

1 The SORTEE Guidelines for Data and Code Quality

2 Control in Ecology and Evolutionary Biology

3 Joel L. Pick^{1*}, Bethany J. Allen², Benedicte Bachelot³, Kevin R. Bairos-Novak⁴, Jack A. Brand^{5,6},
4 Barbara Class⁷, Tad Dallas⁸, Pietro B. D'Amelio⁹, Erola Fenollosa¹⁰, Esteban Fernández-
5 Juricic¹¹, Dylan G. E. Gomes¹², Matthew J. Grainger¹³, Thomas Guillemaud¹⁴, Christian John¹⁵,
6 Ruby Krasnow¹⁶, Malgorzata Lagisz^{17,18}, Sebastian Lequime¹⁹, Daniel S. Maynard²⁰, Shinichi
7 Nakagawa¹⁸, Rose E. O'Dea²¹, Matthieu Paquet²², Quentin Petitjean²³, Alfredo Sánchez-Tójar²⁴,
8 Natalie E. van Dis^{25,26}, Laura A. B. Wilson^{17,27}, Edward R. Ivimey Cook^{28*}

9

10 *Corresponding authors: joel.l.pick@gmail.com and e.ivimeycook@googlemail.com

11

12 ¹Institute of Ecology and Evolution, University of Edinburgh, Edinburgh UK

13 ²GFZ Helmholtz Centre for Geosciences, Potsdam, Germany

14 ³Oklahoma State University, OK, USA

15 ⁴Australian Institute of Marine Science, PMB 3, Townsville MC, QLD, 4810, Australia

16 ⁵Department of Wildlife, Fish, and Environmental Studies, Swedish University of Agricultural
17 Sciences, Umeå 907 36, Sweden

18 ⁶Institute of Zoology, Zoological Society of London, London NW1 4RY, UK

19 ⁷Direction pour la Science Ouverte (DipSO), INRAE, France

20 ⁸University of South Carolina, SC, USA

21 ⁹Department of Biology, Reed College, Portland, Oregon, 97202, USA

22 ¹⁰Department of Biology, University of Oxford, UK

23 ¹¹Purdue University, USA

24 ¹²Marine Reserves, Oregon Department of Fish and Wildlife, Newport, OR, 97365, USA

25 ¹³Knowledge Synthesis Department, Norwegian Institute for Nature Research (NINA),
26 Trondheim, Norway

27 ¹⁴Isa, Université Côte d'Azur, INRAE, Sophia-Antipolis, France

28 ¹⁵Marine Science Institute, University of California, Santa Barbara. Santa Barbara, CA 93106
29 USA

30 ¹⁶University of Maine, School of Marine Sciences, Orono, ME, USA

31 ¹⁷Evolution & Ecology Research Centre, School of Biological, Earth & Environmental Sciences,
32 University of New South Wales, Kensington, NSW, 2052, Australia
33 ¹⁸Department of Biological Sciences, University of Alberta, Edmonton, AB T6G 2E9, Canada
34 ¹⁹Cluster of Microbial Ecology, Groningen Institute for Evolutionary Life Sciences, University of
35 Groningen, Groningen, The Netherlands
36 ²⁰Department of Genetics, Evolution, and Environment, University College London, London, UK
37 ²¹School of Agriculture, Food and Ecosystem Sciences, University of Melbourne
38 ²²SETE, Station d'Écologie Théorique et Expérimentale, CNRS, Moulis, France
39 ²³Abeilles et Environnement (UR406), Institut National de Recherche pour l'Agriculture,
40 l'Alimentation et l'Environnement (INRAE), Avignon, France.
41 ²⁴Department of Evolutionary Biology, Bielefeld University, Germany
42 ²⁵Organismal and Evolutionary Biology, University of Helsinki, P.O. Box 4, 00014 Helsinki,
43 Finland
44 ²⁶Department of Animal Ecology, Netherlands Institute of Ecology (NIOO-KNAW), PO Box 50,
45 6700 AB Wageningen, The Netherlands
46 ²⁷School of Archaeology and Anthropology, The Australian National University, Acton ACT
47 2601, Australia
48 ²⁸University of East Anglia, Norwich, UK
49

50 Abstract

51 Open data and code are crucial to increasing transparency and reproducibility, and in building
52 trust in scientific research. However, despite an increasing number of journals in ecology and
53 evolutionary biology mandating for data and code to be archived alongside published articles,
54 the amount and quality of archived data and code, and subsequent reproducibility of results, has
55 remained worryingly low. As a result, a handful of journals have recruited dedicated data
56 editors, whose role is to help authors increase the overall quality of archived data and code.
57 There is, however, a general lack of consensus around what a data editor should check, how to
58 do it, and to what level of detail, and the process is often vague and hidden from readers and
59 authors alike. Here, with the input from multiple data editors across several journals in ecology
60 and evolutionary biology, we establish and describe the first standardised guidelines for Data
61 and Code Quality Control on behalf of the Society for Open, Reliable, and Transparent Ecology
62 and Evolutionary Biology (SORTEE). We then introduce the SORTEE-led guidelines as a
63 flexible six-stage framework that journals can implement incrementally and/or apply on a case-
64 by-case basis, particularly when some checks (e.g., computational reproducibility) are not
65 feasible (e.g., proprietary software). We conclude with practical advice for journals and authors,
66 arguing that flexible adoption of these standardised guidelines will improve the consistency and
67 transparency of the data editor process for readers, authors, data editors, and the wider
68 scientific community.

69

70 **Keywords:** Data sharing, Code Sharing, Computational Reproducibility, Open Science, Data
71 Re-use, Methodological Rigor, FAIR principles, Transparency, Data Editor

72 Introduction

73 A major focus of open science efforts in the last two decades, especially in ecology and
74 evolutionary biology, has been open data, and more recently, open code. Open data and open
75 code refer to the public archiving of the data and code associated with published research. The
76 goals and societal benefits of data and code archiving have been widely discussed (e.g., Parr &
77 Cummings 2005, Barnes 2010, Molloy 2011, Wilkinson *et al* 2016, Goldacre 2019, Gomes *et al*
78 2022, Ivimey-Cook *et al* 2023; see Box 1). Despite some reticence about open data and code
79 (see Gomes *et al* 2022 for an overview of these fears), previous work has shown that data
80 archiving is supported by the majority of academics in ecology and evolutionary biology, who
81 perceive that the benefits outweigh any costs (Soeharjono & Roche 2021). Indeed, the two most
82 important issues to the more than 1,000 members of the Society for Open, Reliable, and
83 Transparent Ecology and Evolutionary Biology (SORTEE) from 2021-2025 have consistently
84 been open data and open code (SORTEE 2026).

BOX 1: Goals of data and code archiving

From an idealistic perspective, data and code archiving has three main goals: to allow data reuse, to increase transparency, and to provide computational reproducibility (components that have been highlighted previously, e.g., Wilkinson *et al* 2016). We cover each of these in turn below.

A. Allow data reuse

The main focus of data archiving in the past has been on ensuring the potential for data reuse. Most prominently, this has included the development of the FAIR principles (Findable, Accessible, Interoperable, Reusable; Wilkinson *et al* 2016). There are several key motivations for this. First, data archiving prevents data loss. Data is typically collected using public money, and so can be seen as a public good that should not be lost when a researcher leaves their job or when a computer is lost/broken. Loss of data (and code) represents a massive source of research waste (Purgar *et al.* 2022). Second, archived data better allows for research synthesis (Culina *et al.* 2018, Hennessy *et al* 2022). Meta-analysis plays a key role in generalising results across systems, however, published

results are often not described in enough detail to be included in a meta-analysis. Provision of the data allows re-analysis to gain the information a meta-analyst needs and can substantially increase the data available for synthesis (Kim *et al* 2021). Third, methods are frequently updated, and data archiving allows for data to be re-analysed when new methods become available. Finally, data archiving allows for new questions to be asked with existing data. This is not only an efficient use of time and money, but reduces the use of animals in experiments, a central tenet of animal ethics (Janssens *et al* 2023).

To achieve the goal of allowing data reuse, it is essential that data and their accompanying metadata (see Table 1) are both present and in a form that allows for re-use. However, archived data are often incomplete or not in a state where they can be reused (Roche *et al* 2015, Roche *et al* 2022). Helping authors to ensure their archived data are FAIR should therefore be a key goal for data editors.

B. Increase transparency

Another key goal of data and code archiving is to increase transparency. Transparency involves making the research process visible, and is associated with building trust and credibility in science (Vazire 2017). While trust in scientists remains high globally, even a small minority's distrust can shape how research findings are received by the public and decision-makers (Cologna *et al* 2025), reinforcing the need for researchers to actively build and strengthen trust both with other academics and with the general public.

Analyses in ecology and evolutionary biology are becoming increasingly complex (Touchon & McCoy 2016, Feng *et al.*, 2020), and there are often several ways to perform an analysis, with varying outcomes (Gould *et al* 2025). Descriptions of data filtering, processing, and analysis included within articles are not always sufficient to fully reproduce analyses (Archmiller *et al* 2020; Minocher *et al.* 2021; see Table 1). Provision of analysis code alongside a manuscript would therefore allow the analytical methods used to be directly assessed by reviewers when provided during peer review (Fernández-Juricic 2021) and by the general readership upon publication. Transparent methods allow work to be built on more easily, making science more efficient. Unlike data, the goal of code archiving in empirical articles is rarely direct code reuse (if this is the goal, then it is often more appropriate to create software packages, to allow generalisation of a method). Code archiving instead allows software code to be used for reference, adapted for new use, to allow similar methods to be applied to new datasets, or for methods to be extended.

Data and code archiving also allow mistakes to be found. Coding mistakes are easily made, and whilst many may have negligible effects on the results of an article, some will have major effects (Gihawi *et al* 2023; Mandhane 2024). The availability of the data and code that support an article make it possible for these mistakes to be found, and importantly, corrected in the future (Bolnick & Paull 2016, Delgado Manzanedo *et al* 2021). Finally, following several high profile cases of academic fraud (e.g., Viglione *et al* 2020, <https://retractionwatch.com/2017/05/01/remarkable-ever-accepted-says-report-science-retract-study-fish-microplastics/>, <https://retractionwatch.com/2022/08/09/science-retracts-ocean-acidification-paper-more-than-a-year-after-a-report-on-allegations-in-its-own-pages/>), it has become increasingly clear that, as a community, we would benefit from a higher degree of transparency in how the results of published articles are generated. Although the provision of data and code will not stop data fabrication, publicly available data makes detecting data fabrication much easier as the data can be scrutinised and presents an additional hurdle to generating fraudulent results. As data and code are increasingly provided alongside journal articles in our field (although more so for data than code; Culina *et al* 2020; Kimmel *et al* 2023; Sánchez-Tójar *et al* 2025), not providing these resources leads readers to ask ‘What are the reasons why the authors did not want to share their data and code?’ (see Gomez *et al*, 2023).

To fulfil the goal of transparency, readers therefore need to see what has been done to obtain the reported results. This requires the presence of all data and code needed to reproduce the results presented in an article, as well as linking what was done in that article with the structure and form of both the data and code (using metadata, and appropriate code annotation).

C. Provide computational reproducibility

Perhaps the most ambitious goal of data and code archiving is computational reproducibility, which ultimately builds trust and credibility in published results (Power and Hampton 2018, Reinecke *et al* 2022). We can define computational reproducibility as ‘*obtaining consistent results using the same input data, computational methods, and conditions of analysis*’ (National Academies of Sciences, Engineering, and Medicine 2019). Although this definition of computational reproducibility is commonly cited, the terms within it are not clearly defined. In the context of a research article, we interpret this to mean that, given the available data (i.e. input data) and code or workflow (i.e. computational methods), and using the same software versions (and hardware if appropriate) outlined in the article and metadata (i.e. conditions of analysis), we should be able to reproduce the results presented in the article.

In some cases, exact reproducibility is difficult to achieve (i.e. generating the *exact* numbers presented in an article), and often computational reproducibility is assessed with some tolerance level (e.g., Archmiller *et al* 2020; Kambouris *et al* 2024). However, practices such as having appropriate metadata that adequately describes the software and package versions, and setting of seeds (where the same pseudorandom numbers are generated each time) within code for stochastic methods, will help to achieve these goals. We should note that it is unlikely we will be able to demonstrate computational reproducibility in cases where analyses are highly computationally intensive, but solutions exist. We discuss these points in more detail in the main text.

To achieve the goal of computational reproducibility, as a minimum requirement, we need all code and data to be present. The next component of computational reproducibility is that the provided code runs without error. This requires the code to be explicit about what data files it uses and where these are located, it requires data files to be in the same directory structure and with the same names as expected by the code, and it requires the same version of all software packages used in the code to be loaded. Any code that cannot be rerun without error in a clean workspace cannot be considered computationally reproducible.

85

86 In response to this call for open data, starting in 2010, many journals in ecology and
87 evolutionary biology began to mandate data archiving, e.g., Journal of Animal Ecology,
88 Functional Ecology, and Heredity, to name a few (for a full list see the Joint Data Archiving
89 Policy; <https://doi.org/10.25504/FAIRsharing.z67ht2>). As a result, an increasing number of
90 journals in ecology and evolutionary biology have mandatory open data policies (estimated to
91 be 20% out of 196 journals in 2020, 35% in 2023; Berberi and Roche 2022, Berberi and Roche
92 2023 and 41% in 2024; Ivimey-Cook *et al* 2025). This action has resulted in a large increase in
93 the number of publications in ecology and evolutionary biology having open data (Vines *et al.*
94 2013; Culina *et al* 2020 found 79% in 14 journals that have a code archiving policy with no
95 change from 2015/16 to 2018/19; Sanchez-Tojar *et al.* 2025 found 37% in 12 journals without
96 code archiving policies with an increase over time; Kimmel *et al* 2023 found 78.5% in 5 journals

97 from 2018-20; Belkhir *et al.* 2025 49% in 110 journals in 2024). Compared to many other fields,
98 ecology and evolutionary biology are at the forefront of data sharing (Tedersoo *et al.* 2021).
99
100 However, despite a high proportion of ecology and evolutionary biology studies archiving data,
101 archived data are often of low quality (Roche *et al* 2015, 2022), with most datasets either
102 incomplete (some or all of the data allowing the study to be reproduced is not present) or
103 unusable (e.g., data are not machine readable, in a proprietary format, or are archived with no
104 metadata; see Table 1). Based on 362 open datasets from 2013-2019, Roche *et al* (2022)
105 calculated that 56.4% of datasets were complete, and 45.9% were reusable (out of 362), a
106 situation that has only marginally improved over the last decade (from a sample of 100 articles
107 in 2012/13, 44% were complete, and 36% reusable; Roche *et al* 2015), with only reusability
108 having statistically increased from 2013 to 2019 (Roche *et al* 2022). Several studies have
109 further sought to directly assess analytical reproducibility (defined as reproducing the published
110 results using the same data). However, these assessments rely heavily on data provided by
111 authors upon request (Archmiller *et al* 2020; Minocher *et al* 2021), as rates of archived data
112 recovery were low (11% in Minocher *et al.* 2021). Conditional on having the full dataset,
113 reproducibility was moderate (42% and 58% of articles were fully reproducible in Archmiller *et al*
114 2020 and Minocher *et al* 2021, respectively), but whether the quality of data provided directly
115 from authors for these studies differs from data that has been archived is not clear. This is
116 similar to other fields; in the *Journal of Psychological Science*, only 9 out of 25 articles were
117 reproducible (given the data) without author intervention (2014/2015; Hardwicke *et al* 2021) and
118 even when journals have mandated data archiving only 62% (85/136) of datasets were reusable
119 in the journal *Cognition* (2015/2016; Hardwicke *et al*, 2018). Overall, these results suggest that
120 most studies across fields either do not have archived data or provide a dataset that has limited
121 utility, but when full datasets are provided, reproducibility can be high. The lack of high quality
122 data impedes all the goals of data archiving (see Box 1).

123

124 The use of code for data preparation and analysis is now almost ubiquitous, particularly using
125 the R coding language (Lai *et al* 2019, R Core Team 2022). Increasingly, journals encourage or
126 mandate code archiving (15% in 2015 Mislán *et al* 2015; 75% in 2020 Culina *et al* 2020; 88.4%
127 in 2024 Ivimey-Cook *et al* 2025). However, the actual rates of code archiving still remain low
128 (2015-2016: 2.5%, 2018-2019: 7.0% in journals without a code archiving policy; 23% in 2015/16
129 to 30% in 2018/19 in journals that encourage or mandate, Culina *et al* 2020; 18% 2018-2020 in
130 5 major ecology journals, Kimmel *et al* 2023). Several recent studies have further tried to assess
131 computational reproducibility (the ability to reproduce the results given the archived data and
132 code; see Box 1), but have concluded that it is likely to be low in ecology and evolutionary
133 biology. Kambouris *et al* (2024) found that out of 177 meta-analyses in ecology and evolutionary
134 biology from 2015-17, only 26 provided both data and code. From these, only 7 studies (27%)
135 could be exactly reproduced (with the results of 15 (58%) studies being reproduced to within
136 10% of the original results). Kellner *et al* (2025) found 7% of 497 articles on species distribution
137 and abundance from 2018-2022 had code that ran. Trisovic *et al* (2022) found that out of 9,000
138 unique R files from the Harvard Dataverse, 74% failed to complete, which lowered to 56% when
139 basic cleaning was applied. The lack of high quality code impedes all the goals of transparency
140 and computational reproducibility (see Box 1). Together, the lack of functional data *and* code in
141 public repositories limits the verifiability of published empirical research claims (Henderson *et al*
142 2024) and ultimately erodes trust in science.

143

144 Alongside several high profile fraud scandals (see Box 1), the lack of adherence to journal
145 archiving policies, and the low quality of data and code archiving has led several journals to
146 recruit data editors (<https://www.amnat.org/announcements/data-and-code-announcement.html>,
147 Thrall *et al.* 2023, Barrett, 2024, Barrett & Montgomerie 2025). Data editors are responsible for
148 screening and assessing the archived data and code of manuscripts being reviewed by a

149 journal, to assist authors in complying with journal mandates on data and code provision -
150 hereafter, we refer to this process as *data and code quality control*. It is worth stressing that
151 data editors are not acting as gatekeepers; the role of a data editor is to help authors adhere to
152 community standards of data and code archiving. At the time of writing, we are aware of seven
153 journals in ecology and evolutionary biology that have data editors that screen the data and
154 code of some or all manuscripts that are published (American Naturalist, Behavioral Ecology,
155 Ecology Letters, Ethology, Journal of Evolutionary Biology, Proceedings of the Royal Society B,
156 and Peer Community Journal). Behavioural Ecology and Sociobiology has an editor with a
157 related role, but only screens a small number of manuscripts at the request of other editors, and
158 additionally provides statistical support (Fernández-Juricic *pers. comms.*).

159

160 Data and code quality control by data editors is primarily for the benefit of the authors. A large
161 part of the data editors' role is to help authors ensure their data and code more closely adhere
162 to the open principles that we have adopted as a research community. At the end of the
163 process, authors will therefore have higher quality archived data and code for each publication,
164 which has been linked to increased citation rates (Piwowar *et al* 2007, Piwowar and Chapman
165 2010, Christensen *et al.* 2019, Maitner *et al.* 2024) and increases the prospect of future
166 collaboration based on using and developing archived data and code. Having well-archived and
167 documented code also provides a clear advantage for Early Career Researchers that may
168 pursue careers outside of academia, where a proven ability to generate reproducible code is
169 often more of a selling point than publications (Allen and Mehler 2019, König *et al* 2025). There
170 are many benefits to working reproducibly (Markowitz 2015), from helping with the continuation
171 of research, to avoiding errors that could later influence results and ultimately require correction
172 or retraction of published work. Data and code quality control further forces authors to be doubly
173 sure that their dataset is accurate and that the code they used generates the expected results.

174

175 While authors benefit most, we believe there are also a multitude of additional benefits for
176 journals, readers and the wider research community. For instance, it is in a journal's best
177 interest to be the purveyor of high quality and reliable scientific research. Adopting data and
178 code quality control signals to readers and the general community that scientific quality and
179 transparency are priorities for the journal. By increasing transparency of analyses, data and
180 code quality control can allow a journal to build its reputation as a reliable and trustworthy
181 source of high-quality science. Through the provision of higher quality data and code, quality
182 control increases the impact of both the article being evaluated, and the journals in which the
183 manuscript is published. Furthermore, it allows other authors and researchers to extend and
184 reuse data and code for further analyses, which can lead to an extended impact for both the
185 original journal and author(s). Finally, ensuring the archiving of high-quality data and code
186 facilitates rigorous post-publication evaluation of claimed results. By increasing transparency
187 and reproducibility of analyses, data and code quality control will therefore increase the trust of
188 published work.

189

190 We hope that an emphasis on the quality of archived data and code may additionally help to
191 facilitate data and code review (the detailed evaluation of code; Ivimey-Cook *et al* 2023) within
192 research groups prior to submission, creating more opportunities to actively involve co-authors
193 in a study and resulting in a robust and healthy lab culture that promotes cohesion. Such
194 practices can inspire early-career researchers involved in the study by promoting open science
195 through practical experiences, while helping to strengthen trust in scientific integrity in the long
196 term.

197

198 In this article, we outline what data editors are, discuss the costs and benefits of data and code
199 quality control, and then provide detailed guidelines for data editors that can be used for data
200 and code quality control in journals.

202 **Table 1.** Glossary of key terms

Term	Definition
Metadata	Refers to a description and information about the data and code. Typically in the form of a text file called a README. Other variations on this are possible, e.g., a codebook or a data dictionary.
Data Editor	An editorial position at a journal. The responsibility of this editor is to screen and quality control the data and code that will be publicly archived alongside manuscripts under review at the journal.
Data and Code Quality Control	The process of checking the suitability of data and code for public archiving.
Data and Code Archiving	The process of depositing data and code in a public repository.
FAIR principles	Findable, Accessible, Interoperable, and Reusable principles for Data (Wilkinson <i>et al</i> 2016) and Code (Barker <i>et al.</i> 2022). See https://www.go-fair.org/fair-principles/
Raw data	Unprocessed and unfiltered data. This would include any raw files e.g., photos, audio recordings, videos, and data sheets.
Data Filtering	The process of removing some data to create the dataset used in the analysis (e.g., removal of individuals with a certain characteristic). We refer to the resulting data as <i>Filtered Data</i> .
Data Processing	Transforming data from one form to another. Includes data that is extracted from images or videos, data that has been summarised, transformed, or is the result of calculations. We refer to the resulting data as <i>Processed Data</i> .

Repository	A framework providing long term storage for many projects, such as Zenodo, Dryad, Figshare, etc
Project	A collection of files archived for a specific manuscript (note, this is similar to what GitHub refer to as a repository)

203 **What is Data and Code Quality Control and What is it Not?**

204 Data and code quality control by data editors is about increasing the quality of the archived data
205 and code, and ensuring that they meet minimum standards (e.g., the data are complete and
206 usable, and the code is documented and runnable). The guidelines we lay out in the section
207 below give a detailed explanation of what data and code quality control by data editors entails.
208 Ideally, data editors would ensure that archived data and code achieve the goals laid out in Box
209 1, to allow data reuse, to increase transparency, and to provide computational reproducibility.
210 The importance of these goals may vary across different groups; journals will likely focus more
211 on transparency, whereas readers may be more concerned with data reuse and computational
212 reproducibility. These goals require varying levels of code and data checking, and so, during the
213 early stages of journals recruiting data editors, not all of these goals may be achievable. We
214 also acknowledge that not everyone may agree that achieving all of these goals is the ultimate
215 objective of data and code quality control by data editors. In practice, journals may prioritise and
216 implement goals aligned with their editorial policy over time, while allowing the data editor to
217 apply them flexibly on a manuscript-by-manuscript basis depending on the type of data and
218 tools used by the authors. While the importance of each goal may differ among stakeholders,
219 they each help improve the openness, reliability, and transparency of the scientific publication
220 process.

221

222 Importantly, whilst data editors are responsible for checking that the archived data and code
223 meet certain minimum standards, they are not responsible for *reviewing* data and code, and so
224 data editors will rarely themselves detect errors or fraud. Data quality control is not about
225 verifying the actual data (e.g., detecting data fabrication) but rather ensuring that data is
226 available, in the appropriate format, and has the corresponding metadata to be scrutinised. The
227 presence of a data editor at a journal will therefore not necessarily prevent fabricated data being
228 published. We also make a clear distinction here between code *quality control* and code *review*
229 (Ivimey-Cook *et al* 2023, Hillemann *et al* 2025). Code review is the detailed evaluation of code
230 including assessing factors such as alignment between the code's intended or stated purpose
231 and its actual implementation, the consistency of coding style, and the efficiency of the code,
232 that goes beyond the task of a data editor. Code review is an important part of research (Ivimey-
233 Cook *et al* 2023) and we encourage that research groups engage in this practice as a way to
234 improve the quality of published research (Bavota and Russo 2015). However, data editors are
235 not experts in every field of study, nor are they statisticians or specialists in all programming
236 languages. Therefore, data and code quality control should not extend to assessing the
237 suitability of analyses or code itself.

238 Are there Costs to Data and Code Quality Control?

239 Although journals adopting data and code quality control will increase the quality of data and
240 code archiving associated with published articles, which we believe will have widespread
241 benefits to journals, authors, readers and the wider research community (see above), we
242 acknowledge there may be some costs to the widespread adoption of this process.

243

244 First, adopting data and code quality control may present an additional burden for a journal. This
245 is likely to primarily impact the length of time required for peer review. To mitigate this problem,

246 several journals currently have data and code quality control alongside peer review (see
247 *Suggestions for Journals* at the end of this article). For journals that currently have in-house
248 data editors, data and code quality control does not create a per-manuscript burden to find extra
249 reviewers.

250
251 The process of data and code quality control may add a time burden to authors (although not if
252 authors were already adhering to many journal's existing requirements on data and code
253 archiving). However, this time burden will reduce over time as data and code quality control
254 becomes standard practice and making well-documented data and code becomes a natural part
255 of a researcher's workflow. This short-term investment will also come with both short and long-
256 term benefits as outlined above. Increasingly, data and code management skills have wide
257 applicability and are becoming part of routine teaching at undergraduate or postgraduate level
258 (Kohrs *et al* 2023). We acknowledge that the costs to authors will fall disproportionately on
259 those with less access to training on open data and code practices. However, these researchers
260 are actually those that may benefit most from the process of data and code quality control,
261 which is designed to aid researchers adhere to data and code archiving requirements. Those
262 with lower access to training therefore stand to gain the most from interactions with data editors
263 and the resulting skills learned from increasing the quality of their data and code archiving.

264
265 We acknowledge that researchers with higher access to training are more likely to be those
266 appointed as data editors. From one perspective, this can be seen as reinforcing existing
267 hierarchies, however, it can also be argued that these more privileged individuals should take
268 on the burden of these service roles to help increase access to this training across diverse
269 groups.

270

271 Finally, ensuring that all data and code are archived and checked for computational
272 reproducibility will necessitate more resources for data storage and re-running potentially
273 computationally expensive analyses. Data storage is already reported as a leading cause of
274 increased carbon footprint
275 ([https://direct.mit.edu/imag/article/doi/10.1162/imag_a_00043/118246/Ten-recommendations-](https://direct.mit.edu/imag/article/doi/10.1162/imag_a_00043/118246/Ten-recommendations-for-reducing-the-carbon)
276 [for-reducing-the-carbon](https://direct.mit.edu/imag/article/doi/10.1162/imag_a_00043/118246/Ten-recommendations-for-reducing-the-carbon)). However, we would argue that to prevent data loss, the data and code
277 behind any study should always be responsibly archived, regardless of the process of data and
278 code quality control. As we outline in the guidelines below, we also do not advocate for the
279 storage of multiple instances of the data, if it is already stored in a public archive. Re-running
280 analyses will also have an environmental impact. However, there is a limit to what data editors
281 are reasonably expected to re-run, and so it is unlikely that highly computationally expensive
282 analyses will routinely be repeated.

283

284 SORTEE Guidelines for Data and Code Quality Control

285 At the time of writing, the practice of data and code quality control is highly variable both across
286 and within journals. To address this, the Society for Open, Reliable and Transparent Ecology
287 and Evolutionary biology (SORTEE) started a working group with 22 data editors from 6 journals
288 in Ecology and Evolutionary (EE) biology, comprising American Naturalist, Behavioural Ecology
289 and Sociobiology, Ecology Letters, Journal of Evolutionary Biology, Peer Community Journal,
290 and Proceedings of the Royal Society B. The goal of this group was to propose a set of
291 structured guidelines for standardising the process of data and code quality control across all
292 EE journals. As a whole, these guidelines provide a high bar for data and code quality control,
293 and so either all or parts of these guidelines can be adopted by journals and presented to
294 authors and readers. We note that validation of Stages 1-4 of these guidelines *by data editors*

295 achieves the highest level (Level 3) of the Transparency and Openness Promotion Guidelines
296 (TOP2025; Grant *et al* 2025), for data and code.

297

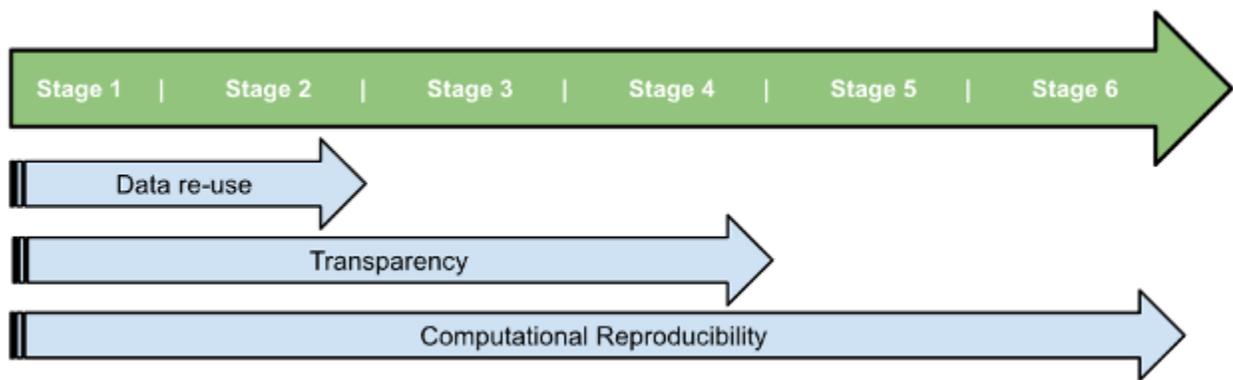
298 We see several benefits of adopting a set of standardised guidelines both for authors and data
299 editors. First, authors know what is expected of them regardless of the journal. By working
300 towards the same standards of data and code archiving in advance of submission, authors can
301 easily submit their manuscripts to a variety of journals, or transfer their manuscripts between
302 them, and know that they do not need to change their submissions to meet different standards
303 across journals. Ultimately, this reduces the burden for authors and streamlines the submission
304 process. In addition, standardisation of this workflow will encourage authors to share data and
305 code even when not explicitly required, help those with less experience of sharing data and
306 code, and facilitate the community-wide adoption of open data and code as a default. Second,
307 data editors have a standardised template for review, designed by both open science advocates
308 and active and experienced data editors from multiple journals, to gain the balance of idealism
309 and practicality. This can make the process both more efficient for data editors and more
310 thorough for the community. Ultimately, standardisation will help inform decision-making for the
311 journals. Third, readers know what checks have occurred prior to publication, which can
312 ultimately help build additional trust in scientific reporting and open science practices,
313 particularly when computational reproducibility has been assessed.

314

315 In an ideal scenario, data and code quality control helps achieve the goals of data reuse,
316 transparency, and computational reproducibility (see Box 1). In reality, depending on the time,
317 computing resources, and expertise of the reviewing data editor, only some of these will be
318 feasible at a large scale. We propose that data and code quality control can be broken into six
319 stages in the following coherent order to sequentially address the goals of data and code
320 archiving outlined above (Figure 1):

- 321 ● Stage 1: Data must be archived and adhere to FAIR guiding principles.
- 322 ● Stage 2: Archived data corresponds with the data reported in the manuscript.
- 323 ● Stage 3: Code must be archived and adhere to FAIR guiding principles.
- 324 ● Stage 4: Archived code corresponds with the workflow reported in the manuscript.
- 325 ● Stage 5: Archived code runs with the archived data.
- 326 ● Stage 6: Results can be computationally reproduced by running the archived code.

327 We discuss each of these stages in more detail below and provide guidelines for how these can
 328 be assessed. Note, each component of each stage is associated with its own guideline. To help
 329 data editors perform their assessment according to these guidelines, we have created an app,
 330 that can be accessed at <https://github.com/SORTEE/DC>.



331
 332 **Figure 1.** A diagrammatic representation of how the three goals of data and code archiving (blue arrows)
 333 match the stages of data and code quality control (green arrow). Stages 1 and 2 (1 = Data must be
 334 archived and adhere to FAIR guiding principles and 2 = Archived data corresponds with the data reported
 335 in the manuscript) are needed to achieve the goal of data re-use. Stages 1-4 (3 = Code must be archived
 336 and adhere to FAIR guiding principles. and 4 = Archived code corresponds with the workflow presented in
 337 the manuscript) are needed to achieve the goal of transparency. Stages 1-6 are needed to achieve the
 338 goal of full computational reproducibility (5 = Archived code runs with the archived data and 6 = Results
 339 can be computationally reproduced by running the archived code). Figure created by EIC.

340
 341 **Stage 1. Data must be archived and adhere to FAIR guiding principles.**

342 For data to be open and amenable for reuse, it must adhere to FAIR guiding principles
 343 (Findable, Accessible, Interoperable and Reusable). Data must be placed in an open repository

344 (*Accessible*) with a permanent Digital Object Identifier (DOI, or another globally unique and
345 persistent identifier) that is cited in the manuscript (*Findable*) with a licence that describes re-
346 use (*Reusable*). Metadata (e.g., a README) must also be present to describe the data
347 (*Findable, Accessible and Reusable*). The data itself must be in a machine-readable, non-
348 proprietary format (*Interoperable*). We discuss each of these points in more detail below. By a
349 data editor assessing the archived data to Stage 2 of these guidelines, journals would achieve
350 TOP 2025 level 3 for Data Transparency (Grant *et al* 2025).

351

352 *Stage 1.1. Data files are accessible and in an open repository*

353 For data to be readily Accessible and Findable, it must be archived in a public data repository
354 with an associated persistent DOI (or any other globally unique and persistent identifier, such as
355 ARK (for archives, and datasets), Handle (for digital objects), Accession Numbers ('omics' data,
356 e.g., GenBank), RRID (research resources)) that is separate from the DOI of the resulting
357 published article. Furthermore, the data must be clearly cited in the manuscript, and listed with
358 its DOI (or other identifier) in the reference list, so readers know where to access the underlying
359 data. There are a multitude of different repositories to fit a variety of needs (see
360 <https://doi.org/10.5281/zenodo.10651775> for information about repositories; Harvard Longwood
361 Medical Area Research Data Management Working Group 2023). An important feature of a
362 repository is that it guarantees long-term storage and file immutability (i.e., they cannot be
363 deleted or modified once published). Common general repositories that provide these include
364 DataVerse, Dryad, Figshare, and Zenodo. There are also additional topic specific repositories,
365 such as GenBank for depositing genetic and other biological sequences. We note that whilst
366 GitHub is popular, it does not produce DOIs and files can be changed or deleted, and so studies
367 that use GitHub should be linked to a data repository for archiving prior to submission (e.g.,
368 Zenodo). Similarly, the Open Science Framework (OSF) does not provide file immutability;
369 projects and files can be changed or deleted. Data should not be provided as supplementary

370 material attached to the manuscript online as this does not provide a globally unique persistent
371 identifier and may not be open and accessible; all data must be archived in a public repository
372 (see Stage 1.3. for sensitive data).

373

374 When necessary, projects in public data repositories can be anonymised to adhere to the
375 journal's policy of double blinding (see: [https://methodsblog.com/2023/08/23/double-
376 anonymous-peer-review-frequently-asked-questions/](https://methodsblog.com/2023/08/23/double-anonymous-peer-review-frequently-asked-questions/)). Furthermore, many repositories offer
377 embargoes, if necessary.

378

379 **Guideline 1.1: Data are open and freely available (with some exceptions, see Stage 1.3.)**
380 **and are located in a permanent public repository with an associated globally unique**
381 **persistent identifier, that is cited in the text and reference list of the manuscript .**

382

383

384 *Stage 1.2. Data are associated with a license*

385 Data must be associated with an appropriate license that indicates how the data can be shared
386 and re-used. In our experience, most researchers have little knowledge of such licenses.

387 Without a license, data cannot be legally reused under many circumstances (e.g., depending on
388 the jurisdiction; <https://choosealicense.com/no-permission/>). Therefore, to avoid confusion, it is
389 important for authors to specify a license that outlines how their data can be used, and whether

390 or not attribution is required. There are several different licenses to choose from but typically

391 Creative Commons licenses are used for data (see: [https://chooser-
392 beta.creativecommons.org/](https://chooser-
392 beta.creativecommons.org/)), with several repositories including a license by default. The most

393 permissive license is the CC0 license, which puts the data freely into the public domain with no
394 requirement for attribution. Some repositories assign this license by default (e.g., Figshare) or

395 mandate the use of this license (e.g., Dryad). Another commonly used license is CC-BY 4.0,

396 where reusers of the data must give credit to the original author but are allowed to distribute,
397 remix, adapt, and build upon the created material (including for commercial uses). This license
398 is also used as a default by some repositories (e.g., Zenodo). There are several other more
399 restrictive licenses, for example the CC-BY-NC 4.0 which prohibits commercial use. Note that
400 these guidelines do not recommend or enforce any specific type of license.

401

402 **Guideline 1.2: Data must be associated with a license.**

403

404

405 *Stage 1.3. Data files are present and complete*

406 The simplest but most important requirement is that the data supporting the results presented in
407 a manuscript must be complete in the archived project. Ideally, raw data and processed data
408 should both be provided (see Table 1). The term “raw data” refers to all collected data prior to
409 any filtering (subsetting of data based on reported exclusion criteria) or processing (extraction,
410 transformation, summarising, aggregation, and prior to any formal calculations; see Table 1).
411 Note that we do not count transcribing data from a written to a digital format and subsequent
412 error-checking of data entry errors as processing or filtering.

413

414 There are several reasons why the raw data should be archived: First, to prevent data loss,
415 which is achieved by archiving the most complete dataset possible; Second, to maximize data
416 re-use, as only providing filtered data can exclude particular future uses; Third, the process of
417 filtering and processing data is prone to mistakes (e.g., coding errors). Such errors are a natural
418 and inevitable part of the research process, but being able to detect them makes the scientific
419 process more efficient and reliable, and identifying and correcting these mistakes is only
420 possible if the raw data are available. Finally, to increase transparency, allowing the reader a

421 clearer insight into the process that resulted in the final dataset used for analysis. We note that
422 the data required to be archived also depends on the goal; computational reproducibility of the
423 results presented in a manuscript can be achieved with processed data, whereas the goal of
424 data reuse is dependent on raw data being archived. What constitutes raw data is often reliant
425 on the nature of the data (see below). Ultimately, whether the archived data are most
426 appropriate for a given manuscript will be at the discretion of the data editor and dependent on
427 journal policy. Below we provide some guidance for specific cases.

428
429 In the simplest case, data have been collected for a stand-alone study. In this case, the raw
430 data are simply all the collected data, and should be provided in full (for exceptions see below).
431 If data originates from videos, images or sound files, then these are considered to be raw data.
432 Therefore, where possible, these files should also be made available. The processed data
433 should be provided alongside the raw data, with a description of the processing/filtering in the
434 metadata (if this is not already described in the code files). This is particularly important if the
435 raw data are not interoperable (e.g., outputs from proprietary software; see 1.4 below). Although
436 most databases allow a considerable amount of data to be stored (Per project: Dryad - 300 GB,
437 Zenodo - 50 GB, Figshare - 20 GB, Dataverse - 10GB), raw data may exceed these limits (e.g.,
438 video data that is several terabytes large). In such cases where the data are too large to be
439 feasibly uploaded, then a representative, manageable subset of this raw data should be
440 provided, so the extraction process can be assessed (e.g., providing several example videos).

441
442 If the data used in an analysis originates from a larger database (e.g., from a long-term study)
443 then ideally the entire database would be considered the raw data. We can foresee many
444 circumstances where the authors may feel this is inappropriate, for example, due to worries
445 about the data being used without permission (Mills *et al.* 2015, but see Evans 2016 for an
446 empirical assessment) or misused (Weissgerber *et al.* 2024). In such cases, filtered data may

447 be provided, alongside clear details of the filtering process that would allow the same data to be
448 extracted at a later point (e.g., location and version of the database that the data was extracted
449 from, how it can be accessed, the database queries used to extract the data or other similar
450 instructions of how to generate the same subset for analysis, and any exclusion criteria). Data
451 editors may need to assess the suitability of archived data on a case-by-case basis to ensure
452 that the data are provided in the rawest form possible according to the journal guidelines, and
453 that sufficient information on the generation of archived datasets is present. If the database is
454 already open, rather than re-archiving the data (which, if large, may come with environmental
455 costs) the authors can cite the database, include a clear description of what data were used, the
456 data extraction procedure, and where appropriate, provide an immutable snapshot of the
457 database if it is dynamic.

458

459 In some cases, restrictions may apply to making raw data publicly available. For instance, if the
460 dataset contains sensitive information about medical records, identifying personal information
461 (which may breach General Data Protection Regulations (GDPR)), or geographic locations
462 pertaining to endangered species or fossil sites at risk of vandalism (see Chapman 2020). In
463 many cases, data can often be obfuscated or anonymised to enable data archiving. There may
464 also be issues with indigenous data sovereignty (for best practices on governance and
465 stewardship of indigenous data in combination with FAIR principles see CARE principles
466 (Collective benefit, Authority to control, Responsibility, Ethics); Carroll *et al* 2020). Processed
467 data used in the manuscript should instead be provided alongside suitable metadata which
468 describes the raw data in as much detail as possible, while still preserving anonymity and
469 sovereignty. Where data cannot be provided, simulated data with the same structure and
470 properties could also be provided, to allow for Stage 5 to be assessed. Information about how
471 and where to make data requests should also be included in the metadata.

472

473 In all cases, the data availability statement in the manuscript should clearly outline whether the
474 authors have archived raw and/or processed data. This section should also contain guidance on
475 how to access and request the raw data if necessary and appropriate.

476

477 **Guideline 1.3: Authors must either provide:**

478 **a) raw data, along with the processed data and/or code to prepare the data for analysis,**

479 **or**

480 **b) a sample of raw data alongside processed/filtered data when full raw data upload is**
481 **not possible , or**

482 **c) processed/filtered data with a detailed description of how to both obtain and**
483 **process/filter the raw data.**

484

485

486 *Stage 1.4. Data files are in an interoperable format*

487 To both facilitate review and allow reuse, data must be in a universally interoperable format,

488 meaning that the data can be exchanged and used across different software and operating

489 systems. File types specific to proprietary software (e.g., .sps files from the SPSS program) are

490 not interoperable, so do not facilitate data re-use. For example, .xls files are a proprietary

491 format, whereas .xlsx files are not, meaning they are interoperable. However, .xlsx files can

492 contain information that is lost when importing data into statistical software (e.g., formatting).

493 Similarly, tabular data are sometimes archived in a .RData file (or equivalent). Although this can

494 be used with open source software (i.e., R), again, this data format restricts its use, as it

495 requires knowledge of R to extract the data, and may be dependent on the version of R that was

496 used to save it. Simpler text-based file formats such as .csv (comma-separated-values), .tsv

497 (tab-separated-values) and .txt (plain text) files provide a more interoperable format, as they can

498 be used by more software and across more systems, and so are preferable. Where possible, it

499 would therefore be more suitable to archive the raw data in a more interoperable format (i.e.,
500 .csv or .txt). Lastly, data should not be stored within PDF or Word documents, which can be
501 prone to error when data are copy-pasted (e.g., for re-using) and, which cannot be readily
502 imported into statistical programs for analysis. In some cases, there might be no option other
503 than to provide data in a non-interoperable format (e.g., if the data was collected using
504 proprietary software) but this should be provided alongside interoperable extracted data with a
505 clear description of the conversion processes in the metadata including the particular software
506 version that was used. There are continuous advances in this area; for instance, parquet files
507 are interoperable, highly compressed, efficiently read, and highly accurate for storing large
508 datasets.

509

510 **Guideline 1.4: Data files must be provided in an interoperable format.**

511

512

513 *Stage 1.5. Data metadata present and adequate*

514 Data files alone do not contain enough information for a user to fully understand their contents.
515 Data files must therefore be accompanied by metadata. The most common form of this
516 metadata is a README file, which describes and explains the content of the data and its
517 provenance. The README should provide general information about the manuscript, e.g., the
518 manuscript title and abstract, the authors and relevant contact information, date and location of
519 data collection, and a list of all relevant funders. In the case of double-blind review, some
520 sections can be left blank until acceptance (see example in Figure 2). The README should also
521 include any relevant licence information (e.g., CC-BY, see above), and information about data
522 derived from other sources (e.g., from other articles or online data). Finally, the metadata should
523 contain detailed descriptions of each data file, describing its structure and what variables it
524 contains, what units of measurement they are in, and how they link to the data described in the

525 manuscript, e.g., each column in a .csv should be explained and described (see example in
526 Figure 2). This information can be provided in several ways: 1) as part of the main README
527 file, 2) by creating additional README files to describe the data and code files (as shown in
528 figure 2), or 3) by providing a data dictionary for each data file (e.g., a .csv file with a column for
529 column names, and another for the description of the variable). We use the term “adequate”
530 here to describe data-associated metadata that is sufficiently detailed so that anyone can
531 understand the data without needing to read the resulting manuscript to understand its contents.

532

533 **Guideline 1.5: Detailed metadata, including (but not limited to) a README file, must**
534 **accompany the data (see example in Figure 2).**

535

Project Structure

Title: Age and habitat influence the weight of offspring at hatching in *Nicrophorus vespilloides*

Contributors: Edward Ivimey-Cook

Date Created: 2024-10-21

Persistent Identifier: DOI/10.12345.6789 [Stages 1.1 + 3.1]



Project Storage

Readme.txt [Stages 1.5 + 3.5]

o Data

Processed.csv [Stages 1.3 + 1.4]

Raw.csv [Stages 1.3 + 1.4]

Data_License.txt [Stages 1.2]

Data_README.txt [Stages 1.5]

o Code

00_Loading.R [Stages 3.3 + 3.4]

01_Cleaning.R [Stages 3.3 + 3.4]

02_Model.R [Stages 3.3 + 3.4]

03_Graphs.R [Stages 3.3 + 3.4]

Code_License.txt [Stages 3.2]

Code_README.txt [Stages 3.5]

[Stage 1.5] Data_README.txt

Authors: Ed Ivimey-Cook

Email: Ed@Ivimey-Cook.com

Title: Age and habitat influence the weight of offspring at hatching in *Nicrophorus vespilloides*

Funders: SORTEE

Data License: CC-BY 4.0 in Data_License.txt

Summary: Data collected in May-June 2024 in Blackford Hill, Edinburgh on age, habitat, and weight of offspring produced across separate breeding attempts.

Raw.csv

ID: Individual identity of female (1-58)

Age: Age of individual sampled during breeding attempt (14, 30, or 60 days)

Habitat: Habitat that individuals were sampled (A = Grassland, B = Forest, C = Swamp)

Offspring_Weight_at_Hatch: Weight of offspring produced /mg

Processed.csv

ID: Individual identity of female (1-53)

Age_days: Age of individual sampled during breeding attempt (14, 30, or 60 days)

Habitat: Habitat that individuals were sampled (A = Grassland, B = Forest, C = Swamp)

Offspring_weight_g: Mean weight of offspring produced /mg

[Stage 3.5] Code_README.txt

Authors: Ed Ivimey-Cook

Email: Ed@Ivimey-Cook.com

Title: Age and habitat influence the weight of offspring at hatching in *Nicrophorus vespilloides*

Funders: SORTEE

Code License: MIT License in Code_License.txt

Code

00_Loading.R: Loading of essential functions and packages. Sourced in 01-03.

01_Cleaning.R: Imports Raw.csv - Cleaning functions to produce Processed.csv.

02_Model.R: Imports Processed.csv - Run the model on mean offspring weight at hatching against age and habitat.

03_Graphing.R: Imports Processed.csv - Ggplot2 graphs to produce Figure 1 of offspring weight at hatching.

Software and {Packages}

R version v4.3.1

{ggplot} v3.5.1

{lme4} v1.1-35.5

{emmeans} v1.10.2

{dplyr} v1.1.4

{lmerTest} v3.1-3

536

537

538

Figure 2. Example project structure and metadata (two README files) showing how the various components adhere to the SORTEE guidelines for data and code curation. The numbers in red refer to

539 *the stages being addressed. The Data README should contain information on the manuscript (authors*
540 *with corresponding contact details, the title of the manuscript, and any funders), the license file, along*
541 *with information of the data (a brief summary of collection, and column-by-column description of the data*
542 *files along with any measurement units or levels of factors). For code, the README should contain the*
543 *same information as the data initially (information on the manuscript and code license) but also contain a*
544 *description of each code file in the order they are meant to be used (which also clearly indicates which*
545 *data file is used in each script). Lastly, the README should contain a list of all software and packages*
546 *used with associated version numbers. Figure created by EIC.*

547

548

549 **Stage 2. Archived data corresponds with the data reported in the**

550 **manuscript**

551 For archived data to support a manuscript, as well as being present in a form that facilitates
552 reuse, it must correspond with the data reported in the manuscript. For this to be assessed, the
553 data editor needs to check that the variables and data described in the manuscript (most likely
554 in the Methods) are present in the data files provided. For example, if the manuscript mentions
555 that offspring weight was measured at three habitats, the data file should contain an offspring
556 weight variable and a habitat variable (see example in Figure 3). The dimensions of the data
557 should also correspond with those described in the manuscript; discrepancies in the size of the
558 dataframe may suggest that some unreported data processing or filtering has taken place. A
559 clear description of all these aspects within the text is essential; without it, the data will not
560 correspond with the manuscript, undermining its potential for reuse, transparency and
561 reproducibility. Some journals use AI to facilitate this process (e.g., the DataSeer.ai application:
562 <https://dataseer.ai/>), which produces a report detailing the expected data that should be
563 provided based on the description within the manuscript.

564

565 **Guideline 2: The structure and contents of the archived data files must match the**
566 **description in the manuscript.**

Manuscript		[Stages 2 + 4] Processed.csv																																																	
<p>Methods</p> <p>We monitored a population of <i>Nicrophorus vespilloides</i> over three habitats (Grassland, Forest, and Swamp) on Blackford Hill, Edinburgh from May-June 2024. Over this time, we discovered 58 broods (23 Grass, 20 Forest and 15 Swamp), from which we took one larva and brought them back to the lab for rearing. We then monitored three breeding attempts (at 14, 30, and 60 days post-eclosion) for each of the resulting beetles. For each breeding attempt, we measured the hatching weight of offspring born to each brood (in mg). Five individuals died during the experiment and so we removed them from the dataset before analysis. In total we analysed records of 159 broods (53 females each with three broods). To test whether age and habitat affected hatching weight, we first took the mean weight of offspring at hatching per individual per age class. We then modelled hatching weight as a function of age (continuous) and habitat (factor), including a random factor of individual ID in a linear mixed model. [Stages 2 + 4]</p> <p>Results</p> <p>The effect of age on offspring weight was significantly negative (Age = -0.03, SE = 0.004, $p < 0.001$). [Stages 5 + 6]</p>		<table border="1"> <thead> <tr> <th>1</th> <th>ID</th> <th>Habitat</th> <th>Age_days</th> <th>Offspring_Weight_g</th> </tr> </thead> <tbody> <tr><td>2</td><td>1</td><td>A</td><td>14</td><td>0.22</td></tr> <tr><td>3</td><td>1</td><td>A</td><td>30</td><td>0.33</td></tr> <tr><td>4</td><td>1</td><td>A</td><td>60</td><td>0.21</td></tr> <tr><td>5</td><td>2</td><td>B</td><td>14</td><td>0.28</td></tr> <tr><td>6</td><td>2</td><td>B</td><td>30</td><td>0.24</td></tr> <tr><td>7</td><td>2</td><td>B</td><td>60</td><td>0.22</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>160</td><td>53</td><td>C</td><td>60</td><td>0.34</td></tr> </tbody> </table>					1	ID	Habitat	Age_days	Offspring_Weight_g	2	1	A	14	0.22	3	1	A	30	0.33	4	1	A	60	0.21	5	2	B	14	0.28	6	2	B	30	0.24	7	2	B	60	0.22	160	53	C	60	0.34
1	ID	Habitat	Age_days	Offspring_Weight_g																																															
2	1	A	14	0.22																																															
3	1	A	30	0.33																																															
4	1	A	60	0.21																																															
5	2	B	14	0.28																																															
6	2	B	30	0.24																																															
7	2	B	60	0.22																																															
...																																															
160	53	C	60	0.34																																															
		[Stages 2 + 4] O2_Model.R																																																	
		<pre>####02.1 Modelling code #Load packages source("00>Loading.R") #Import processed data Processed_data <- read_csv("Processed.csv") #Lmer of offspring weight with age and habitat #with a random effect individual ID - requires #"Processed.csv" model1 <- lmer(Offspring_weight_g ~ Age_days + Habitat + (1 ID), data = Processed_data) summary(model1)</pre>																																																	
[Stages 5 + 6] Terminal																																																			
<pre>> model1 <- lmer(Offspring_weight_g ~ Age_days + Habitat + (1 ID), data = Processed_data) > summary(model1) Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest'] Formula: Offspring_weight_g ~ Age_days + Habitat + (1 ID) Data: Processed_data Fixed effects: Estimate Std. Error df t value Pr(> t) (Intercept) 5.051133 0.196697 195.718112 25.680 < 2e-16 *** Age_days -0.030746 0.003729 191.946582 -8.246 2.54e-14 *** HabitatB -0.056885 0.090364 189.647801 -0.630 0.530 HabitatC 0.058789 0.091050 188.100452 0.646 0.519</pre>																																																			

568

569

570

571

Figure 3. Matching a manuscript to archived data and code. The methods in the manuscript can be checked against the archived data (Stage 2) and code (Stage 4). In terms of the data, the same variables that are described in the manuscript need to be present in the data, and the data need to be the same

572 size as referred to in the manuscript. In terms of the code, the models that are described in the
573 manuscript need to be clearly labelled in the code. The numbers in red refer to the stages of the
574 guidelines being addressed. Figure created by EIC.

575

576 **Stage 3. Code must be archived and adhere to FAIR guiding principles**

577 To facilitate transparency and computational reproducibility, all code used to reproduce the
578 results should be provided alongside data. The guidelines for code archiving are broadly similar
579 to those of data archiving outlined above, with a few subtle differences, which we outline below.
580 By a data editor validating Stages 3 and 4 of these guidelines, journals would achieve TOP
581 2025 level 3 for Analytic Code Transparency (Grant *et al* 2025).

582

583 *Stage 3.1. Code files are accessible and in an open repository*

584 As with data, code files must be accessible within an open repository with an associated
585 globally unique persistent identifier (see Stage 1.1 for suitable repositories). This can either be
586 the same repository as the data or a separate one. This choice may depend on the chosen
587 repository. For example, Dryad recommends only archiving data, and directs users to archive
588 code with Zenodo, as code is not always compatible with a CC0 license, which Dryad
589 mandates. Care must be taken when archiving code and data separately, as the two archived
590 projects should clearly link to each other as well as to the manuscript, and authors must provide
591 information about how to structure the data and code directories so that the code will run with
592 the data. For example, if the code assumes that the data are in the same parent directory within
593 a folder called 'Data' (see Figure 2), then the data files will have to be organised like this for the
594 code to run. Where possible, we suggest that data and code are archived together as this will
595 minimise issues with computational reproducibility (see Stage 5). In some cases, this may not
596 be possible, particularly if there are large data files that are stored in a field-specific repository
597 (e.g., genomic data stored on GenBank). In this case, we do not expect data and code to be

598 archived together. Instead, details of how to access and organise the data to work with the
599 archived code need to be included in the code README. Similar to data, code should not be
600 included in the supplementary material of a submission. Again, if necessary, the archived
601 projects can be anonymised (see Stage 1.1 above).

602

603 **Guideline 3.1: Code files are open and freely available and are located in a permanent**
604 **public repository with an associated globally unique persistent identifier, preferably in**
605 **the same archived project as the data files, and are cited in the text and the reference list**
606 **of the manuscript.**

607

608

609 *Stage 3.2. Code is associated with a licence*

610 As with data, any archived code must also have an associated license to enable code sharing
611 and repurposing. It is worth noting that typically licenses used for code differ from those used
612 with data (i.e. creative commons licenses), and there is a multitude to choose from. We suggest
613 using permissive code licences whenever possible, for example, MIT, BSD 2-Clause, GNU, and
614 Apache (see: <https://choosealicense.com/>).

615

616 **Guideline 3.2: The code must be associated with a license.**

617

618 *Stage 3.3. Code files are present and complete*

619 Without code alongside the data used in the analysis (and the computational environment in
620 which the analysis took place - see 3.5), full transparency and computational reproducibility are
621 impossible to achieve. At a minimum, the analytical code used for statistical analyses and

622 graphing should be present, but we recommend providing all parts of the analysis pipeline, from
623 data filtering and processing to model analysis and graphing.

624

625 Analyses are not always done in programming languages (e.g., R). However, several analytical
626 programs (particularly those that use Graphical User Interfaces (GUIs)) will output a script or log
627 detailing the analysis procedure (e.g., SPSS or Minitab), which should then be archived. If this is
628 not possible, the researcher should clearly document which menu options were selected in the
629 GUI and in what order, with sufficient detail to enable reproducibility. Alternatively, they can
630 provide screenshots showing all selected options during the analysis. It should be noted that
631 these output scripts or instructions with GUI-based software are often proprietary so will still limit
632 reproducibility (discussed further in stage 6).

633

634 **Guideline 3.3: All code used for generating the results of the manuscript (including**
635 **filtering, processing, graphing, and analysis) must be present either in one or more code**
636 **files.**

637

638

639 *Stage 3.4. Code is in an interoperable format*

640 For code files to be opened and usable, it is essential they are provided in an interoperable
641 format such as a text (.txt), R (.R) or Python (.py) file. These file formats can be readily opened
642 by text editors and other integrated development environments (such as VSCode). Code must
643 not be provided as a .pdf or pasted into a Word document, because even if the script can be
644 copied and pasted, this increases the risk of unintentional errors, as these programs often insert
645 additional characters (or spacing) that can be misinterpreted by the analysis software (e.g.,
646 Python code failing to run due to improper indentation).

647

648 **Guideline 3.4: Code files must be provided in an interoperable format.**

649

650

651 *Stage 3.5. Code metadata present and adequate*

652 Code metadata must be present in two different forms, in a separate README file and also

653 within the code file itself. As with data, a detailed README file must be provided along with the

654 code files that describes general information about the manuscript, e.g., manuscript title and

655 abstract, the authors and contact information, list of all relevant funders, the globally unique

656 persistent identifier of related data (if different), and information about the code license.

657 Additionally, the README file should also include an outline of the workflow of the code (if

658 multiple files exist), how to use it with the data (if archived separately) and a brief description of

659 each code file including what data they require (i.e., raw or processed data), what they do (i.e.,

660 filtering, processing, modelling etc) and what they produce (e.g., Figure X or Table X). The

661 README should include details of the name and version of the analytical software used (e.g., R

662 or Python) along with the names and version numbers of the loaded (not base) packages used

663 (e.g., these can be obtained using `sessionInfo()` in R). This information is essential for detailing

664 the computational environment and enabling computational reproducibility. Finally, the

665 README should also state whether and how the authors have used large language models

666 (LLMs) in code generation (Resnik & Hosseini 2025). We note that it is unlikely that a data

667 editor would be able to verify this.

668

669 The second form of metadata is included within the code files, in the form of detailed code

670 annotation and sectioning. A header at the top of the script with a title and a quick overview of

671 what the code does can also be very helpful, especially when the whole analysis pipeline is split

672 across multiple scripts. The code should be broken down into distinct sections with clear

673 headings describing their purpose (e.g., loading packages, data processing, data filtering).

674 Annotation should then clearly describe what the code does, how to run it (if necessary, the
675 length of time it might take, for example if the code takes multiple hours to complete), and what
676 it produces. From the perspective of a data editor, the most important thing is that sections of
677 the code are clearly signposted to help assess Stage 4 (see below). Therefore, line-by-line
678 annotation, whilst important to readers and users, is not as vital as clear labelling of sections of
679 code and their purpose for data editors. There is no expectation that the data editor will
680 understand all the code they quality control, and it should not be the role of data editors to
681 review, interpret, or correct the code. Similar to the data metadata (stage 1.5), the term
682 “adequate” refers to the code metadata providing all the information necessary to understand
683 the analysis code without reading the manuscript.

684

685 **Guideline 3.5: A sufficiently detailed README must accompany the code. Code must**
686 **also be broken into sections with clear annotation stating the purpose of the code with**
687 **clear links to the relevant sections, figures, and tables in the manuscript.**

688

689

690 **Stage 4. Archived code corresponds with the workflow reported in the**
691 **manuscript**

692 It is crucial that all the code needed to reproduce the results of the manuscript, including any
693 supplementary material, is archived (Figure 3). This stage should not involve the data editor
694 critiquing the analytical techniques used or performing a formal code review (Ivimey-Cook *et al*
695 2023, Hillemann *et al* 2025; although annotation is required for transparency, see Stage 3.5). It
696 should rather involve an assessment of whether the specific code is present to perform all
697 stages of the analyses, including producing any graphs and subsequent results stated within the
698 manuscript. At this stage, we are also not interested in whether the code reproduces the results

699 in the manuscript, just that the code is *present* to produce the results. As a data editor is unlikely
700 to be an expert in all analyses, across all packages or all programming languages, clear code
701 annotation and signposting by the author is necessary for this to be assessed. To our
702 knowledge, there is currently no software that performs the same task for code as DataSeer
703 does for data. However, given the rapid progress in AI, such a tool may become available soon
704 (Cooper *et al* 2024).

705

706 **Guideline 4: The structure and content of the archived code must match the description**
707 **of data filtering, processing, and analysis, and the presentation of results in the**
708 **manuscript .**

709

710

711 **Stage 5. Archived code runs with the archived data**

712 This stage is a prerequisite for full computational reproducibility. The data editor must be able to
713 run the code with the provided data and code metadata, using the described software, without
714 errors. The metadata provided must therefore be sufficient for a reader to install appropriate
715 programs and libraries (and their versions) required to run the code, and to understand which
716 code files should be run and in what order. If the data editor cannot run the code with the
717 archived data and metadata, they cannot progress to the last stage of the guidelines, and so the
718 data editor should then ask the authors to fix the issue. Common issues include a package or
719 module not being installed or loaded within the code, a missing code chunk, variable names in
720 the code and data files not matching, and code referencing data files with names that do not
721 match the archived data files. We stress that it is not the responsibility of the data editor to solve
722 these problems and make the code run as intended, but rather the onus should be placed on
723 the authors.

724

725 One of the most common reasons code does not run is due to the use of local or absolute file
726 paths that do not transfer to another user's operating system. A more reproducible way of
727 specifying file paths is to use relative file paths and there are multiple ways to do this. A
728 common way for RStudio users is creating an RStudio project file (.Rproj), or similarly using the
729 {here} R package (Muller and Bryan 2020) outside of RStudio (see
730 <https://docs.posit.co/ide/user/ide/get-started/>). Alternatively, local file paths can be specified
731 when R is opened, for example by opening R within a certain directory when using the terminal
732 or using an integrated development environment (IDE) that allows users to specify a project
733 folder (e.g., the R GUI or VSCode). Whichever method is used should be noted in the metadata.
734 Given the multitude of methods, the use of absolute file paths or a different method of specifying
735 relative file paths than the data editor is familiar with should not be a reason for a data editor to
736 return the code to an author, as long as the data editor can make it run on their computer with
737 minor changes. We class this as a minor error that should simply be noted in the final review.

738

739 Similarly, data editors should not be expected to install exact software versions in the first
740 instance. If there are errors upon running (or results do not match - see Stage 6), then the data
741 editor should note this in their review, and the corresponding version should be installed and
742 code run again. Finding that the results are not reproducible with different versions can be an
743 insightful piece of information regarding the robustness of the results. In this instance, the
744 authors should clearly flag that their results are sensitive to the software version used, and
745 justify the use of a particular version in the manuscript and metadata if the results substantially
746 differ (see Stage 6).

747

748 In some cases, the data preparation or analysis may be computationally expensive and so
749 either require specialist hardware (such as access to a high performance cluster) or take a

750 considerable time to run. This should be clearly indicated within the metadata, alongside a
751 saved output. Ideally, example code should be provided that demonstrates that the code will
752 run. For example, if a statistical model will take a long time to run, the authors can provide
753 example code for an analysis of a subset of the data, or present a model that runs for a reduced
754 duration. Alternatively, data editors could also simply check that the code initiates, and then
755 terminate the run before completion. Although this does not allow computational reproducibility
756 to be fully assessed (see Stage 6), it at least demonstrates that the code runs. Similarly, the
757 code may come from proprietary software or use packages from proprietary software (e.g.,
758 ASReml; Butler *et al* 2017). In such cases, the data editor will not be able to run the code and
759 so full computational reproducibility cannot be assessed. If this is the case, this should be
760 clearly documented in the metadata. In the case that only part of the analysis requires
761 proprietary software, the metadata should clearly indicate which parts of the code can be
762 assessed by the data editor. As we outline below, in both the case of computationally expensive
763 analyses and the use of proprietary software, where possible the authors must provide saved
764 outputs from these analyses for the data editor to review. For example, outputs of large
765 Bayesian models that can take a considerable time to run can be saved (e.g., as a .rds file) and
766 archived. Build systems, such as the *targets* package in R or *snakemake* in Python can provide
767 reproducibility signatures (hashes) for computationally-intensive steps along with intermediate
768 data objects.

769
770 In some cases, the problems of using proprietary software can be overcome by ensuring that
771 the code can be executed using non-proprietary software or by providing alternative executable
772 formats. For example, GNU Octave can be used to run MATLAB code and MATLAB Compiler
773 allows converting MATLAB (.m) files into standalone applications, ensuring data editors and
774 users can run the code without owning the proprietary software. Although such alternatives can

775 provide computational reproducibility, authors must carefully test for compatibility and note any
776 limitations or differences in the metadata.

777

778 **Guideline 5: Code must be able to run without error using the archived data. With the**
779 **exception of easy to fix file path errors, all errors should be addressed by the author.**

780

781

782 **Stage 6. Results can be computationally reproduced by running the**
783 **archived code**

784 For this final stage, the data editor should assess computational reproducibility by checking
785 whether the results in text, tables, and graphs within the manuscript and supplementary material
786 match those obtained by running the archived code with the archived data. This can only be
787 assessed if the archived code runs without error (Stage 5). In most cases, we expect that exact
788 reproducibility of the results is possible (i.e. the exact number in the manuscript should be
789 generated by running the code), and any deviations would mean that the computational
790 reproducibility test has failed. In some cases, authors may have used additional software to
791 post-process figures. In these cases the data represented within the figure is still expected to be
792 the same, but the code may not reproduce the figure exactly.

793

794 One reason that the reproduced results may slightly differ is through the use of stochastic
795 methods that involve (pseudo) random number generation, such as Monte Carlo methods (e.g.,
796 simulations or Bayesian analysis using Monte Carlo Markov Chains (MCMC)) as these will
797 produce a slightly different result each time they run. However, this variation can be avoided by
798 setting a seed (e.g., using `set.seed()` function in R or `random.seed()` in Python; see Box 1) at the
799 beginning of any code section that would be run independently, which means that the

800 pseudorandom number generation is the same each time the code is run, enabling the same
801 results to be reproduced, including for analyses and figure generation (e.g., with point jittering).
802 We note that setting seed does not always ensure computational reproducibility, for instance the
803 use of `rmvnorm()` from the {MASS} R package does not create the same random numbers
804 across different operating systems due to floating point errors. The use of different hardware
805 may similarly lead to subtly different results.

806

807 In some circumstances, the authors may have used LLMs for data generation or analysis (for
808 instance, extracting data from images or video recordings or extraction of data from literature).
809 As of writing, the use of LLMs in analysis poses a significant problem for reproducibility due to
810 the variability of generated outputs (Fukataki *et al.* 2025, Meyer *et al.* 2025, Staudinger *et al.*
811 2024). The use of setting seed parameters is currently only available for some LLM models
812 (e.g., OpenAI) but even this has been suggested to not guarantee reproducible results
813 (Vadlapati 2023, Morin & Willetts 2020). Therefore, although the code/prompts used can still be
814 archived, the results may not be exactly reproducible. The use of LLMs in this way is similar to
815 the use of proprietary software, in that it causes one part of the analysis pipeline to become
816 unreproducible. This, however, should not affect the reproducibility of results that comes from
817 further analysing the processed data generated by LLMs, which should be archived. The use of
818 LLMs in this context may also be highly computationally expensive, further limiting
819 reproducibility, as we discuss in Stage 5 above.

820

821 If there is no way for the data editor to generate the exact result (e.g., because the software
822 does not allow setting a seed) then the data editor can allow a degree of tolerance for the result
823 which should be noted in their review. Archmiller *et al.* (2020) suggest comparing the conclusion
824 (the direction and significance of results) as well as the numbers of the original and reproduced
825 results. In the first case, if the direction of the effect changes, or the statistical significance

826 changes, then this should be viewed as failing the computational reproducibility test. For results
827 close to zero or the significance threshold, small changes in the results might change direction
828 or significance, respectively. Hardwicke *et al.* (2021) therefore suggested using % error (i.e.
829 $(\text{reproduced} - \text{original}) / \text{original} \times 100$), as this is not dependent on the scale of the results, where
830 0-10% was classified as a minor deviation and >10% as a major deviation, and therefore, not
831 reproducible. However, this % error method (1) still allows for a substantial deviation from the
832 reported values, (2) would result in different tolerances for different effects within the same
833 model, and (3) is most meaningful when effect sizes are on a ratio scale, which typically they
834 are not. Perhaps most importantly, reproduced results should fall well within the reported
835 uncertainty of the original result, and if they do not, this should be viewed as a failure to
836 reproduce the results. The data editor should communicate the conditions under which
837 computational reproducibility was assessed (e.g., the tolerance threshold) in their review. As
838 opposed to in-text results and tables, figures cannot be exactly compared without the use of
839 specialist software, but should be compared by eye for reproducibility.

840

841 The use of computationally expensive methods or proprietary software may mean that the data
842 editor cannot feasibly re-run the analysis in full (see Stage 5 above). If none of the code can be
843 run by the data editor, for example if it all takes place using proprietary software or it involves
844 very computationally expensive analysis, then computational reproducibility cannot be assessed
845 (both Stages 5 and 6 would fail). Clearly, this should not prevent the publication of a manuscript
846 containing such analyses in journals where data editors assess Stages 5 and 6 of these
847 guidelines. In this case, we would therefore recommend that it is highlighted in the manuscript
848 that computational reproducibility could not be assessed (e.g., in the data and code availability
849 statement, or open research sections). If it is only part of the code that cannot be run by the
850 data editor (e.g., a computationally expensive model), then the output of this code should be
851 provided by the author in the archived project and noted in the metadata, so that the output can

852 be compared to the manuscript by the data editor. Given the large variation in data workflows
853 and different proprietary software that may be used, we encourage data editors, editors and
854 journals to be very open to discussion, constructive and flexible in their roles to adopt, or work
855 towards adopting these guidelines.

856

857 **Guideline 6: Results reproduced by the data editor with the archived data and code must**
858 **match those presented in the manuscript. A tolerance threshold can be given when there**
859 **is not an exact match but the authors must state clearly in the code metadata why this**
860 **mismatch might occur. If saved model outputs are instead provided, this must also be**
861 **clearly stated in the metadata.**

862

Table 2. Summary table of the guidelines for each the six Stages of the SORTEE Guidelines for Data and Code Quality Control in Ecology and Evolutionary Biology

Stage	Guidelines
1. Data must be archived and adhere to FAIR guiding principles	
<i>1.1 Data files are accessible and in an open repository</i>	Data are open and freely available and are located in a permanent public repository with an associated globally unique persistent identifier, that is cited in the text and reference list of the manuscript.
<i>1.2. Data are associated with a license</i>	Data must be associated with a license.
<i>1.3. Data files are present and complete</i>	Authors must either provide: a) raw data, along with the processed data and/or code to prepare the data for analysis, or b) a sample of raw data alongside processed/filtered data when full raw data upload is not possible, or c) processed/filtered data with a detailed description of how to both obtain and process/filter the raw data.
<i>1.4. Data files are in an interoperable format</i>	Data files must be provided in an interoperable format.
<i>1.5. Data metadata present and adequate</i>	Detailed metadata, including (but not limited to) a README file, must accompany the data.
2. Archived data corresponds with the data reported in the manuscript	The structure and contents of the archived data files must match the description in the manuscript.
3. Code must be archived and adhere to FAIR guiding principles	
<i>3.1. Code files are accessible and in an open repository</i>	Code files are open and freely available and are located in a permanent public repository with an associated globally unique persistent identifier, preferably in the same archived project as the data files, and are cited in the text and the reference list of the manuscript.
<i>3.2. Code is associated with a licence</i>	The code must be associated with a license.

<i>3.3. Code files are present and complete</i>	All code used for generating the results of the manuscript (including filtering, processing, graphing, and analysis) must be present either in one or more code files.
<i>3.4. Code is in an interoperable format</i>	Code files must be provided in an interoperable format.
<i>3.5. Code metadata present and adequate</i>	A sufficiently detailed README must accompany the code. Code must also be broken into sections with clear annotation stating the purpose of the code with clear links to the relevant sections, figures, and tables in the manuscript.
4. Archived code corresponds with the workflow reported in the manuscript	The structure and content of the archived code must match the description of data filtering, processing, and analysis, and the presentation of results in the manuscript.
5. Archived code runs with the archived data	Code must be able to run without error using the archived data. With the exception of easy to fix file path errors, all errors should be addressed by the author.
6. Results can be computationally reproduced by running the archived code	Results reproduced by the data editor with the archived data and code must match those presented in the manuscript. A tolerance threshold can be given when there is not an exact match but the authors must state clearly in the code metadata why this mismatch might occur. If saved model outputs are instead provided, this must also be clearly stated in the metadata.

864 Suggestions to Authors

865 Data and code quality control is becoming increasingly common across journals in ecology and
866 evolutionary biology. Consequently, authors will have to adhere to certain guidelines for data
867 and code sharing. Although the guidelines presented here are largely aimed at data editors,
868 knowledge of the checks that a data editor is expected to perform will help authors understand
869 what is needed from their data and code prior to submission. It may also increase the likelihood
870 of authors catching their own mistakes because of the checks that will be performed. We hope
871 that the widespread adoption of these guidelines will make the process more transparent for
872 authors and also consistent across journals in the event of manuscript resubmission elsewhere.
873 We acknowledge that making data and code readily accessible and reusable adds to the work
874 load of authors (at least initially). We have several suggestions to ease this process:

875

876 **Adhere to the data and code quality control guidelines from the beginning of the study**

877 Working to make a project accessible and reproducible at the end of a study is a lot of work. We
878 would recommend creating a clear directory structure and creating metadata (e.g., a README)
879 at the beginning of the study, and updating the metadata as new files are added. Similarly,
880 annotating code as authors produce it, not only with section descriptions but also with
881 information about how they run and what output they produce, is far easier than going back and
882 annotating code at the end of the study. Bearing reproducibility in mind while working on a
883 project also makes it far more likely that someone else will be able to reuse the project and
884 successfully run the code. This is even more useful if authors plan on collaborating with multiple
885 people during the study's lifetime. Generally, there exists a multitude of benefits to working
886 reproducibly (Markowetz 2015). We acknowledge that for many authors this may present a
887 steep learning curve, however adherence and knowledge of these guidelines will promote
888 learning and progression over time. Whilst these guidelines are generally aimed at data editors,

889 there are several resources designed for authors, which we would direct authors to, for example
890 the British Ecological Society guides on reproducible code (Cooper & Hsing 2025) and data
891 management (Harrison 2018) and the TADA (Transferable, Available, Documented, Annotated)
892 guidelines (Ivimey-Cook *et al.* 2025), aimed specifically at ecologists and evolutionary biologists.

893

894 **Prepare data and code according to the data and code quality control guidelines**

895 **guidelines before submission to any journal**

896 Inherently linked to the point above, if authors have not prepared data and code according to
897 the data and code quality control guidelines from the start of the study, it is advisable to at least
898 have data and code ready for submission. This will minimise any problems during both the
899 submission process and the data and code quality control, and allow for easy transfer between
900 journals. We have included a summary of the guidelines for data editors that authors may find
901 useful in Table 2.

902

903 **Perform a pre-submission code review**

904 It is advisable for authors to send their project containing data and code to a colleague or co-
905 author for a code review prior to submission to a journal (Ivimey-Cook *et al* 2023). This enables
906 checking whether the code runs with the data in the project structure provided. They can then
907 check whether there is appropriate and adequate metadata, whether data and code match the
908 manuscript, and whether the code reproduces the results in the manuscript. Importantly, co-
909 authors may also be more likely to spot any mistakes in the code as they are familiar with the
910 study and data, and data editors do not check the reliability of code. This could be done within
911 research groups, where the task of code reviewing is shared between members of the team, or
912 as part of a larger 'code club'. Open science organisations, such as SORTEE, have their own
913 code clubs which are open to join. For further advice on setting up code clubs see Ivimey-Cook
914 *et al* (2023).

915

916 **Consider presenting code and associated outputs using Markdown or Quarto**

917 Presenting everything in one self-contained document such as a Markdown or Quarto file can
918 be very helpful for data editors and future readers or users (Buckley *et al* 2025). It allows for a
919 clear link between the code, the data, and the resulting outputs that may need to be assessed.

920 **Suggestions for Journals**

921 **Data and code quality control should start at submission**

922 Currently, in many journals, data and code quality control occurs after (or close to) acceptance.
923 We recommend that data and code are required at submission (see above for methods to
924 anonymise data and code), and that data editors perform a light check of the data, code, and
925 associated metadata (e.g., Stages 1 and 3) early on (before sending to review). This enables
926 reviewers to both see and review the data and code during peer review (if they choose to), and
927 also engages the authors in the data and code quality control process at an early stage. That
928 way, any problems can be highlighted and addressed early in the process. Computational
929 reproducibility checks (Stages 5 and 6) would ideally be conducted later in the process, at a
930 point where the code (particularly related to statistical analysis) is unlikely to change because of
931 further review, to avoid a data editor having to perform these checks multiple times. Any checks
932 need to also be clearly communicated to those not performing the data and code quality control
933 (i.e., editors and reviewers).

934

935 **Ensure journals have data editors with a mixture of coding expertise**

936 There exists a multitude of different languages in which to write code and analyse data.
937 Although R is one of the most popular in Ecology (from 58-80% of studies in ecology and
938 evolution; Lai *et al* 2019, Culina *et al.* 2020, Kambouris *et al.* 2024), code is often written in

939 other languages such as Python, Matlab, SAS, Julia, to name a few. It is therefore important
940 that a journal considers having multiple data editors with varying coding language, data type
941 and area expertise. This means that data editors can be suitably paired to each manuscript.

942

943 **Have clear guidelines on the journal website**

944 Authors will be more likely to adhere to the guidelines adopted by the journal prior to submission
945 if these are clearly displayed on the website, ideally under 'Instructions to Authors' sections.

946 These need to outline what stages of the guidelines the data editors check (e.g., Stages 1-4),
947 what they expect at each stage from the author, and what the authors need to state in their data
948 availability statement. Having data, code, and associated metadata already in a state ready for
949 quality control will reduce much of the work for the data editor. Some journals additionally
950 provide template README files to help authors.

951

952 **Have clear statements within manuscripts**

953 For readers to know what quality control checks have been performed and to highlight the
954 journal's endeavours to ensure the highest quality research, it should be clearly stated within the
955 data and code availability section what checks have been performed. For instance "Data and
956 code were checked from Stage 1-4 of the SORTEE guidelines for Data and Code Quality
957 Control". This statement should also contain information if a check has not been able to be
958 performed, for instance, if the use of proprietary software, sensitive data, or computationally
959 expensive analyses were involved, impeding computational reproducibility tests.

960

961 In psychology and medicine, open science badges have previously been used to indicate
962 manuscripts that adhere to certain open science practices (e.g., open data, open code, open
963 materials, pre-registration) with the ultimate goal of encouraging authors to adopt these
964 practices. Evidence for their effectiveness in increasing data and code sharing is mixed, with

965 early observational studies reporting increases in data sharing after badge implementation
966 (Kidwell *et al.*, 2016), but a subsequent randomized controlled trial finding no such effect in a
967 biomedical journal context (Rowhani-Farid *et al.*, 2020). Note that the journals surveyed in these
968 cases did not have data editors actively checking the data and code archiving.
969 The presence of badges has also been shown to increase the trust of researchers in published
970 articles (Schneider *et al* 2022). Journals could choose to use such badges following data and
971 code quality control to indicate that presence of open data (Stage 1-2) and open code (Stage 3)
972 has been verified, and further badges could be developed for computational reproducibility
973 (Stages 5-6).

974

975 **Have clear definitions and policies of what code and data the journal requires**

976 Ideally, all the data and code used to generate the results should be archived including both raw
977 and processed data and all the code used to process, filter, model, and graph. However, this is
978 at the discretion of the journal and therefore should be made explicit to the authors prior to
979 submission. We recommend that the form of data and code is clearly described in the data
980 availability statement of the manuscript, for instance, “Processed data and code used in
981 modeling and graphing are archived here...”. Again, the journal requirements need to also be
982 clearly communicated to all levels of the journal hierarchy to ensure successful implementation.

983

984 **Conclusion**

985 Here we present a standardised set of guidelines for data and code quality control for journals in
986 ecology and evolutionary biology. As it stands, rates of data and code archiving, and importantly
987 the quality of archived data and code, are low. By recruiting data editors, journals can positively
988 impact the state of open data and code, and in doing so increase research transparency and

989 reproducibility. With the SORTEE data and code quality control guidelines, we propose steps to
990 increase the quality and consistency of data and code quality control across journals that
991 currently have data editors, and provide a template for journals wanting to start data and code
992 quality control. We believe that these guidelines will have substantial benefits for journals, for
993 authors, and for the wider scientific community.

994

995 **Acknowledgments**

996 We thank Bob Montgomerie for extensive discussion and comments on the manuscript. Thanks
997 also to Lars Vilhuber for a useful discussion on the role of data editors in Economics.

998

999 **Author Contributions**

1000 Conceptualisation - JLP and EIC
1001 Writing - Original Draft - JLP and EIC
1002 Writing - Review & Editing - All authors
1003 Visualisation - EIC
1004 Project administration - JLP
1005 Supervision - EIC

1006

1007 **Conflict of Interest**

1008 SORTEE has been financially supported by Dryad, Figshare, the Center of Open Science
1009 (which hosts the Open Science Framework; OSF), Peer Community In, the American Society of

1010 Naturalists and the Royal Society, all of which are mentioned in the guidelines. EIC was the
1011 2025 president of SORTEE. EIC, ML, MP, AST were on SORTEEs board of directors. JLP,
1012 KBN, CJ, SN, and EIC were members of the 2025 and 2026 SORTEE advocacy committee.
1013 JLP, BJA, KBN, JAB, BC, PDA, DG, CJ, RK, ML, SN, ROD, MP, QP, AST, NvD, EIC are
1014 SORTEE members. BB is a data editor at the American Naturalist. EIC, AST, ROD, NvD, MJG,
1015 TD, EF, PDA and QP are data editors at Ecology Letters. JAB, DG, DSM and LW are data
1016 editors at Proceedings B, as was BJA at the initiation of this project. SL is the data editor at
1017 Journal of Evolutionary Biology. EFJ is the data editor from Behavioural Ecology and
1018 Sociobiology. BC, RK, ML, MP, are data editors at PCI. TG is on the Executive board of Peer
1019 Community Journal and president of Peer Community In and BC is the editorial coordinator for
1020 PCI. They did not intervene in the evaluation process made by PCI Ecology.

1021 References

- 1022 Allen, C. & Mehler, D.M.A. 2019. Open science challenges, benefits and tips in early career and
1023 beyond. *PLOS Biol.* **17**: e3000246.
- 1024 Archmiller, A.A., Johnson, A.D., Nolan, J., Edwards, M., Elliott, L.H., Ferguson, J.M., *et al.* 2020.
1025 Computational Reproducibility in The Wildlife Society's Flagship Journals. *J. Wildl. Manag.*
1026 **84**: 1012–1017.
- 1027 Barker, M., Chue Hong, N.P., Katz, D.S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F.,
1028 *et al.* 2022. Introducing the FAIR Principles for research software. *Sci. Data* **9**: 622.
- 1029 Barnes, N. 2010. Publish your computer code: it is good enough. *Nature* **467**: 753–753.
- 1030 Barrett, L. & Montgomerie, R. 2025. A data editor for behavioral ecology. *Behav. Ecol.* **36**:
1031 araf077.
- 1032 Barrett, S.C.H. 2024. Proceedings B 2024: the year in review. *Proc. R. Soc. B Biol. Sci.* **292**:
1033 20250065.

- 1034 Bavota, G. & Russo, B. 2015. Four eyes are better than two: On the impact of code reviews on
1035 software quality. In: *2015 IEEE International Conference on Software Maintenance and*
1036 *Evolution (ICSME)*, pp. 81–90.
- 1037 Belkhir, K., Smadja, C.M., Antoine, P.-O., Scornavacca, C. & Galtier, N. 2025. An overview of
1038 open science in eco-evo research and the publisher effect. *EcoEvoRxiv*.
- 1039 Berberi, I. & Roche, D. 2023. Living database of journal data policies in E&E. , doi:
1040 10.17605/OSF.IO/D6SP3. OSF.
- 1041 Berberi, I. & Roche, D.G. 2022. No evidence that mandatory open data policies increase error
1042 correction. *Nat. Ecol. Evol.* **6**: 1630–1633.
- 1043 Bolnick, D. & Paull, J. 2016. Retraction: Morphological and dietary differences between
1044 individuals are weakly but positively correlated within a population of threespine stickleback.
1045 *Evol. Ecol. Res.* **17**: 849.
- 1046 Buckley, Y.M., Bardgett, R., Gordon, R., Iler, A., Mariotte, P., Ponton, S., *et al.* 2025. Using
1047 dynamic documents to mend cracks in the reproducible research pipeline. *J. Ecol.* **113**:
1048 270–274.
- 1049 Butler, D.G., Cullis, B.R., Gilmour, A.R., Gogel, B.G. & Thompson, R. 2017. ASReml-R
1050 Reference Manual. VSN International Ltd, Hemel Hempstead, HP1 1ES, UK.
- 1051 Carroll, S.R., Herczog, E., Hudson, M., Russell, K. & Stall, S. 2021. Operationalizing the CARE
1052 and FAIR Principles for Indigenous data futures. *Sci. Data* **8**: 108.
- 1053 Chapman, A.D. 2020. Current Best Practices for Generalizing Sensitive Species Occurrence
1054 Data.
- 1055 Christensen, G., Dafoe, A., Miguel, E., Moore, D.A. & Rose, A.K. 2019. A study of the impact of
1056 data sharing on article citations using journal policies as a natural experiment. *PLOS ONE*
1057 **14**: e0225883.
- 1058 Cologna, V., Mede, N.G., Berger, S., Besley, J., Brick, C., Joubert, M., *et al.* 2025. Trust in
1059 scientists and their role in society across 68 countries. *Nat. Hum. Behav.* **9**: 713–730.

- 1060 Cooper, N., Clark, A.T., Lecomte, N., Qiao, H. & Ellison, A.M. 2024. Harnessing large language
1061 models for coding, teaching and inclusion to empower research in ecology and evolution.
1062 *Methods Ecol. Evol.* **15**: 1757–1763.
- 1063 Cooper, N. & Hsing, P.-Y. 2025. Guide to Reproducible Code. British Ecological Society.
1064 <https://doi.org/10.5281/zenodo.16421733>
- 1065 Culina, A., Berg, I. van den, Evans, S. & Sánchez-Tójar, A. 2020. Low availability of code in
1066 ecology: A call for urgent action. *PLOS Biol.* **18**: e3000763.
- 1067 Evans, S.R. 2016. Gauging the Purported Costs of Public Data Archiving for Long-Term
1068 Population Studies. *PLOS Biol.* **14**: e1002432.
- 1069 Feng, X., Qiao, H. & Enquist, B.J. 2020. Doubling demands in programming skills call for
1070 ecoinformatics education. *Front. Ecol. Environ.* **18**: 123–124.
- 1071 Fernández-Juricic, E. 2021. Why sharing data and code during peer review can enhance
1072 behavioral ecology research. *Behav. Ecol. Sociobiol.* **75**: 103.
- 1073 Fukataki, Y., Hayashi, W., Nishimoto, N. & Ito, Y.M. 2025. Developing artificial intelligence tools
1074 for institutional review board pre-review: A pilot study on ChatGPT's accuracy and
1075 reproducibility. *PLOS Digit Health* 4(6): e0000695.
1076 <https://doi.org/10.1371/journal.pdig.0000695>
- 1077 Gihawi, A., Ge, Y., Lu, J., Puiu, D., Xu, A., Cooper, C.S., *et al.* 2023. Major data analysis errors
1078 invalidate cancer microbiome findings. *mBio* **14**: e01607-23.
- 1079 Goldacre, B., Morton, C.E. & DeVito, N.J. 2019. Why researchers should share their analytic
1080 code. *BMJ* **367**: l6365.
- 1081 Gomes, D.G.E., Pottier, P., Crystal-Ornelas, R., Hudgins, E.J., Foroughirad, V., Sánchez-
1082 Reyes, L.L., *et al.* 2022. Why don't we share data and code? Perceived barriers and
1083 benefits to public archiving practices. *Proc. R. Soc. B Biol. Sci.* **289**: 20221113.
- 1084 Gould, E., Fraser, H.S., Parker, T.H., Nakagawa, S., Griffith, S.C., Vesk, P.A., *et al.* 2025. Same
1085 data, different analysts: variation in effect sizes due to analytical decisions in ecology and
1086 evolutionary biology. *BMC Biol.* **23**: 35.

- 1087 Grant, S., Corker, K., Mellor, D., Stewart, S., Cashin, A., Lagisz, M., *et al.* 2025. TOP 2025: An
1088 Update to the Transparency and Openness Promotion Guidelines. OSF.
- 1089 Hardwicke, T.E., Bohn, M., MacDonald, K., Hembacher, E., Nuijten, M.B., Peloquin, B.N., *et al.*
1090 2021. Analytic reproducibility in articles receiving open data badges at the journal
1091 *Psychological Science: an observational study. R. Soc. Open Sci.* **8**: 201494.
- 1092 Hardwicke, T.E., Mathur, M.B., MacDonald, K., Nilsonne, G., Banks, G.C., Kidwell, M.C., *et al.*
1093 2018. Data availability, reusability, and analytic reproducibility: evaluating the impact of a
1094 mandatory open data policy at the journal *Cognition. R. Soc. Open Sci.* **5**: 180448.
- 1095 Harrison, K. 2018. Guide to Data Management. British Ecological Society
- 1096 Harvard Longwood Medical Area Research Data Management Working Group. 2023. Harvard
1097 Biomedical Repository Matrix. Zenodo. <https://doi.org/10.5281/zenodo.10651775>
- 1098 Henderson, A.S., Hickson, R.I., Furlong, M., McBryde, E.S. & Meehan, M.T. 2024.
1099 Reproducibility of COVID-era infectious disease models. *Epidemics* **46**: 100743.
- 1100 Hennessy, E.A., Acabchuk, R.L., Arnold, P.A., Dunn, A.G., Foo, Y.Z., Johnson, B.T., *et al.* 2022.
1101 Ensuring Prevention Science Research is Synthesis-Ready for Immediate and Lasting
1102 Scientific Impact. *Prev. Sci.* **23**: 809–820.
- 1103 Hillemann, F. [freddy], Burant, J.B., Culina, A. & Vriend, S.J.G. 2025. Code review in practice: A
1104 checklist for computational reproducibility and collaborative research in ecology and
1105 evolution. *EcoEvoRxiv*.
- 1106 Ivimey-Cook, E.R., Pick, J.L., Bairos-Novak, K.R., Culina, A., Gould, E., Grainger, M., *et al.*
1107 2023. Implementing code review in the scientific workflow: Insights from ecology and
1108 evolutionary biology. *J. Evol. Biol.* **36**: 1347–1356.
- 1109 Ivimey-Cook, E.R., Sánchez-Tójar, A., Berberi, I., Culina, A., Roche, D.G., Almeida, R.A., *et al.*
1110 2025. From Policy to Practice: Progress towards Data- and Code-Sharing in Ecology and
1111 Evolution. *Proc. R. Soc. B Biol. Sci* **292** 20251394.
- 1112 Ivimey-Cook, E.R., Culina, A., Dimri, S., Lagisz, M., Moran, N., Nakagawa, S., *et al.* 2025.
1113 TADA! Simple guidelines to improve analytical code sharing for transparency and
1114 reproducibility. *EcoEvoRxiv*, <https://doi.org/10.32942/X2D93K>

- 1115
- 1116 Janssens, M., Gaillard, S., Haan, J.J. de, Leeuw, W. de, Brooke, M., Burke, M., *et al.* 2023. How
1117 open science can support the 3Rs and improve animal research. *Res. Ideas Outcomes* **9**:
1118 e105198.
- 1119 Kambouris, S., Wilkinson, D.P., Smith, E.T. & Fidler, F. 2024. Computationally reproducing
1120 results from meta-analyses in ecology and evolutionary biology using shared code and
1121 data. *PLOS ONE* **19**: e0300333.
- 1122 Kellner, K.F., Doser, J.W. & Belant, J.L. 2025. Functional R code is rare in species distribution
1123 and abundance papers. *Ecology* **106**: e4475.
- 1124 Kidwell, M.C., Lazarević, L.B., Baranski, E., Hardwicke, T.E., Piechowski, S., Falkenberg, L.-S.,
1125 *et al.* 2016. Badges to Acknowledge Open Practices: A Simple, Low-Cost, Effective Method
1126 for Increasing Transparency. *PLOS Biol.* **14**: e1002456.
- 1127 Kim, B., Moran, N.P., Reinhold, K. & Sánchez-Tójar, A. 2021. Male size and reproductive
1128 performance in three species of livebearing fishes (*Gambusia* spp.): A systematic review
1129 and meta-analysis. *J. Anim. Ecol.* **90**: 2431–2445.
- 1130 Kimmel, K., Avolio, M.L. & Ferraro, P.J. 2023. Empirical evidence of widespread exaggeration
1131 bias and selective reporting in ecology. *Nat. Ecol. Evol.* **7**: 1525–1536.
- 1132 Kohrs, F.E., Auer, S., Bannach-Brown, A., Fiedler, S., Haven, T.L., Heise, V., *et al.* 2023.
1133 Eleven strategies for making reproducible research and open science training the norm at
1134 research institutions. *eLife* **12**: e89736.
- 1135 König, L., Gärtner, A., Slack, H., Dhakal, S., Adetula, A., Dougherty, M., *et al.* 2025. How to
1136 bolster employability through open science. OSF.
- 1137 Lai, J., Lortie, C.J., Muenchen, R.A., Yang, J. & Ma, K. 2019. Evaluating the popularity of R in
1138 ecology. *Ecosphere* **10**: e02567.
- 1139 Maitner, B., Santos Andrade, P.E., Lei, L., Kass, J., Owens, H.L., Barbosa, G.C.G., *et al.* 2024.
1140 Code sharing in ecology and evolution increases citation rates but remains uncommon.
1141 *Ecol. Evol.* **14**: e70030.

- 1142 Mandhane, P.J. 2024. Notice of Retraction: Hahn LM, et al. Post–COVID-19 Condition in
1143 Children. *JAMA Pediatrics*. 2023;177(11):1226-1228. *JAMA Pediatr.* **178**: 1085–1086.
- 1144 Manzanedo, R.D., HilleRisLambers, J., Rademacher, T.T. & Pederson, N. 2021. Retraction
1145 Note: Evidence of unprecedented rise in growth synchrony from global tree ring records.
1146 *Nat. Ecol. Evol.* **5**: 1047–1047.
- 1147 Markowetz, F. 2015. Five selfish reasons to work reproducibly. *Genome Biol.* **16**: 274.
- 1148 Meyer, A., Schömig, E. & Streichert, T. 2025. ChatGPT and reference intervals: a comparative
1149 analysis of repeatability in GPT-3.5 Turbo, GPT-4, and GPT-4o. *Front. Artif. Intell.*
1150 **8**:1681979. <https://doi.org/10.3389/frai.2025.1681979>
- 1151 Mills, J.A., Teplitsky, C., Arroyo, B., Charmantier, A., Becker, Peter.H., Birkhead, T.R., et al.
1152 2015. Archiving Primary Data: Solutions for Long-Term Studies. *Trends Ecol. Evol.* **30**:
1153 581–589.
- 1154 Minocher, R., Atmaca, S., Bavero, C., McElreath, R. & Beheim, B. 2021. Estimating the
1155 reproducibility of social learning research published between 1955 and 2018. *R. Soc. Open*
1156 *Sci.* **8**: 210450.
- 1157 Mislán, K. a. S., Heer, J.M. & White, E.P. 2016. Elevating The Status of Code in Ecology.
1158 *Trends Ecol. Evol.* **31**: 4–7.
- 1159 Molloy, J.C. 2011. The Open Knowledge Foundation: Open Data Means Better Science. *PLOS*
1160 *Biol.* **9**: e1001195.
- 1161 Morin, M. & Willetts, M. 2020. Non-Determinism in TensorFlow ResNets. arXiv.
1162 <https://doi.org/10.1145/3673791.3698432>
- 1163 Müller, K. & Bryan, J. 2020. here: A Simpler Way to Find Your Files.
- 1164 National Academies of Sciences, E., Affairs, P. and G., Committee on Science, E., Information,
1165 B. on R.D. and, Sciences, D. on E. and P., Statistics, C. on A. and T., et al. 2019.
1166 Reproducibility. In: *Reproducibility and Replicability in Science*. National Academies Press
1167 (US).

- 1168 Parr, C.S. & Cummings, M.P. 2005. Data sharing in ecology and evolution. *Trends Ecol. Evol.*
1169 **20**: 362–363.
- 1170 Piwowar, H.A. & Chapman, W.W. 2010. Public sharing of research datasets: A pilot study of
1171 associations. *J. Informetr.* **4**: 148–156.
- 1172 Piwowar, H.A., Day, R.S. & Fridsma, D.B. 2007. Sharing Detailed Research Data Is Associated
1173 with Increased Citation Rate. *PLOS ONE* **2**: e308.
- 1174 Powers, S.M. & Hampton, S.E. 2019. Open science, reproducibility, and transparency in
1175 ecology. *Ecol. Appl.* **29**: e01822.
- 1176 Purgar, M., Klanjscek, T. & Culina, A. 2022. Quantifying research waste in ecology. *Nat. Ecol.*
1177 *Evol.* **6**: 1390–1397.
- 1178 R Core Team. 2022. R: A Language and Environment for Statistical Computing. R Foundation
1179 for Statistical Computing, Vienna, Austria.
- 1180 Reinecke, R., Trautmann, T., Wagener, T. & Schüler, K. 2022. The critical need to foster
1181 computational reproducibility. *Environ. Res. Lett.* **17**: 041005.
- 1182 Resnik, D.B. & Hosseini, M. 2025. Disclosing artificial intelligence use in scientific research and
1183 publication: When should disclosure be mandatory, optional, or unnecessary?
1184 *Accountability in Research*, 1–13. <https://doi.org/10.1080/08989621.2025.2481949>
- 1185 Roche, D.G., Berberi, I., Dhane, F., Lauzon, F., Soeharjono, S., Dakin, R., *et al.* 2022. Slow
1186 improvement to the archiving quality of open datasets shared by researchers in ecology
1187 and evolution. *Proc. R. Soc. B Biol. Sci.* **289**: 20212780.
- 1188 Roche, D.G., Kruuk, L.E.B., Lanfear, R. & Binning, S.A. 2015. Public Data Archiving in Ecology
1189 and Evolution: How Well Are We Doing? *PLOS Biol.* **13**: e1002295.
- 1190 Rowhani-Farid, A., Aldcroft, A. & Barnett, A.G. 2020. Did awarding badges increase data
1191 sharing in BMJ Open? A randomized controlled trial. *R. Soc. Open Sci.* **7**: 191818.
- 1192 Sánchez-Tójar, A., Bezine, A., Purgar, M. & Culina, A. 2025. Code-sharing policies are
1193 associated with increased reproducibility potential of ecological findings. *Peer Community J.*
1194 **5**.

- 1195 Schneider, J., Rosman, T., Kelava, A. & Merk, S. 2022. Do Open-Science Badges Increase
1196 Trust in Scientists Among Undergraduates, Scientists, and the Public? *Psychol. Sci.* **33**:
1197 1588–1604.
- 1198 Soeharjono, S. & Roche, D.G. 2021. Reported Individual Costs and Benefits of Sharing Open
1199 Data among Canadian Academic Faculty in Ecology and Evolution. *BioScience* **71**: 750–
1200 756.
- 1201 SORTEE. 2026. SORTEE 2025 Annual Report Accessed 2026-01-13 <https://osf.io/gsw6x>
- 1202 Staudinger, M., Kusa, W., Piroi, F., Lipani A. & Hanbury, A. 2024 A Reproducibility and
1203 Generalizability Study of Large Language Models for Query Generation. *Proceedings of the*
1204 *2024 Annual International ACM SIGIR Conference on Research and Development in*
1205 *Information Retrieval in the Asia Pacific Region*. 186 - 196
1206 <https://doi.org/10.1145/3673791.3698432>
- 1207 Tedersoo, L., Küngas, R., Oras, E., Köster, K., Eenmaa, H., Leijen, Ä., *et al.* 2021. Data sharing
1208 practices and data availability upon request differ across scientific disciplines. *Sci. Data* **8**:
1209 192.
- 1210 Thrall, P.H., Chase, J., Drake, J., Espuno, N., Hello, S., Ezenwa, V., *et al.* 2023. From raw data
1211 to publication: Introducing data editing at Ecology Letters. *Ecol. Lett.* **26**: 829–830.
- 1212 Touchon, J.C. & McCoy, M.W. 2016. The mismatch between current statistical practice and
1213 doctoral training in ecology. *Ecosphere* **7**: e01394.
- 1214 Trisovic, A., Lau, M.K., Pasquier, T. & Crosas, M. 2022. A large-scale study on research code
1215 quality and execution. *Sci. Data* **9**: 60.
- 1216 Vadlapati, P. 2023. Does Seed Matter?: Investigating the Effect of Random Seeds on LLM
1217 Accuracy. *International Journal on Science and Technology*, 14, 1-5.
1218 <https://doi.org/10.5281/zenodo.14288248>
- 1219 Vazire, S. 2017. Quality Uncertainty Erodes Trust in Science. *Collabra Psychol.* **3**: 1.
- 1220 Viglione, G. 2020. ‘Avalanche’ of spider-paper retractions shakes behavioural-ecology
1221 community. *Nature* **578**: 199–200.

- 1222 Vines, T.H., Andrew, R.L., Bock, D.G., Franklin, M.T., Gilbert, K.J., Kane, N.C., *et al.* 2013.
1223 Mandated data archiving greatly improves access to research data. *FASEB J.* **27**: 1304–
1224 1308.
- 1225 Weissgerber, T.L., Gazda, M.A., Nilsson, G., ter Riet, G., Cobey, K.D., Prieß-Buchheit, J., *et*
1226 *al.* 2024. Understanding the provenance and quality of methods is essential for responsible
1227 reuse of FAIR data. *Nat. Med.* **30**: 1220–1221.
- 1228 Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., *et al.*
1229 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Sci.*
1230 *Data* **3**: 160018.
- 1231
- 1232