Received: 4 March 2026 | Accepted: 23 April 2026

DOI: 10.1111/2041-210x.70338

BRITISH ECOLOGICAL SOCIETY Methods in Ecology and Evolution

PERSPECTIVE

Data- and code-archiving in the British Ecological Society journals: Present status and recommendations for future improvements

Natalie Cooper | the BES Data and Code Hackathon Group†

Science Group, Natural History Museum London, London, UK

Correspondence
Natalie Cooper
Email: natalie.cooper@nhm.ac.uk

Funding information
British Ecological Society

Handling Editor: Jeffrey W. Doser

Abstract

1. Data- and code-archiving are important components of open science, as both make research more transparent, reproducible, accountable and credible, allowing future researchers to build on previous work. Despite progress in implementing data- and code-archiving policies in journals publishing ecology and evolution research, issues remain. To be more useful to future researchers, archived data and code must not only be archived but also meet good practice standards.
2. We collected data from 1861 papers published between 2017 and 2024 in the seven British Ecological Society (BES) journals, during a hackathon event. We systematically checked associated data and/or code, metadata, help files and annotations to assess archiving practices. We determined if and where data and code files were archived, whether they could be located, downloaded and opened, and whether they had associated READMEs, digital object identifiers (DOI) and licences. We also recorded the file extensions used to save data/code files, and which programming languages code was written in.
3. 93% of the 1861 papers we examined used data and ~90% used code. While 97% of the 1735 papers that used data also archived it, only 35% of the 1670 papers that used code also archived code. Over 85% of archived data and code could be located, downloaded and opened. Reusability, however, was more limited; around a third of papers did not have a README or similar to explain their data/code files, and the quality of READMEs varied substantially.
4. We recommend that researchers archive their code and that archived code be explicitly mentioned in the Data (or Code) Availability statement. We also encourage researchers to provide more accessible and informative READMEs for data and code. To help achieve these recommendations, we advocate that journals employ Data/Code editors to review data and code quality, research institutions deliver more training in open science practices, and funding bodies set clear expectations on open data and code practices.

†Authors, ORCIDs and affiliations are listed in the Supporting Information. The list includes 137 authors from 27 countries across six continents.

KEYWORDS
code-sharing, data-sharing, open code, open data, open science, repository, research integrity, responsible research

1 | INTRODUCTION

Open science is a set of inclusive principles and practices that ensures scientific research and its outputs can be found, accessed, reused and built upon by everyone without restrictions (UNESCO, 2021). Archiving the data and code underlying published results is key to open science (Goldacre et al., 2019; O'Dea et al., 2021; Poisot et al., 2019; Whitlock, 2011). This allows others to reproduce analyses, identify errors and build on previous work, enhancing the quality, credibility and reach of research (Fernández-Juricic, 2021; Fidler et al., 2017; Powers & Hampton, 2019). Furthermore, data- and code-archiving offer several benefits for authors, especially for early career researchers, including higher impact, more collaborations, increased citations and enhanced employability (Allen & Mehler, 2019; Colavizza et al., 2024; Maitner et al., 2024; McKiernan et al., 2016; Piwowar et al., 2007; Poisot et al., 2013; Vandewalle, 2012). The open resources provided by data- and code-archiving may be particularly beneficial for researchers in the Global South (Noble et al., 2025), by helping to bridge gaps in funding or access to training and computing resources and connecting Global South researchers with the international scientific community. However, some authors remain reluctant to share their data and/or code due to perceived issues such as fear of 'scooping', lack of time and incentives, and insecurities about the quality of their data or code (Allen & Mehler, 2019; Evans, 2016; Gomes et al., 2022; Soeharjono & Roche, 2021).

Meta-research has shown that journals have a key role to play in encouraging data- and code-archiving (Ivimey-Cook, Sánchez-Tójar, et al., 2025; Powers & Hampton, 2019; Roche et al., 2022; Sholler et al., 2019). Over the last few decades, the scientific community has embraced data-archiving, and many funders, journals and some preprint servers (e.g. EcoEvoRxiv) now require data to be archived (Noble et al., 2025; Sholler et al., 2019). Most ecology and evolution journals have data-archiving policies (Berberi & Roche, 2023; Ivimey-Cook, Sánchez-Tójar, et al., 2025; Roche et al., 2022), and surveys show large improvements in compliance where archiving is mandatory (78% versus 7% sharing where data-sharing was mandatory versus optional; Vines et al., 2013) or where journals strongly encourage archiving (Maitner et al., 2024; Sánchez-Tójar et al., 2025). Code-archiving is equally important but less widely mandated (Maitner et al., 2024; Sánchez-Tójar et al., 2025), though an increasing number of ecology journals now have code-archiving policies (from 15% in 2015 to 88% in 2024; Culina et al., 2020; Ivimey-Cook, Sánchez-Tójar, et al., 2025; Mislan et al., 2016).

Despite the increase in data- and code-archiving policies and mandates across the publishing ecosystem, several issues remain. These include broken or outdated links to archived data/code, data or code that are incorrect or incomplete, file types that are proprietary or outdated and can no longer be opened, and absent or limited metadata and/or documentation making data or code impossible to interpret (Ivimey-Cook et al., 2023; Roche et al., 2022). This highlights that for data- and code-archiving to be effective, merely archiving data and code is insufficient; research outputs must also be FAIR, that is Findable, Accessible, Interoperable and Reusable (Wilkinson et al., 2016). An analysis of 100 ecology and evolution journal articles in 2022 found that only 46% of datasets met some reusability criteria of FAIR (Roche et al., 2022); though this is an improvement compared to 27% in 2015 (Roche et al., 2015). While data-archiving is now expected, code-archiving is still limited in ecology journals. Only 27% of 346 articles from 14 ecological journals with code-archiving policies archived their code (Culina et al., 2020), despite increased concerns about reproducibility in ecology (Kambouris et al., 2024; Kimmel et al., 2023). There is also a great deal of variety in data- and code-archiving policies across journals (Ivimey-Cook, Sánchez-Tójar, et al., 2025; Sánchez-Tójar et al., 2025), which may obscure patterns in broad cross-journal studies.

Here we focus on one group of journals with common editorial policies, the British Ecological Society (BES) journals. The BES publishes seven journals: *Ecological Solutions and Evidence (ESE)*, *Functional Ecology*, *Journal of Animal Ecology*, *Journal of Applied Ecology*, *Journal of Ecology*, *Methods in Ecology and Evolution (MEE)* and *People and Nature (PAN)*. Data-archiving has been mandatory, barring exceptional circumstances (e.g. indigenous data sovereignty; Carroll et al., 2020), at all seven journals since January 2014, and code-archiving has been required for papers presenting simulations, new applications and non-standard analyses since 2017 (2015 for *MEE*). Although journal staff check each accepted paper to ensure authors have archived data and code according to BES editorial policies (https://besjournals.onlinelibrary.wiley.com/hub/editorial-policies), they do not have the skills or capacity to guarantee all requirements for data- and code-archiving, for example completeness and reusability, have been met. Editors and reviewers with subject-specific expertise can check data and code during review, but this is uncommon (Natalie Cooper *pers. comm.*). Thus, the system assumes that authors have acted appropriately. We also do not know whether the data and/or code remain accessible long term. Here, we investigate data- and code-archiving in 1861 papers published in the BES journals between 2017 and 2024 (~20% of all papers published in the BES journals over this time period). BES journals have been among those leading the way in data- and code-archiving policies, thus these journals represent an ideal benchmark for best practice. We expected high data-archiving rates but potentially lower compliance for code. Our results provide a current snapshot of data- and code-archiving in papers published across the BES journals, allowing us to provide recommendations for future improvements that will benefit authors and readers alike.

2 | MATERIALS AND METHODS

2.1 | Data collection

2.1.1 | Assembling the list of papers

We collected data from papers published in the seven BES journals between 2017 and the end of 2024. *PAN* and *ESE* only began publishing papers in 2019 and 2020, respectively. We excluded reviews, perspectives, forum articles, commentaries and opinion pieces that rarely have data or code to archive, leaving 8112 eligible papers (Supporting Information B: Table S1).

2.1.2 | Collection of data by hackathon participants

Data were collected as part of a hackathon event (29-30th September 2025), where 145 (in-person and online) participants randomly selected papers from the 8112 eligible papers and then followed a bespoke protocol (see *Data collection protocol* below) to collect the required data. Participants systematically checked associated data and/or code, and examined available metadata, help files and annotations. To minimise input errors, participants submitted data for each paper according to the protocol via a form, using dropdown or multiple-choice menus whenever possible. Participants could share any queries through a dedicated online chat community (https://discord.com) to enhance a homogeneous assessment and data collection. In addition, all participants collected data for one common paper (paper number 2272) to explore data recorder variability.

2.1.3 | Data collection protocol

The full data collection protocol is in Supporting Information A. For each paper, participants first collected data on several general variables as follows. (1) paper number (these were randomly assigned to papers before data collection); (2) DOI; (3) publication year (dropdown); (4) journal (dropdown); and (5) identity of the data recorder(s). Data recorders were anonymised post-data collection and each unique data recorder, or group of data recorders, was given a recorder ID number.

Next, we extracted details about data- and code-archiving for each paper. Rather than list each question in our protocol (see Supporting Information A), we have grouped questions thematically and described how we collected data for each. Information about data and code were collected separately, but they are described together below. We used existing guidelines, for example FAIR, FAIR4RS, TADA!, European Open Science Cloud Research Software MetaData (RSMD), to develop our protocol; though our protocol does not cover every aspect of these (Barker et al., 2022; Gruenpeter et al., 2024; Ivimey-Cook, Culina, et al., 2025; Wilkinson et al., 2016).

1. Does the paper use data/code? If so, are they archived?

We used the Data Availability statement to determine whether data were used and archived. A paper used data if it generated or collated a dataset essential for reproducing the main results of the study. Some papers analysed only pre-existing datasets (this was most common in *MEE* papers), so did not generate data but often did use code. Pilot data collection indicated that code was rarely mentioned in the Data Availability statement, so to determine whether a paper used code we read the Methods section of the paper and identified statements that made it clear that code was used, even if it was not archived (e.g. 'analyses were conducted in R'). Data related to other questions listed below were only collected for papers that used data/code as appropriate.

2. Where were the data/code archived?

Not all archiving options are equally appropriate or accessible long term (Jones et al., 2025). Therefore, we recorded where the data/code were archived. Institutional or governmental repositories, or specific projects (e.g. MoveBank), were simplified as 'Other repo/database'.

3. Can the archived data/code be located, downloaded and opened?

We recorded whether the data/code were mentioned in the Data Availability statement, and whether we could find the data using the link or instructions provided in that statement. For code, which was rarely mentioned in the Data Availability statement (see Results), we also searched the main text for any links to archived code. We also recorded whether the data/code files could be downloaded at the time of investigation and, if so, whether they could be opened using standard software.

4. What file extensions do the archived data/code files use?

Data/code are not accessible if proprietary software (i.e. paid for use) is required to open them, because only researchers with access to that software will be able to access the data/code. To investigate this issue, we recorded the file extensions of archived data/code files. Each paper could contain multiple data/code files and thus multiple file extensions. We recorded all unique data/code file extensions for each paper. For archived code, we also recorded the programming language the code was written in.

5. Do the archived data/code have a *README* (or equivalent)? If so, how useful is it?

We recorded whether the data/code had a README and how useful this README was on a scale of 1–10. For data, READMEs **1** = *very brief and incomplete*, and **10** = *you can understand the dataset in just a few minutes. Contains all column headers, abbreviations, units, data sources,*

data dictionary, *licence info*, *paper info* etc. For code, READMEs **1** = *very brief and incomplete*, and **10** = *all information about script functionality, outputs, software, packages, workflows comprehensively documented*. A detailed description of a good data/code README was provided in the protocol (Supporting Information A; Box 1).

6. How complete is the archived data?

Archived data should contain all the data and metadata necessary to reproduce all the analyses and results. Issues could include providing only summary data, not providing raw data, only providing a subsample of the data and providing insufficient data to repeat the analyses. To assess this, we recorded the completeness of the data based on a comparison of the analyses featured in the manuscript and the archived data. Completeness is difficult to assess thoroughly without reproducing the analyses in the paper, but this requires time and advanced computational skills. As a compromise, we recorded completeness on a four point scale as follows. **low** = *the main analyses of the paper cannot be repeated with the data that has been archived*; **fair** = *some analyses can be repeated but not all (~50% of analyses can be repeated)*; **high** = *most data are provided with only small omissions, for example exploratory analyses (~75% of analyses can be repeated)*; **complete** = *all the data necessary to reproduce all analyses and results are archived*.

BOX 1 What is in a good data/code README?

- Information on the manuscript it came from.
- Contact details of at least one author.
- Licence information. Note that this can also be provided as a separate LICENCE file.
- Recommended citation for the data/code.
- A concise description of which data/code files are needed to reproduce specific analyses/figures/tables in the paper.
- For data:
  - A brief summary of how the data were collected, from where and when as appropriate.
  - Sources of data if it was from a literature review.
  - A list of all data files, whether they contain raw or processed/cleaned/summarised data, and briefly what they contain, for example life history variables.
  - Column-by-column description of the data files, along with column headers, measurement units, allowed options for categorical variables, explanations of any abbreviations and missing-data codes.
- For code:
  - A list of all scripts and what they do, that is processing, analysis, plotting etc. and what they output (e.g. table 1, figure 2). Detailed descriptions may be in the script files themselves, especially for functions, but the README should list the basics of what the scripts do.
  - Details of the workflow of the code if there are multiple scripts, that is what order do the scripts need to be run in?
  - Which data files are needed for each script.
  - The name of the software used (e.g. R), version, and names and versions of all packages required to run the analyses, along with any particular hardware or operating system requirements.

7. How good is the code annotation?

For code to be useful, it needs to be adequately annotated. We therefore recorded how good the code annotation was on a scale of 1–10, where **1** = *not annotated at all* and **10** = *thorough annotation throughout*. A detailed description of a good code annotation was provided in the protocol (Supporting Information A).

8. Are the data/code citable?

Citation is another aspect of reusability. To cite a data/code archive we generally use a DOI (Digital Object Identifier) or similar type of globally unique persistent identifier. A DOI is a unique number assigned to online digital objects, which enhances the accessibility, discoverability and interoperability of digital objects (https://www.doi.org/the-identifier/what-is-a-doi/). DOIs are more reliable than URLs, which can change over time. A DOI refers to a unique digital object, so data/code archives should have a separate DOI from the paper they are associated with. This has the additional benefit that if the data/code are reused in a later paper the data/code can be cited using the DOI (meaning you gain a citation), and authors do not need to create a new data/code archive. In addition, data/code should have a licence to be fully reusable. A licence informs future users of their rights and obligations when using the code (https://choosealicense.com/). From a legal point of view, without a licence any code you write is functionally proprietary because it is impossible to determine whether its reuse is legal or not. This is particularly important for anyone working for a company or government body with associated legal obligations. We recorded whether data/code had a DOI, and/or a licence and if so, what type of licence.

2.2 | Data analysis

The data were cleaned, summarised and plotted in R version 4.5.2 (R Core Team, 2025) using the following R packages: janitor v. 2.2.1 (Firke, 2016), naniar v. 1.1.0 (Tierney & Cook, 2023), patchwork v. 1.3.2 (Pedersen, 2025) and tidyverse v. 2.0.0 (Wickham et al., 2019). We cleaned all data to ensure consistency in how free text answers were categorised. We removed 23 records with data quality issues

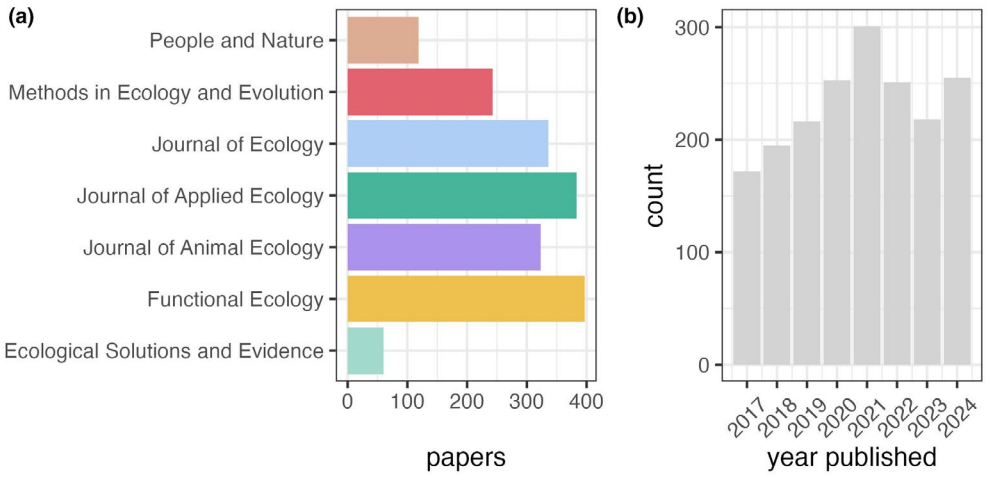

FIGURE 1 Number of papers in our dataset across (a) the seven BES journals and (b) years published (2017–2024). Colours in panel (a) are the official BES journal colours (https://besjournals.onlinelibrary.wiley.com/).

(papers 42, 1015, 1343, 1523, 1707, 1802, 1914, 3390, 3414, 3430, 3433, 3436, 3440, 5636, 6589, 6654, 6656, 6979, 7422 and 7453, plus two records where the paper number was not recorded and a duplicate of paper 190). All data and code required to reproduce the analyses are available on Zenodo at https://doi.org/10.5281/zenodo.19737911 (Cooper & BES Data and Code Hackathon Group, 2026) and on GitHub under the open source MIT licence at https://github.com/nhcooper123/reproduce-reuse-recycle.

To summarise and visualise the data we removed missing values and re-coded several options as follows. Data that were only available on request or were embargoed were coded as 'No' for data availability and data-archiving questions. Data or code that required specific software or were too large to be downloaded/opened were coded as 'Maybe' for questions about whether data/code could be downloaded/opened. Where participants specified that some, but not all, data or code files could be downloaded or opened, we re-coded this as 'Yes' on the principle that something is better than nothing.

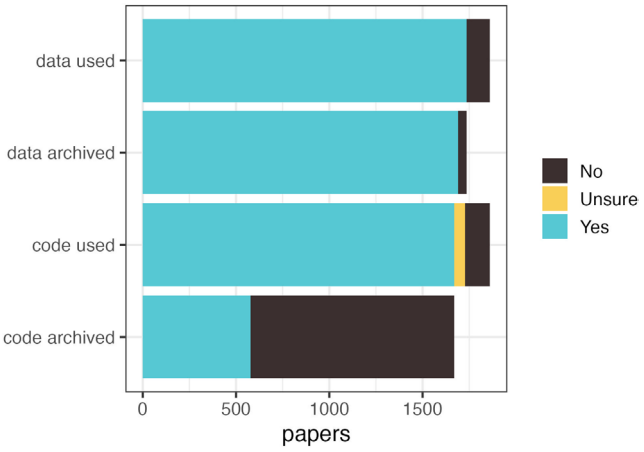

FIGURE 2 Number of papers in our dataset that use data or code, and whether these were formally archived. Data and code archived bars contain information only on papers that used data ($n = 1735$) or code ($n = 1670$) as appropriate.

3 | RESULTS

We collected data on 1861 papers (~20% of all papers published in BES journals from 2017 to 2024). The papers were fairly evenly spread across the seven journals and years and roughly reflected the percentages of papers published in each journal and/or year (Figure 1; Supporting Information B: Table S1). Results below focus on all journals combined, but journal-specific results are shown in Supporting Information B: Figures S1–S14. A Shiny app allowing readers to view the data in finer subsets (by journals and years etc.) is available at https://www.lewisajones.com/shiny-bes-hackathon/.

Data recorder agreement was high overall (mean % agreement ±SE = 91.7 ± 3.67). More details can be found in Supporting Information B. The README quality (data: median = 8, range 3–8; code: median = 4, range = 1–10) and code annotation quality (median = 7, range = 2–10) variables were less consistent among data recorders (Supporting Information B: Figure S15). These variables should therefore be interpreted with care.

1. Does the paper use data/code? If so, are they archived?

Of the 1861 papers in our dataset, 93% ($n = 1735$) used data and 90% ($n = 1670$) used code. Of these papers, 97% ($n = 1690$) archived their data, but only 35% ($n = 577$) archived their code (Supporting Information B: Table S2, Figure 2). These numbers differed significantly across BES journals (Supporting Information B: Tables S3–S5). Almost all papers (>90%) in all journals used data; the lowest percentage was for *MEE* (75%) where papers often present software tools without using newly generated data (range excluding *MEE* = 92%–98%). The percentage of papers using code was similarly high across all journals, except in the interdisciplinary journal *PAN* where only

62% used code (range excluding *PAN* = 88%–94%). Rates of code-archiving were low (overall 35%; range excluding *MEE* = 19%–34%) in all journals except *MEE*, for which 89% of papers had archived at least some code (Supporting Information B: Tables S3 and S4).

2. Where were the data/code archived?

Overall, 57% of data files were archived in Dryad ($n = 1022$), with smaller but substantial numbers archived in other repositories like Figshare (9%), Zenodo (12%) and various institutional and government repositories (17%; Figure 3). The biggest repository for code was Zenodo (39%; $n = 284$), followed by GitHub, GitLab, Codeberg or similar platforms (24%; $n = 177$; Figure 3). Note that 60% of code files that are hosted on GitHub or similar are also archived on Zenodo, which provides a permanent DOI. The BES archiving policy does not allow archiving at GitHub (or similar) alone.

3. Can the archived data/code be located, downloaded and opened?

Of the 1735 papers in our dataset that used data, 98% ($n = 1700$) mentioned the data in the Data Availability statement and 97% archived data ($n = 1690$; Supporting Information B: Table S2). For the 45 papers (<3%) where data were not archived, the data were either embargoed, available on request, or were not archived for privacy or security reasons. Of the 1690 archived datasets, 97% could be accessed via the links provided in the Data Availability statement, 95% could be downloaded, and 91% could be opened (range 72%–99%; Supporting Information B: Table S2). These numbers differed slightly across BES

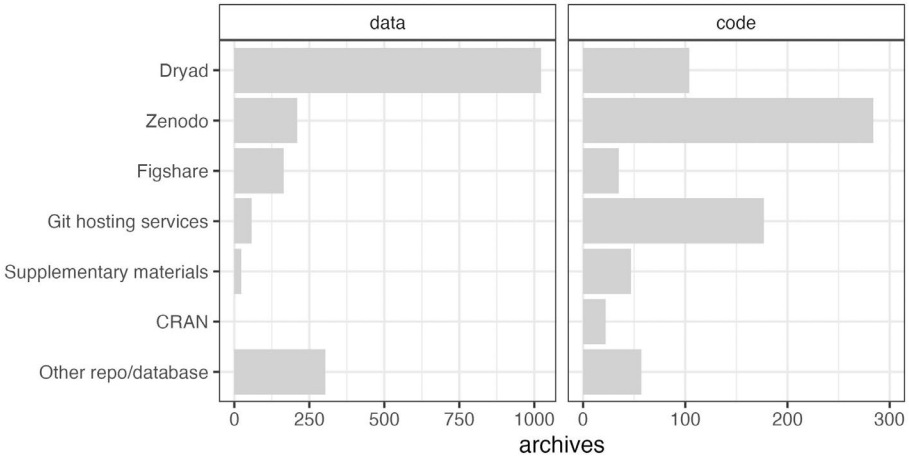

FIGURE 3 Archiving locations for archived data and code from the papers in our dataset. Note the different x-axis scales for data and code. Two data archives were archived on personal websites and have been omitted here to improve the clarity of the figure. Git hosting services represent an aggregate of GitHub, GitLab, Codeberg or similar platforms.

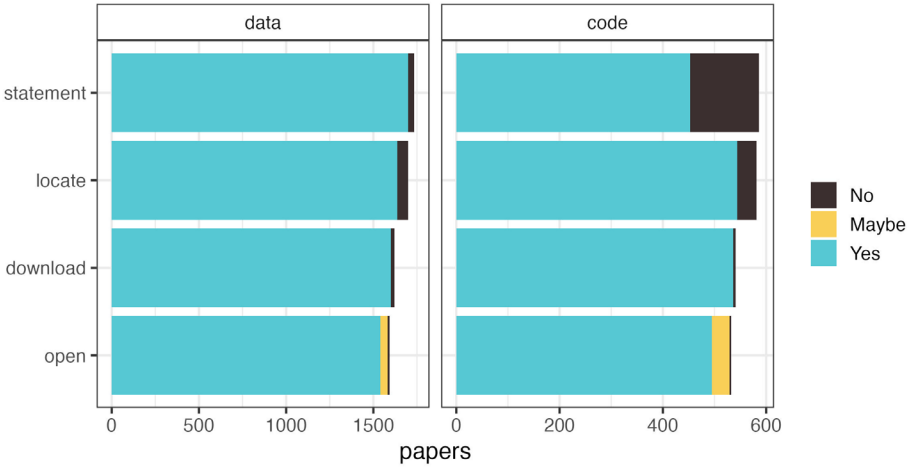

FIGURE 4 Number of papers where data or code were mentioned in the Data Availability statement, could be located via the link provided, could be downloaded and could be opened. These plots only contain papers where data ($n = 1690$) or code ($n = 577$) were archived. 'Maybe' refers to files that were either too large or required specialist software to download/open. Note the different x-axis scales for data and code.

journals (Supporting Information B: Table S3). A small number of datasets could not be downloaded or opened because they were too large or needed specific software to access ($n=41$; Figure 4).

Of the 1670 papers in our dataset that used code, only 35% ($n=577$) archived it, and only 27% ($n=453$, Supporting Information B: Table S2) mentioned code availability in the Data Availability statement. Of the 577 papers with archived code, 94% could be located, 93% could be downloaded, and 86% could be opened. These numbers differed slightly across BES journals; over 87% of archived files could be downloaded in all journals, whereas between 82% and 100% could be opened (Supporting Information B: Table S4). A small number of code files could not be downloaded or opened because they were too large or needed specific software to access ($n=34$; Figure 4).

4. What file extensions do the archived data/code files use?

Data were archived with 96 different file extensions, and many papers archived multiple files with different file extensions. Of the 2100 unique data archive file extensions recorded, 88% ($n=1857$) were saved with the following 10 file extensions: .csv/.tsv, .doc(x), .fasta, .pdf, .rda /.rdata/.rds, .shp, .tif, .xls(x), .xml (Figure 5). Code was primarily saved with seven different file extensions: .csv/.tsv, .doc(x), .html, .pdf, .txt, native source code (e.g. R, .py, .jl) or notebook files (e.g. .qmd, .Rmd). Of the 645 unique code file extensions recorded, 73% ($n=473$) were saved as native source code, for example R (.R), Python (.py) or Julia (.jl), and a further 16% ($n=101$) were saved as notebook files containing native source code, for example Quarto, RMarkdown or Jupyter notebooks (Figure 5). Most code (77%, $n=497$) was written in R (Supporting Information B: Figure S16).

5. Do the archived data/code have a README (or equivalent)? If so, how useful is it?

A README or equivalent was present for 66% ($n=1114$) and 61% ($n=351$) of the papers with archived data or code, respectively (Supporting Information B: Table S2). We did not record whether the same README was used for both data and code. The median README quality score was 7 for data and 6 for code (Supporting Information B: Figure 6a,b).

6. How complete is archived data?

Most datasets were scored as complete or as missing minor information (69% scored as complete; $n=1087$; Figure 6c). However, this should be considered an informed judgement about completeness, not verified reproducibility.

7. How good is the code annotation?

The median code annotation quality score was 7 (range 1–10; Figure 6d).

8. Are the data/code citable?

Of the 1690 papers with archived data, 91% ($n=1543$) had a DOI and 85% ($n=1438$) had a licence (Supporting Information B: Table S2), with CC0 licences ($n=1076$; the licence used by Dryad) being most used (Figure 7). Of the 577 papers with archived code, 79% ($n=457$) had a DOI and 74% ($n=428$) had a licence (Supporting Information B: Table S2), with CC BY ($n=122$; the licence Zenodo uses by default) and CC0 licences most used ($n=104$; Figure 7).

4 | DISCUSSION

The contrast between data- and code-archiving was a central finding of this study. Data-archiving policies have achieved high compliance, and most archived datasets were accessible and citable. Code-archiving, despite being required in many cases, remained

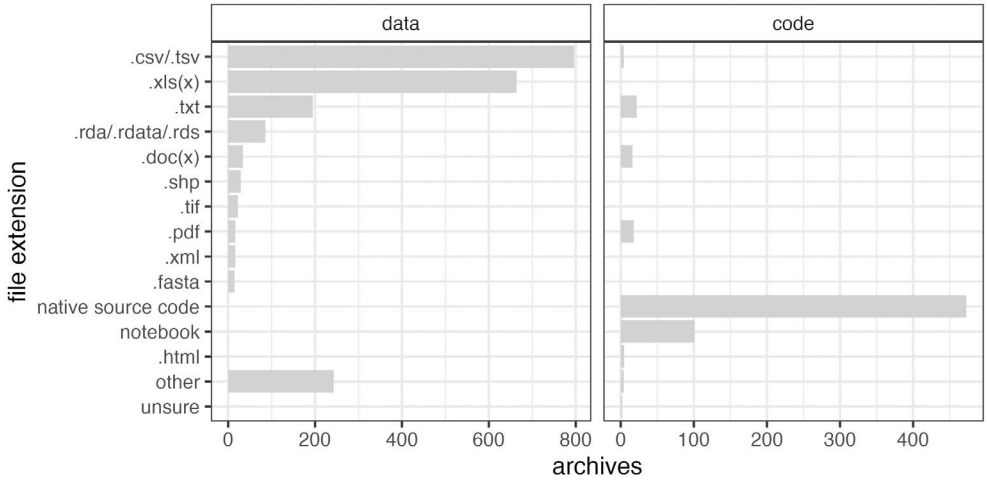

FIGURE 5 Number of files with unique file extensions recorded for archived data/code. Only the top ten most recorded data file extensions are displayed; other formats are grouped together as 'other'. Native source code is code saved with R (.R), Python (.py) or Julia (.jl), etc. file extensions. Notebooks are either Quarto, RMarkdown or Jupyter notebooks. Note the different x-axis scales for data and code.

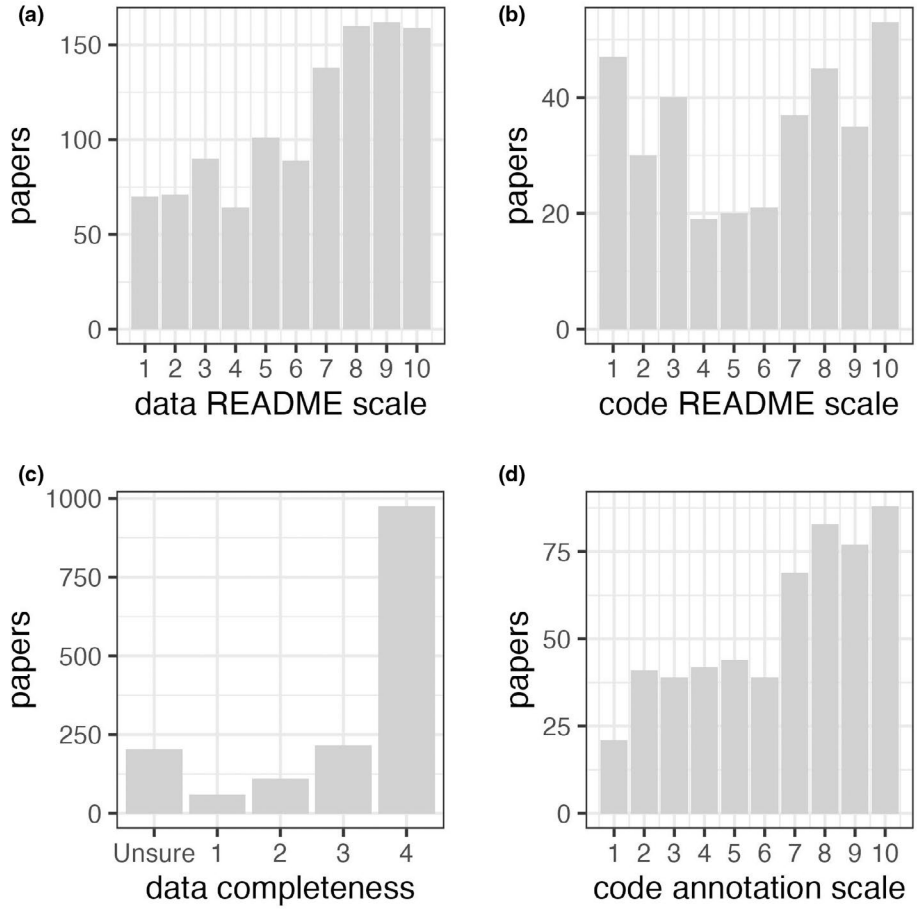

FIGURE 6 README usefulness (a, b), data completeness (c) and code annotation score (d). For READMEs, scores range from 1 = very brief and incomplete to 10 = comprehensive (see protocol). For data completeness, 1 = low, 2 = fair, 3 = high, 4 = complete and 'Unsure' means data recorders were unable to assess this. For code annotation, scores range from 1 = not annotated at all to 10 = thorough annotation throughout. Note the different y-axis scales across the four plots.

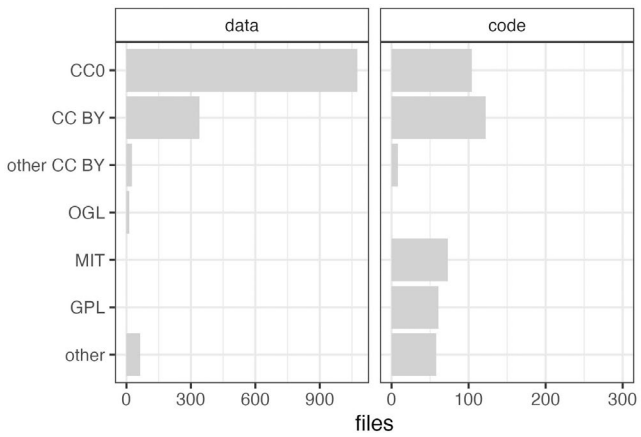

FIGURE 7 Number of files with unique licences for archived data and code. Note the different x-axis scales for data and code. Four files with 'Unsure' licence types have been removed from the code panel to improve readability. Other CC BY licences include CC BY-NC, CC BY-ND, CC BY-SA and other combinations of those restrictions.

uncommon; only 35% of papers using code archived it. For both data and code, however, the main limitation was not whether files could be located, downloaded or opened but whether they could be understood (and thus potentially reused) by someone outside the original research group. Documentation was frequently absent or insufficient, for example, around a third of papers lacked a README or similar to accompany their data and/or code, and many existing READMEs were not particularly useful. For code, even where code READMEs and annotation were rated highly, other factors are likely to impact the ability to reuse code, such as hard-coded paths, outdated package versions and changes to hardware, among other flaws. Tools exist that can help with these issues (e.g. renv, docker; Merkel, 2014; Ushey & Wickham, 2026) but uptake has been slow. Overall, while most data and code archived technically met good practice criteria, many papers met only minimum requirements.

Our results here show higher rates of compliance than in broader studies of ecology and evolution journals (Culina et al., 2020; Roche et al., 2015, 2022). This may reflect the presence of data- and code-archiving policies at the BES journals, supporting meta-research showing that instituting such policies improves compliance rates (Ivimey-Cook, Sánchez-Tójar, et al., 2025; Vines et al., 2013). However, good practice can plateau once policies are implemented unless further incentives are provided. The implementation and effectiveness of policies must therefore continue to be monitored and assessed. Allowing for reasonable exceptions due to privacy,

security or sovereignty concerns (Carroll et al., 2020), there may be little room for improvement in the amount of data archived in papers published by the BES journals, but improvements to other aspects of data-archiving are still possible. Rates of code-archiving require extensive improvement, as does the quality of data and code documentation. Below we have identified seven recommendations for future improvement.

Recommendations for authors:

1. **Archive your code**. We understand there are technological, time and expertise barriers to doing so, but there are resources that can help. The BES Guide to Reproducible Code (Cooper & Hsing, 2025) is an excellent starting point for learning more about making code more reproducible and thus easier to share. The TADA! and FAIR4RS guidelines are also excellent resources (Barker et al., 2022; Ivimey-Cook, Culina, et al., 2025). Software is an important output of many papers, thus retaining it for the future is of key importance (Di Cosmo & Zacchiroli, 2017). Archiving your code also has many benefits, especially for early career researchers, including higher impact, more collaborations, increased citations and enhanced employability (Allen & Mehler, 2019; Colavizza et al., 2024; Maitner et al., 2024; McKiernan et al., 2016; Piwowar et al., 2007; Poisot et al., 2013; Vandewalle, 2012). In addition, 99% of the time the future user of code will be future you! Thus any efforts made to improve the functionality of code before sharing it will disproportionately benefit the author of the code in the long term.
2. **Archive data and code in appropriate repositories, with a globally unique persistent identifier and using non-proprietary file types (e.g. .csv/.tsv)**. Supplemental materials are not appropriate, as these are not formally archived or connected to a globally unique persistent identifier like a DOI, so they can be (and have been) lost. Likewise, depositing code on GitHub, GitLab, Codeberg or similar is not sufficient because they are designed for version control, not archiving. A simple solution if using GitHub, GitLab, Codeberg or similar is to also deposit the version of the code used in the paper in a repository (e.g. Zenodo) that will provide a globally unique persistent identifier, for example a DOI. This is simple and quick to do via GitHub (https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content). DOIs are important because they provide stability when URLs change, for example when journals switch publishers and link to a 'version of record', that is the exact version of the data/code used in the paper. They also increase the reach and impact of your work and ensure that your data/code will always be findable and citable (https://www.doi.org/the-identifier/what-is-a-doi/). Data/code archives should have a separate DOI from the paper they are associated with. If the data/code are reused in a later paper, the data/code can be cited using the DOI, meaning you gain a citation!

   Additionally, archiving data and code with file extensions like .xls(x), .doc(x) and .pdf is not appropriate. Files that require proprietary (i.e. paid for use) software to open (for example various Microsoft products) are not accessible and create a paywall for researchers. Even if they can sometimes be opened with open alternatives (e.g. Open Office), some features may be obscured, and file contents may be corrupted. It is simple to save files in non-proprietary formats, for example Excel files can be saved as .csv using the 'Save As' option. This also has advantages for authors, as none of us know whether our next employer will have access to certain software. Thus by saving and sharing data/code in non-proprietary formats, authors can future-proof access to their own outputs, at the same time as making them more open and reproducible for others.

3. **Ensure minimum requirements for effective data- and code-archiving are met**. This includes providing all raw and processed data (or simulated data to run the code when the actual data cannot be shared, with any real processed/summarised data that can be shared), well-annotated code to run all analyses, a globally unique persistent identifier (e.g. a DOI), and appropriate licences and documentation for example READMEs (see Box 1) and worked examples or manuals. Optionally other data products such as supplemental materials from the paper could also be archived together with data and code. Note that data and code should have different licences as standard CC licences used for data are not intended for code (https://creativecommons.org/faq/#can-i-apply-a-creative-commons-license-to-software). As noted in recommendation (1), ensuring all components are present and functional also has advantages for authors, as often the future user of data/code will be the authors themselves, for example when revising the paper or expanding the project. This is much easier when authors can be confident that nothing is missing from their data/code (especially when continuing projects after changing computers and/or institutions) and that enough documentation exists to get them (and their collaborators/students) up to speed quickly.
4. **Finally**, all of the above are easier if we **develop the habit of thinking about data- and code-archiving throughout the life cycle of a project**, not just when writing a data management plan at the start and before submitting/publishing a paper. Data management plans are excellent tools but they need to be updated/implemented regularly to be useful.

Recommendations for journals:

5. **Journals should require a 'Data and Code Availability Statement' section** rather than the commonly used 'Data Availability Statement'. In the papers we assessed, data and code were regularly archived together but the Data Availability statement often referred only to the data. An explicit 'Data *and* Code Availability Statement' would support authors in structuring and reporting complete research archives of data and code. These statements should be presented in a structured and machine-readable format. Our study involved laborious manual

parsing of Data Availability statements, which hinders continued meta-research on the topics discussed. Through coordination between journals, publishers and funders, structured data and code availability statements could be key metadata to incorporate into publishing and funding management systems.

6. **Journals should ensure minimum requirements for effective data- and code-archiving are met**. This includes checking that (i) data/code have been archived in suitable repositories, not supplemental materials or solely at GitHub, GitLab, Codeberg or similar; (ii) data/code archives have a globally unique persistent identifier, for example a DOI and appropriate licences; (iii) data/code files are not saved in proprietary formats, for example .xls(x), .doc(x) or .pdf; (iv) all raw and processed data (or simulated data to run the code when the actual data cannot be shared, with any real processed/summarised data that can be shared) are provided; (v) well-annotated code to run all analyses is provided; and (vi) sufficient data/code documentation is provided, for example READMEs and worked examples or manuals.
7. **Journals should provide clear and explicit guidelines for preparing data/code archives, in particular READMEs**. In the papers we assessed, READMEs were sometimes a file, sometimes a section within a data document (e.g. a tab in Excel) and varied in quality. We recommend these be separate files and suggest some standards for what to include in Box 1. More journal or repository-specific templates and examples, along with training, would be helpful. We should aim for everyone to understand that the analysis is not finished until the README is completed to a high standard.

Journals have a major role to play in implementing these recommendations. We encourage journals to develop and enforce code-archiving policies (Ivimey-Cook, Sánchez-Tójar, et al., 2025), as it is difficult to make changes without incentives. Editors and reviewers should also be encouraged to pay more attention to data and code during assessment. One way to maximise the usefulness of data/code-archiving policies is to implement a mandatory checklist linked to instructions and examples during submission (or at resubmission) that ensures authors include essential features for their data and code. These should use existing guidelines, for example FAIR, FAIR4RS, TADA!, European Open Science Cloud's Research Software MetaData (RSMD), STREAMS for microbiome data (Barker et al., 2022; Gruenpeter et al., 2024; Ivimey-Cook, Culina, et al., 2025; Kelliher et al., 2025; Wilkinson et al., 2016) to avoid 'reinventing the wheel'. *MEE*, however, (along with many other journals) already implements checklists and our results show this has not fixed all issues. We would, therefore, strongly encourage more journals to employ Data/Code Editors (Pick et al., 2026), to check data and code before publication. This would help to identify issues like incomplete data archives, poor documentation, missing metadata etc. Importantly, these editors should focus on implementing 'good' rather than 'perfect' practice, with respect to code-archiving in particular, following guidance in Pick et al. (2026). We want to encourage and empower people to share their code, not overwhelm them with overly complex technical solutions.

We also encourage learned societies and institutions to improve training for data- and code-archiving (Kohrs et al., 2023), particularly for early career researchers but also for senior researchers who struggle to keep up with current best practice in the area but are responsible for setting the standards and expectations for their research groups. Crucially, research funders should set clear expectations for data- and code-archiving practices, and explicitly encourage budgeting for data and research software engineering labour in grant applications. Good data- and code-archiving practices require dedicated, highly skilled labour that is currently neither appropriately recognised nor systematically funded.

Finally, we emphasise that effective data- and code-archiving is only the first step towards ensuring analyses are reproducible. Studies suggest that much code in ecology is not functional, making analyses impossible to reproduce (Archmiller et al., 2020; Kellner et al., 2025). We encourage researchers regularly writing code to make use of resources to help them improve their programming skills (Barker et al., 2022; Cooper & Hsing, 2025; Ivimey-Cook, Culina, et al., 2025) and to improve the functionality of their code.

5 | CONCLUSION

Mandatory data-archiving policies at the BES journals appear largely effective at ensuring data is archived and accessible, at least in the sample we analysed. The harder problem, which policies alone have not solved, was that while most papers met minimum data-archiving requirements, few authors engaged with practices that would make the data/code reusable. In addition, code-archiving remains uncommon even in journals with explicit requirements. Closing these gaps will require better documentation by authors, active quality checks through dedicated Data/Code Editors, institutional training and funding structures that recognise data and code stewardship as skilled labour. It is also important to inform authors about the benefits of data/code-archiving not only for the broader community but for their own work and career prospects, so they see this as a worthwhile investment rather than yet another hurdle to progression. The collaborative hackathon approach we used here could itself be a model other journal groups could adopt to monitor whether their policies are working as intended; and repeating such assessments at intervals would help the community track where further intervention is needed.

AUTHOR CONTRIBUTIONS

Natalie Cooper: Conceptualisation, data curation, formal analysis, investigation, methodology, project administration, visualisation, writing—original draft. **BES Data and Code Hackathon Group:** conceptualisation, data curation, investigation, methodology, writing—review and editing. All authors approved the final version for submission.

ACKNOWLEDGEMENTS

Thanks to the attendees of our BES Data and Code Hackathon and to various contributors on social media and members of SORTEE, for discussion and suggestions which informed this perspective. We also thank Riva Quiroga (http://orcid.org/0000-0002-1147-4135) and Esther Plomp (https://orcid.org/0000-0003-3625-1357) of The Turing Way community for their feedback on the questions in the online form.

FUNDING INFORMATION

Methods in Ecology and Evolution funded catering at the in-person hackathon.

CONFLICT OF INTEREST STATEMENT

Natalie Cooper, Hooman Latifi, Nicolas Lecomte, Jennifer Meyer and Harriet Rhodes are compensated by the BES for their work at Methods in Ecology and Evolution. Res Altwegg, Edward Codling, Laura Graham, Pen-Yuan Hsing, Graziella Iossa, Sydne Record, Fernando. Taboada, Saras M. Windecker are (unpaid) editors at Methods in Ecology and Evolution. None of these editors were involved in the handling of this manuscript.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210x.70338.

DATA AVAILABILITY STATEMENT

Anonymised data, and all code to reproduce the analyses, are available on Zenodo at https://doi.org/10.5281/zenodo.19737911 (Cooper & BES Data and Code Hackathon Group, 2026) and on GitHub under the MIT open source licence: https://github.com/nhcooper123/reproduce-reuse-recycle. A Shiny app allowing users to view the data is available here: https://www.lewisajones.com/shiny-bes-hackathon/.

ORCID

Natalie Cooper https://orcid.org/0000-0003-4919-8655

REFERENCES

Allen, C., & Mehler, D. M. A. (2019). Open science challenges, benefits and tips in early career and beyond. *PLoS Biology*, *17*(5), e3000246. https://doi.org/10.1371/journal.pbio.3000246

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., & Fieberg, J. (2020). Computational reproducibility in the wildlife Society's flagship journals. *Journal of Wildlife Management*, *84*, 1012–1017. https://doi.org/10.1002/jwmg.21855

Barker, M., Chue Hong, N. P., Katz, D. S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., & Honeyman, T. (2022). Introducing the FAIR principles for research software. *Scientific Data*, *9*(1), 622. https://doi.org/10.1038/s41597-022-01710-x

Berberi, I., & Roche, D. G. (2023). Reply to: Recognizing and marshalling the pre-publication error correction potential of open data for more reproducible science. *Nature Ecology & Evolution*, *7*(10), 1595–1596. https://doi.org/10.1038/s41559-023-02142-5

Carroll, S. R., Garba, I., Figueroa-Rodríguez, O. L., Holbrook, J., Lovett, R., Materechera, S., Parsons, M., Raseroka, K., Rodriguez-Lonebear, D., Rowe, R., Sara, R., Walker, J. D., Anderson, J., & Hudson, M. (2020). The CARE principles for indigenous data governance. *Data Science Journal*, *19*, 1–12. https://doi.org/10.5334/dsj-2020-043

Colavizza, G., Cadwallader, L., LaFlamme, M., Dozot, G., Lecorney, S., Rappo, D., & Hrynaszkiewicz, I. (2024). An analysis of the effects of sharing research data, code, and preprints on citations. *PLoS One*, *19*(10), e0311493. https://doi.org/10.1371/journal.pone.0311493

Cooper, N., & BES Data and Code Hackathon Group. (2026). Data and code for MEE paper v1.0. *Zenodo*. https://doi.org/10.5281/zenodo.19737911

Cooper, N., & Hsing, P.-Y. (2025). *Guide to reproducible code*. British Ecological Society. https://doi.org/10.5281/ZENODO.16421733

Culina, A., van den Berg, I., Evans, S., & Sánchez-Tójar, A. (2020). Low availability of code in ecology: A call for urgent action. *PLoS Biology*, *18*(7), e3000763. https://doi.org/10.1371/journal.pbio.3000763

Di Cosmo, R., & Zacchiroli, S. (2017). *Software heritage: Why and how to preserve software source code*. iPRES 2017—14th International Conference on Digital Preservation. https://hal.science/hal-01590958

Evans, S. R. (2016). Gauging the purported costs of public data archiving for long-term population studies. *PLoS Biology*, *14*(4), e1002432. https://doi.org/10.1371/journal.pbio.1002432

Fernández-Juricic, E. (2021). Why sharing data and code during peer review can enhance behavioral ecology research. *Behavioral Ecology and Sociobiology*, *75*(7), 1–5. https://doi.org/10.1007/s00265-021-03036-x

Fidler, F., Chee, Y. E., Wintle, B. C., Burgman, M. A., McCarthy, M. A., & Gordon, A. (2017). Metaresearch for evaluating reproducibility in ecology and evolution. *Bioscience*, *67*(3), 282–289. https://doi.org/10.1093/biosci/biw159

Firke, S. (2016). Janitor: Simple tools for examining and cleaning dirty data [dataset]. In *CRAN: Contributed packages*. The R Foundation. https://doi.org/10.32614/cran.package.janitor

Goldacre, B., Morton, C. E., & DeVito, N. J. (2019). Why researchers should share their analytic code. *BMJ (Clinical Research Ed.)*, *367*, l6365. https://doi.org/10.1136/bmj.l6365

Gomes, D. G. E., Pottier, P., Crystal-Ornelas, R., Hudgins, E. J., Foroughirad, V., Sánchez-Reyes, L. L., Turba, R., Martinez, P. A., Moreau, D., Bertram, M. G., Smout, C. A., & Gaynor, K. M. (2022). Why don't we share data and code? Perceived barriers and benefits to public archiving practices. *Proceedings of the Royal Society B: Biological Sciences*, *289*(1987), 20221113. https://doi.org/10.1098/rspb.2022.1113

Gruenpeter, M., Granger, S., Monteil, A., Chue Hong, N., Breitmoser, E., Antonioletti, M., Garijo, D., González Guardia, E., Gonzalez Beltran, A., Goble, C., Soiland-Reyes, S., Juty, N., & Mejias, G. (2024). D4.4—Guidelines for recommended metadata standard for research software within EOSC. *Zenodo*. https://doi.org/10.5281/ZENODO.10786147

Ivimey-Cook, E. R., Culina, A., Dimri, S., Grainger, M., Kar, F., Lagisz, M., Moran, N. P., Nakagawa, S., Roche, D. G., Sanchez-Tojar, A., Windecker, S. M., & Pick, J. L. (2025). *TADA! Simple guidelines to improve code sharing*. https://ecoevorxiv.org/repository/view/9806/

Ivimey-Cook, E. R., Pick, J. L., Bairos-Novak, K. R., Culina, A., Gould, E., Grainger, M., Marshall, B. M., Moreau, D., Paquet, M., Royauté, R., Sánchez-Tójar, A., Silva, I., & Windecker, S. M. (2023). Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology. *Journal of Evolutionary Biology*, *36*(10), 1347–1356. https://doi.org/10.1111/jeb.14230

Ivimey-Cook, E. R., Sánchez-Tójar, A., Berberi, I., Culina, A., Roche, D. G., A Almeida, R., Amin, B., Bairos-Novak, K. R., Balti, H., Bertram, M. G., Bliard, L., Byrne, I., Chan, Y.-C., Cioffi, W. R., Corbel, Q., Elsy, A. D., Florko, K. R. N., Gould, E., Grainger, M. J., … Moran, N. P. (2025). From policy to practice: Progress towards data- and code-sharing in ecology

and evolution. *Proceedings of the Royal Society B: Biological Sciences*, *292*(2055), 20251394. https://doi.org/10.1098/rspb.2025.1394

Jones, L. A., Dean, C. D., Allen, B. J., Drage, H. B., Flannery-Sutherland, J. T., Gearty, W., Chiarenza, A. A., Dillon, E. M., Farina, B. M., & Godoy, P. L. (2025). Ten simple rules to follow when cleaning occurrence data in palaeobiology. *Palaeontology*, *68*(5), e70028. https://doi.org/10.1111/pala.70028

Kambouris, S., Wilkinson, D. P., Smith, E. T., & Fidler, F. (2024). Computationally reproducing results from meta-analyses in ecology and evolutionary biology using shared code and data. *PLoS One*, *19*(3), e0300333. https://doi.org/10.1371/journal.pone.0300333

Kelliher, J. M., Mirzayi, C., Bordenstein, S. R., Oliver, A., Kellogg, C. A., Hatcher, E. L., Berg, M., Baldrian, P., Aljumaah, M., Miller, C. M. L., Mungall, C., Novak, V., Palucki, A., Smith, E., Tabassum, N., Bonito, G., Brister, J. R., Chain, P. S. G., Chen, M., … Eloe-Fadrosh, E. A. (2025). STREAMS guidelines: Standards for technical reporting in environmental and host-associated microbiome studies. *Nature Microbiology*, *10*(12), 3059–3068. https://doi.org/10.1038/s41564-025-02186-2

Kellner, K. F., Doser, J. W., & Belant, J. L. (2025). Functional R code is rare in species distribution and abundance papers. *Ecology*, *106*(1), e4475. https://doi.org/10.1002/ecy.4475

Kimmel, K., Avolio, M. L., & Ferraro, P. J. (2023). Empirical evidence of widespread exaggeration bias and selective reporting in ecology. *Nature Ecology & Evolution*, *7*(9), 1525–1536. https://doi.org/10.1038/s41559-023-02144-3

Kohrs, F. E., Auer, S., Bannach-Brown, A., Fiedler, S., Haven, T. L., Heise, V., Holman, C., Azevedo, F., Bernard, R., Bleier, A., Bössel, N., Cahill, B. P., Castro, L. J., Ehrenhofer, A., Eichel, K., Frank, M., Frick, C., Friese, M., Gärtner, A., … Weissgerber, T. L. (2023). Eleven strategies for making reproducible research and open science training the norm at research institutions. *eLife*, *12*, e89736. https://doi.org/10.7554/eLife.89736

Maitner, B., Santos Andrade, P. E., Lei, L., Kass, J., Owens, H. L., Barbosa, G. C. G., Boyle, B., Castorena, M., Enquist, B. J., Feng, X., Park, D. S., Paz, A., Pinilla-Buitrago, G., Merow, C., & Wilson, A. (2024). Code sharing in ecology and evolution increases citation rates but remains uncommon. *Ecology and Evolution*, *14*(8), e70030. https://doi.org/10.1002/ece3.70030

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., & Yarkoni, T. (2016). How open science helps researchers succeed. *eLife*, *5*, e16800. https://doi.org/10.7554/eLife.16800

Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, *2014*(239), 2. https://doi.org/10.5555/2600239.2600241

Mislan, K. A. S., Heer, J. M., & White, E. P. (2016). Elevating the status of code in ecology. *Trends in Ecology & Evolution*, *31*(1), 4–7. https://doi.org/10.1016/j.tree.2015.11.006

Noble, D. W. A., Xirocostas, Z. A., Wu, N. C., Martinig, A. R., Almeida, R. A., Bairos-Novak, K. R., Balti, H., Bertram, M. G., Bliard, L., Brand, J. A., Byrne, I., Chan, Y.-C., Clink, D. J., Corbel, Q., Correia, R. A., Crawford-Ash, J., Culina, A., D'Bastiani, E., Deme, G. G., … Lagisz, M. (2025). The promise of community-driven preprints in ecology and evolution. *Proceedings of the Royal Society B: Biological Sciences*, *292*(2039), 20241487. https://doi.org/10.1098/rspb.2024.1487

O'Dea, R. E., Parker, T. H., Chee, Y. E., Culina, A., Drobniak, S. M., Duncan, D. H., Fidler, F., Gould, E., Ihle, M., Kelly, C. D., Lagisz, M., Roche, D. G., Sánchez-Tójar, A., Wilkinson, D. P., Wintle, B. C., & Nakagawa, S. (2021). Towards open, reliable, and transparent ecology and evolutionary biology. *BMC Biology*, *19*(1), 68. https://doi.org/10.1186/s12915-021-01006-3

Pedersen, T. L. (2025). Patchwork: The composer of plots [dataset]. In *CRAN: Contributed packages*. The R Foundation. https://doi.org/10.32614/cran.package.patchwork

Pick, J., Allen, B., Bachelot, B., Bairos-Novak, K., Brand, J., Class, B., Dallas, T., D'Amelio, P., Fenollosa, E., Fernández-Juricic, E., Gomes, D., Grainger, M., Guillemaud, T., John, C., Krasnow, R., Lagisz, M., Lequime, S., Maynard, D., Nakagawa, S., … Ivimey-Cook, E. R. (2026). The SORTEE guidelines for data and code quality control in ecology and evolutionary biology. *Peer Community Journal*, *6*, e20. https://doi.org/10.24072/pcjournal.687

Piwowar, H. A., Day, R. S., & Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. *PLoS One*, *2*(3), e308. https://doi.org/10.1371/journal.pone.0000308

Poisot, T., Bruneau, A., Gonzalez, A., Gravel, D., & Peres-Neto, P. (2019). Ecological data should not be so hard to find and reuse. *Trends in Ecology & Evolution*, *34*(6), 494–496. https://doi.org/10.1016/j.tree.2019.04.005

Poisot, T., Mounce, R., & Gravel, D. (2013). Moving toward a sustainable ecological science: Don't let data go to waste! *Ideas in Ecology and Evolution*, *6*(2), 11–19. https://doi.org/10.4033/iee.2013.6b.14.f

Powers, S. M., & Hampton, S. E. (2019). Open science, reproducibility, and transparency in ecology. *Ecological Applications*, *29*(1), e01822. https://doi.org/10.1002/eap.1822

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Roche, D. G., Berberi, I., Dhane, F., Lauzon, F., Soeharjono, S., Dakin, R., & Binning, S. A. (2022). Slow improvement to the archiving quality of open datasets shared by researchers in ecology and evolution. *Proceedings of the Royal Society B: Biological Sciences*, *289*(1975), 20212780. https://doi.org/10.1098/rspb.2021.2780

Roche, D. G., Kruuk, L. E. B., Lanfear, R., & Binning, S. A. (2015). Public data archiving in ecology and evolution: How well are we doing? *PLoS Biology*, *13*(11), e1002295. https://doi.org/10.1371/journal.pbio.1002295

Sánchez-Tójar, A., Bezine, A., Purgar, M., & Culina, A. (2025). Code-sharing policies are associated with increased reproducibility potential of ecological findings. *Peer Community Journal*, *5*, e37. https://doi.org/10.24072/pcjournal.541

Sholler, D., Ram, K., Boettiger, C., & Katz, D. S. (2019). Enforcing public data archiving policies in academic publishing: A study of ecology journals. *Big Data & Society*, *6*(1), 205395171983625. https://doi.org/10.1177/2053951719836258

Soeharjono, S., & Roche, D. G. (2021). Reported individual costs and benefits of sharing open data among Canadian academic faculty in ecology and evolution. *Bioscience*, *71*(7), 750–756. https://doi.org/10.1093/biosci/biab024

Tierney, N., & Cook, D. (2023). Expanding tidy data principles to facilitate missing data exploration, visualization and assessment of imputations. *Journal of Statistical Software*, *105*(7), 1–31. https://doi.org/10.18637/jss.v105.i07

UNESCO. (2021). *UNESCO recommendation on Open Science*. UNESCO. https://doi.org/10.54677/mnmh8546

Ushey, K., & Wickham, H. (2026). *renv: Project Environments*. R package version 1.1.7.9000. https://github.com/rstudio/renv

Vandewalle, P. (2012). Code sharing is associated with research impact in image processing. *Computing in Science & Engineering*, *14*(4), 42–47. https://doi.org/10.1109/MCSE.2012.63

Vines, T. H., Andrew, R. L., Bock, D. G., Franklin, M. T., Gilbert, K. J., Kane, N. C., Moore, J.-S., Moyers, B. T., Renaut, S., Rennison, D. J., Veen, T., & Yeaman, S. (2013). Mandated data archiving greatly improves access to research data. *FASEB Journal*, *27*(4), 1304–1308. https://doi.org/10.1096/fj.12-218164

Whitlock, M. C. (2011). Data archiving in ecology and evolution: Best practices. *Trends in Ecology & Evolution*, *26*(2), 61–65. https://doi.org/10.1016/j.tree.2010.11.006

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D.,

Seidel, D., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *The Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*, 160018. https://doi.org/10.1038/sdata.2016.18

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Supporting Information A: Hackathon Protocol.

Supporting Information B:

Figure S1. Numbers of papers published each year in each of the seven BES journals.

Figure S2. Data-archiving variables across the seven BES journals.

Figure S3. Code-archiving variables across the seven BES journals.

Figure S4. Archive location for archived data across the seven BES journals.

Figure S5. Archive location for archived code across the seven BES journals.

Figure S6. File extensions for archived data across the seven BES journals.

Figure S7. File extensions for archived code across the seven BES journals.

Figure S8. README quality for archived data across the seven BES journals.

Figure S9. README quality for archived code across the seven BES journals.

Figure S10. Data completeness for archived data across the seven BES journals.

Figure S11. Code annotation quality for archived code across the seven BES journals.

Figure S12. Licences for archived data across the seven BES journals.

Figure S13. Licences for archived code across the seven BES journals.

Figure S14. Programming languages of code across the seven BES journals.

Figure S15. Recorder variability.

Figure S16. Numbers of unique code files written in each programming language.

Table S1. Numbers of papers published in the seven BES journals between 2017 and 2024.

Table S2. Summary results for data and code across all seven journals combined.

Table S3. Summary results for data divided by journal.

Table S4. Summary results for code divided by journal.

BES Data and Code Hackathon Group author list.

How to cite this article: Cooper, N., & (2026). Data- and code-archiving in the British Ecological Society journals: Present status and recommendations for future improvements. *Methods in Ecology and Evolution*, *00*, 1–13. https://doi.org/10.1111/2041-210x.70338