that is kept in the appropriate directory.

Analytical code should be thoroughly annotated with comments. Comments embedded within code serve as metadata for that code, substantially increasing its usefulness. Comments should contain enough information for an informed stranger to easily understand what the code does, but not so much that sorting through comments is a chore. Code comments can be tested for this balance by a friend who is knowledgeable about the general area of research but is not a project collaborator. In most scripting languages, the first few lines of a script should include a description of what the script does and who wrote it, followed by small blocks that import data, packages, and external functions. Data cleaning and analytical code then follows those sections, and sections are demarcated using a consistent protocol and sufficient comments to explain what function each section of code performs.

Following a clean, consistent coding style makes code easier to read. Many well-known organizations (e.g., RStudio, Google) offer style guidelines for software code that were developed by many expert coders. Researchers should take advantage of these while keeping in mind that all style guides are subjective to some extent. Researchers should work to develop a style that works for them. This includes using a consistent naming convention (e.g., `camelCase` or `snake_case`) to name objects and embedding meaningful information in object names (e.g., using "`_mat`" as a suffix for objects to denote matrices or "`_df`" to denote data frames). Code should also be written in relatively short lines and grouped into blocks, as our brains process narrow columns of data more easily than longer ones (Martin, 2009). Blocks of code also keep related tasks together and can function like paragraphs to make code more comprehensible.

There are several ways to prevent coding mistakes and make code easier to use. First, researchers should automate repetitive tasks. For example, if a set of analysis steps are being used repeatedly, those steps can be saved as a function and loaded at the top of the script. This reduces the size of a script and eliminates the possibility of accidentally altering some part of a function so that it works differently in different locations within a script. Similarly, researchers can use loops to make code more efficient by performing the same task on multiple values or objects in series (though it is also important to note that nesting too many loops inside one another can quickly make code incomprehensible). A third way to reduce mistakes is to reduce the number of hard-coded values that must be changed to replicate analyses on an updated or new data set. It is often best to read in the data file(s) and assign parameter values at the beginning of a script, so that those variables can then be used throughout the rest of the script. When operating on new data, these variables can then be changed once at the beginning of a script rather than multiple times in locations littered throughout the script.

Because incompatibility between operating systems or program versions can inhibit the reproducibility of research, the current gold standard for ensuring that analyses can be used in the future is to create a software container, such as a `Docker` (Merkel, 2014) or `Singularity` (Kurtzer et al., 2017) image (Table 1). Containers are lightweight, standalone, portable environments that contain the entire computing environment used in an analysis: software, all of its dependencies, libraries, binaries, and configuration files, all bundled into one package. Containers can then be archived or shared, allowing them to be used in the future, even as packages, functions, or libraries change over time. If creating a software container is infeasible or a larger step than readers are willing to take, it is important to thoroughly report all software packages used, including version numbers.

After data analysis: finalizing results and sharing

After the steps above have been followed, it is time for the step most people associate with reproducible research: sharing research with others. As should be clear by now, sharing the data and code is far from the only component of reproducible research; however, once Steps 1 and 2 above are followed, it becomes the easiest step. All input data, scripts, program versions, parameters, and important intermediate results should be made publicly and easily accessible. Various solutions are now available to make data sharing convenient, standardized, and accessible in a variety of research areas. There are many ways to do this, several of which are described below.

Just as it is better to use scripts than interactive tools in analysis, it is better to produce tables and figures directly from code than to manipulate these using Adobe Illustrator, Microsoft Powerpoint, or other image editing programs. A large number of errors in finished manuscripts come from not remembering to change *all* relevant numbers or figures when a part of an analysis changes, and this task can be incredibly time-consuming when revising a manuscript. Truly reproducible figures and tables are created directly with code and integrated into documents in a way that allows automatic updating when analyses are re-run, creating a "dynamic" document. For example, documents written in LaTeX and `markdown` incorporate figures directly from a directory, so a figure will be updated in the document when the figure is updated in the directory (see Xie 2015 for a much lengthier discussion of dynamic documents). Both LaTeX and `markdown` can also be used to create presentations that can incorporate live-updated figures when code or data change, so that presentations can be reproducible as well. If using one of these tools is too large a leap, then simply producing figures directly from code—instead of adding annotations and arranging panels post-hoc—can make a substantial difference in increasing the reproducibility of these products.

Beyond creating dynamic documents, it is possible to make data wrangling, analysis, and creation of figures, tables, and manuscripts a "one-button" process using `GNU Make` (https://www.gnu.org/software/make/). `GNU Make` is a simple, yet powerful tool that can be used to coordinate and automate command-line processes, such as a series of independent scripts. For example, a `Makefile` can be written that will take the input data, clean and manipulate it, analyze it, produce figures and tables with results, and update a LaTeX or `markdown` manuscript document with those figures, tables, and any numbers included in the results. Setting up research projects to run in this way takes some time, but it can substantially expedite re-analyses and reduce copy-paste errors in manuscripts.

Currently, code and data that can be used to replicate research are often found in the supplementary material of journal articles. Some journals (e.g., *eLife*) are even experimenting with embedding data and code in articles themselves. However, this is not a fail-safe method of archiving data and analyses: supplementary materials can be lost if a journal switches publishers or when a publisher changes its website. In addition, research is only reproducible if it can be accessed, and many papers are published in journals that are locked behind paywalls that make them inaccessible to many researchers (Desjardins-Proulx et al., 2013; McKiernan et al., 2016; Alston, 2019). To increase access to publications, authors can post pre-prints of final (but pre-acceptance) versions of manuscripts on a pre-print server, or post-prints of manuscripts on post-print servers. There are several widely used pre-print servers (see Table 1 for three examples), and libraries at many research institutions host post-print servers.

Similarly, data and code shared on personal websites are only available as long as websites are maintained, and can be difficult to transfer when researchers migrate to another domain or website provider. Materials archived on personal websites are also often difficult for other scientists to find, as they are not usually linked to the published research and lack a permanent digital object identifier (DOI). To make research accessible to everyone, it is therefore better to use tools like data and code repositories than personal websites.

Data archiving in online repositories has become more popular in recent years, a trend resulting from a combination of improvements in technology for sharing data, an increase in omics-scale data sets, and an increasing number of publisher and funding organizations who encourage or mandate data archiving (Whitlock et al., 2010; Whitlock, 2011; Nosek et al., 2015). Data repositories are large databases that collect, manage, and store data sets for analysis, sharing, and reporting. Repositories may be either subject- or data-specific, or cross-disciplinary general repositories that accept multiple data types. Some are free and others require a fee for depositing data. Journals often recommend appropriate repositories on their websites, and these recommendations should be consulted when submitting a manuscript. Three commonly used general purpose repositories are `Dryad`, `Zenodo`, and `Figshare`; each of these creates a DOI that allows data and code to be citable by others. Before choosing a repository, researchers should explore commonly used options in their specific fields of research.

When data, code, software, and products of a research project are archived together, these are termed a "research compendium" (Gentleman and Lang, 2007). Research compendia are increasingly common, although standards for what is included in research compendia differ between scientific fields. They provide a standardized and easily recognisable way to organize the digital materials of a research project, which enables other researchers to inspect, reproduce, and extend research (Marwick et al., 2018).

In particular, the Open Science Framework (OSF; http://osf.io/) is a project management repository that goes beyond the repository features of `Dryad`, `Zenodo`, and `Figshare` to integrate and share components of a research project using collaborative tools. The goal of the OSF is to enable research to be shared at every step of the scientific process—from developing a research idea and designing a study, to storing and analyzing collected data and writing and publishing reports or papers (Sullivan et al., 2019). OSF is integrated with many other reproducible research tools, including widely used pre-print servers, version control software, and publishers.

Conclusions

While many researchers associate reproducible research primarily with a set of advanced tools for sharing research, reproducibility is just as much about simple work habits as the tools used to share data and code. We ourselves are not perfect reproducible researchers—we do not use all the tools mentioned in this commentary all the time and often fail to follow our own advice (almost always to our regret). Nevertheless, we recognize that reproducible research is a process rather than a destination and work hard to consistently increase the reproducibility of our work. We encourage others to do the same. Researchers can make strides toward a more reproducible research process by simply thinking carefully about data management and organization, coding practices, and processes for making figures and tables (e.g., Fig. 1). Time and expertise must be invested in learning and adopting these tools and tips, and this investment can be substantial. Nevertheless, we encourage our fellow researchers to work toward more open and reproducible research practices so we can all enjoy the resulting improvements in work habits, collaboration, scientific rigor, and trust in science.

Acknowledgements

Many thanks to J.G. Harrison, B.J. Rick, and A.L. Lewanski for providing helpful comments on early versions of this manuscript, and to C.A. Buerkle for inspiring this project during his Computational Biology course at the University of Wyoming.

References

Alston, J. M. (2019). Open access principles and practices benefit conservation. *Conservation Letters*, 12(6):e12672.

Archmiller, A. A., Johnson, A. D., Nolan, J., Edwards, M., Elliott, L. H., Ferguson, J. M., Iannarilli, F., Vélez, J., Vitense, K., Johnson, D. H., and Fieberg, J. (2020). Computational reproducibility in The Wildlife Society's flagship journals. *Journal of Wildlife Management*, 84(5):1012–1017.

Bohannon, J. (2015). Many psychology papers fail replication test. *Science*, 349(6251):910–911.

Cokol, M., Ozbay, F., and Rodriguez-Esteban, R. (2008). Retraction rates are on the rise. *EMBO reports*, 9(1):2–2.

Crocker, J. and Cooper, M. L. (2011). Addressing scientific fraud. *Science*, 334(6060):1182–1182.

Culina, A., Crowther, T. W., Ramakers, J. J. C., Gienapp, P., and Visser, M. E. (2018). How to do meta-analysis of open datasets. *Nature Ecology & Evolution*, 2(7):1053–1056.

Desjardins-Proulx, P., White, E. P., Adamson, J. J., Ram, K., Poisot, T., and Gravel, D. (2013). The case for open preprints in biology. *PLOS Biology*, 11(5):e1001563.

Elliott, D. B. (2014). The impact factor: a useful indicator of journal quality or fatally flawed? *Ophthalmic and Physiological Optics*, 34(1):4–7.

Essawy, B. T., Goodall, J. L., Voce, D., Morsy, M. M., Sadler, J. M., Choi, Y. D., Tarboton, D. G., and Malik, T. (2020). A taxonomy for reproducible and replicable research in environmental modelling. *Environmental Modelling & Software*, page 104753.

Fiedler, A. K., Landis, D. A., and Arduser, M. (2012). Rapid shift in pollinator communities following invasive species removal. *Restoration Ecology*, 20(5):593–602.

Gentleman, R. and Lang, D. T. (2007). Statistical analyses and reproducible research. *Journal of Computational and Graphical Statistics*, 16(1):1–23.

Gerstner, K., Moreno-Mateos, D., Gurevitch, J., Beckmann, M., Kambach, S., Jones, H. P., and Seppelt, R. (2017). Will your paper be used in a meta-analysis? Make the reach of your research broader and longer lasting. *Methods in Ecology and Evolution*, 8(6):777–784.

Haddaway, N. R. and Verhoeven, J. T. A. (2015). Poor methodological detail precludes experimental repeatability and hampers synthesis in ecology. *Ecology and Evolution*, 5(19):4451–4454.

Hewitt, J. K. (2012). Editorial policy on candidate gene association and candidate gene-by-environment interaction studies of complex traits. *Behavior Genetics*, 42(1):1–2.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLOS Medicine*, 2(8):e124.

Kurtzer, G. M., Sochat, V., and Bauer, M. W. (2017). Singularity: scientific containers for mobility of compute. *PLOS ONE*, 12(5):e0177459.

Laine, C., Goodman, S. N., Griswold, M. E., and Sox, H. C. (2007). Reproducible research: moving toward research the public can really trust. *Annals of Internal Medicine*, 146(6):450.

Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Prentice Hall, Upper Saddle River, NJ, USA.

Marwick, B., Boettiger, C., and Mullen, L. (2018). Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, 72(1):80–88.

McKiernan, E. C., Bourne, P. E., Brown, C. T., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B. A., Ram, K., Soderberg, C. K., Spies, J. R., Thaney, K., Updegrove, A., Woo, K. H., and Yarkoni, T. (2016). How open science helps researchers succeed. *eLife*, 5:e16800.

Merkel, D. (2014). Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239):2:2.

Mislan, K. A. S., Heer, J. M., and White, E. P. (2016). Elevating the status of code in ecology. *Trends in Ecology & Evolution*, 31(1):4–7.

Moonesinghe, R., Khoury, M. J., and Janssens, A. C. J. W. (2007). Most published research findings are false—but a little replication goes a long way. *PLOS Medicine*, 4(2):e28.

Noble, W. S. (2009). A quick guide to organizing computational biology projects. *PLOS Computational Biology*, 5(7):e1000424.

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D., Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson, E., McNutt, M., Miguel, E., Paluck, E. L., Simonsohn, U., Soderberg, C., Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers, E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research culture. *Science*, 348(6242):1422–1425.

Open Science Collaboration (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716.

Parr, C. S. and Cummings, M. P. (2005). Data sharing in ecology and evolution. *Trends in Ecology & Evolution*, 20(7):362–363.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060):1226–1227.

Piwowar, H. A., Day, R. S., and Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. *PLOS ONE*, 2(3):e308.

Roche, D. G., Kruuk, L. E. B., Lanfear, R., and Binning, S. A. (2015). Public data archiving in ecology and evolution: how well are we doing? *PLOS Biology*, 13(11):e1002295.

Sandve, G. K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):e1003285.

Schooler, J. W. (2014). Metascience could rescue the 'replication crisis'. *Nature*, 515(7525):9–9.

Steen, R. G., Casadevall, A., and Fang, F. C. (2013). Why has the number of scientific retractions increased? *PLOS ONE*, 8(7):e68397.

Stewart, G. (2010). Meta-analysis in applied ecology. *Biology Letters*, 6(1):78–81.

Stodden, V., Guo, P., and Ma, Z. (2013). Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals. *PLOS ONE*, 8(6):e67111.

Sullivan, I., DeHaven, A., and Mellor, D. (2019). Open and reproducible research on Open Science Framework. *Current Protocols Essential Laboratory Techniques*, 18(1):e32.

Whitlock, M., McPeek, M., Rausher, M., Rieseberg, L., and Moore, A. (2010). Data archiving. *American Naturalist*, 175(2):145–146.

Whitlock, M. C. (2011). Data archiving in ecology and evolution: best practices. *Trends in Ecology & Evolution*, 26(2):61–65.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(i10).

Xie, Y. (2015). *Dynamic documents with R and knitr*. CRC Press.

Tables

Table 1: A list of advanced tools commonly used for reproducible research, aggregated by function. This list is not intended to be comprehensive, but should serve as a good starting point for those interested in moving beyond basic requirements.

	Free	Open Source	Website
Data and Code Management			
Version control			
GitHub	Y[a]	N	https://github.com
BitBucket	Y[a]	N	https://bitbucket.com
GitLab	Y[a]	Y	https://www.gitlab.com
Make			
GNU Make	Y	Y	https://www.gnu.org/software/make/
Software containers and virtual machines			
Docker	Y	Y	https://docker.com
Singularity	Y[a]	Y	https://syslabs.io
Oracle VM VirtualBox	Y	Y	https://virtualbox.org
Sharing Research			
Preprint Servers			
ArXiv	Y		https://arxiv.org/
bioRxiv	Y		https://www.biorxiv.org/
EcoEvoRxiv	Y		https://ecoevorxiv.org/
Manuscript creation			
Overleaf	Y[a]	Y	https://overleaf.com
TeXstudio	Y	Y	https://www.texstudio.org/
Rstudio	Y	Y	https://rstudio.org
Data Repositories			
Dryad	N		https://datadryad.org/
Figshare	Y[a]		https://figshare.com/
Zenodo	Y		https://zenodo.org/
Open Science Framework	Y		https://osf.io/

[a] free to use, but paid premium options with more features are available

Figure Captions

Figure 1. A ten-point checklist to guide researchers toward greater reproducibility in their research. Researchers should give careful thought before, during, and after analysis to ensure reproducibility of their work.

Figures

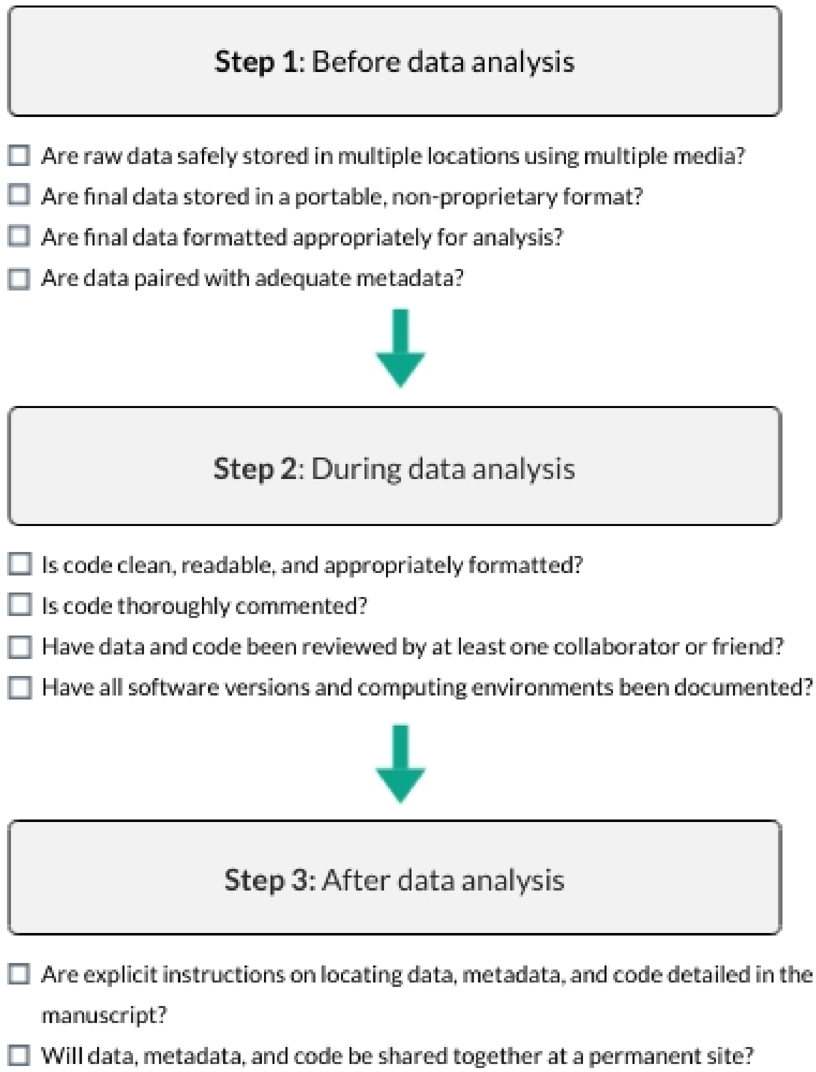

Figure 1: