






ORIGINAL ARTICLE OPEN ACCESS

Sharing Is Caring? International Society for Pharmacoepidemiology Review and Recommendations for Sharing Programming Code

John Tazare¹  | Shirley V. Wang²  | Rosa Gini³  | Daniel Prieto-Alhambra^{4,5}  | Peter Arlett⁶ | Daniel R. Morales Leaver^{6,7} | Caroline Morton⁸ | John Logie⁹ | Jennifer Popovic¹⁰ | Katherine Donegan¹¹ | Sebastian Schneeweiss²  | Ian Douglas¹ | Anna Schultze¹

¹Faculty of Epidemiology and Population Health, London School of Hygiene & Tropical Medicine, London, UK | ²Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts, USA | ³Agenzia Regionale di Sanità della Toscana, Florence, Italy | ⁴Pharmaco- and Device Epidemiology, Botnar Research Centre, NDORMS, University of Oxford, Oxford, UK | ⁵Data Analytics and Methods Taskforce, Department of Medical Informatics, Erasmus MC, Rotterdam, Netherlands | ⁶European Medicines Agency, Amsterdam, Netherlands | ⁷Division of Population Health and Genomics, University of Dundee, Dundee, UK | ⁸Queen Mary University of London, London, UK | ⁹GlaxoSmithKline, Brentford, UK | ¹⁰GlaxoSmithKline, Waltham, Massachusetts, United States | ¹¹UK Medicines and Healthcare Products Regulatory Agency, London, UK

Correspondence: Anna Schultze (anna.schultze@lshtm.ac.uk)

Received: 1 February 2024 | **Revised:** 6 May 2024 | **Accepted:** 6 June 2024

Funding: This guidance report was supported by a manuscript proposal grant from the International Society for Pharmacoepidemiology (ISPE).

Keywords: open science | pharmacoepidemiology | programming code sharing | reproducibility | transparency

ABSTRACT

Purpose: There is increasing recognition of the importance of transparency and reproducibility in scientific research. This study aimed to quantify the extent to which programming code is publicly shared in pharmacoepidemiology, and to develop a set of recommendations on this topic.

Methods: We conducted a literature review identifying all studies published in *Pharmacoepidemiology and Drug Safety* (PDS) between 2017 and 2022. Data were extracted on the frequency and types of programming code shared, and other key open science practices (clinical codelist sharing, data sharing, study preregistration, and stated use of reporting guidelines and preprinting). We developed six recommendations for investigators who choose to share code and gathered feedback from members of the International Society for Pharmacoepidemiology (ISPE).

Results: Programming code sharing by articles published in PDS ranged from 1.8% in 2017 to 9.5% in 2022. It was more prevalent among articles with a methodological focus, simulation studies, and papers which also shared record-level data.

Conclusion: Programming code sharing is rare but increasing in pharmacoepidemiology studies published in PDS. We recommend improved reporting of whether code is shared and how available code can be accessed. When sharing programming code, we recommend the use of permanent digital identifiers, appropriate licenses, and, where possible, adherence to good software practices around the provision of metadata and documentation, computational reproducibility, and data privacy.

Prior postings and presentations: Preliminary results from the literature review and initial recommendations were presented as a symposium at the International Society for Pharmacoepidemiology (ISPE)'s Annual Meeting, August 2023 in Halifax.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *Pharmacoepidemiology and Drug Safety* published by John Wiley & Sons Ltd.

Summary

- This study aimed to quantify current trends of programming code sharing in pharmacoepidemiology research and provide recommendations on this topic.
- Out of 968 eligible “Pharmacoepidemiology and Drug Safety” articles between 2017 and 2022, 4.8% ($N=46$) shared programming code. Code sharing increased from 1.8% in 2017 to 9.5% in 2022, with higher prevalence in methodological and simulation studies.
- Standardized reporting of open science practices, use of permanent identifiers, proper licensing, and adherence to good software practices are recommended when sharing programming code.
- This study suggests a growing trend in programming code sharing in pharmacoepidemiology, emphasizing the need for consistent reporting of whether and how code can be accessed, as well as adherence to good programming practices for increased utility.

1 | Introduction

The past decade has seen an increased focus on the transparency, replicability, and reproducibility of scientific research. This is, at least in part, a result of the so-called “replication crisis,” and the failure of many published findings to be independently replicated in many disciplines [1, 2]. Within pharmacoepidemiology, the REPEAT initiative recently evaluated the independent reproducibility of 150 studies that used electronic healthcare databases. Wang, Sreedhara, and Schneeweiss found that the majority of the results could be closely reproduced with a correlation coefficient of 0.84 for the original and reproduced measures of effect, but that key study parameters necessary for reproducibility were often incompletely reported leading to inexplicable discrepancies [3].

Open science practices, an umbrella term covering practices such as study preregistration, protocol sharing, data sharing, and programming code sharing, have been suggested as a possible means of improving the reliability and integrity of scientific research [1, 4–6]. Patient-level data sharing is challenging given privacy protection regulations for studies using real-world data or data prospectively collected for specific purposes. However, code sharing, where programming code written to process and analyze research data are made public, is often feasible. Researchers may choose to share study code to facilitate direct computational reproducibility, to enable re-use of the code, and/or because it increases the transparency of a study and therefore engenders trust in the findings. These are not all necessarily true at once, for example, code could enable direct computational reproducibility without meaningfully increasing transparency; and code sharing in and of itself is not enough to guarantee computational reproducibility [7].

Advocates of code sharing point to a range of proposed benefits including enabling the detection of potential coding errors, promoting consistency and efficiency across studies, encouraging the adoption of best-practice software engineering tools (such as version control) as well as promoting a more rapid and widespread

use of novel analytical strategies [4, 8–11]. However, concerns have been raised that poorly commented or unwieldy code might not help to assess whether a study protocol was implemented as intended, and that a focus on code sharing could result in a counterproductive, reduced focus on natural language or graphical reporting of methods [12, 13]. An ISPE/ISPOR joint task force on the reporting of database studies concluded that the full reporting of all study parameters is at least as important, and of higher utility for decision makers, than sharing raw programming code [13].

Despite open science practices in biomedical research receiving increasing attention by funders [14–16], journals [17], and in independent reviews [18], there has been little research on the extent of code sharing in either applied or methodological pharmacoepidemiology, and existing reporting guidelines do not include any recommendations on how to best share code [13, 19, 20]. This is particularly important given the vast amounts of code necessary to process, manage and analyze the large existing datasets used throughout the field of pharmacoepidemiology. Our objectives were to address this gap by quantifying the extent of, and trends in, code sharing in pharmacoepidemiology publications through a targeted literature review of studies published in the journal *Pharmacoepidemiology and Drug Safety* and to provide recommendations to support effective code sharing among pharmacoepidemiologists. This manuscript has been endorsed by the International Society for Pharmacoepidemiology (ISPE).

2 | Methods

To capture the views of stakeholders from across the field of pharmacoepidemiology, a working group of representatives from academic research centers, medicines regulatory authorities, and the pharmaceutical industry (summarized in Table S1) contributed to the planning and conduct of all elements of the project.

2.1 | Literature Review

First, we conducted a literature review describe code sharing in pharmacoepidemiology. The protocol for this was not formally preregistered, but a time-stamped version was uploaded to Github on May 16, 2023, prior to downloading publications on May 17, 2023, and initiating data extraction on June 15, 2023. The time-stamped version is available at <https://github.com/ehr-lshtm/code-sharing/tree/main/docs>.

2.2 | Search Strategy

We automatically downloaded all articles with a publication issue date in *Pharmacoepidemiology and Drug Safety* (PDS) between January 1, 2017 and December 31, 2022 using the pubmedR R package on May 17, 2023 <https://cran.r-project.org/web/packages/pubmedR/index.html>. The time period was chosen to provide a balance between having a reasonable number of years over which time trends could be assessed, and resource constraints given an increasing number of papers. The review was restricted to PDS as this offered a pragmatic means of identifying pharmacoepidemiology articles.

Commentaries, abstract only (i.e., conference abstracts), and letters to the editor were excluded, as were articles with no analysis of data (real or simulated). The text search string is provided in Table S2.

2.3 | Data Screening of Potentially Eligible Articles

Each of the articles identified was screened for eligibility using a two-step process split between two reviewers (A.S. and J.T.), with each article screened by a single reviewer. First, we performed abstract screening to exclude non-eligible articles. Second, we performed full-text screening, assessing whether code sharing was applicable by considering whether the article involved analysis of real or simulated data, or a description of an algorithm or method for which programming code was required. Where there was uncertainty in paper eligibility from abstract screening a joint decision was taken on whether that article should be included.

2.4 | Data Extraction From Eligible Studies

For all eligible studies, basic article information including publication year, author list and permanent digital identifier (DOI), was extracted using the pubmedR. Further data extraction was conducted manually by a single reviewer (A.S. or J.T.) and focused on two outcomes, code sharing and other open science indicators, summarized in Table S3. Our primary outcome assessed whether the article had shared programming code. We defined this as some or all of the code used to process and/or analyze the data being available without requiring contact with the corresponding author. Any generic reference to use of an open-source package (e.g., “analyses were performed using tidyverse and survival packages in R”) was not considered code sharing unless the authors developed the package as part of the study. For instances of broken links where minor and straight-forward modifications to the URL allowed us to access code without further contacting the authors, we classified this as code sharing (although these instances were very rare). Where programming code was shared, we extracted information on its location, whether there were instructions (from none, through to basic commenting of the scripts, detailed instructions (e.g., in a README file) and published packages with full documentation), and/or real or synthetic data provided to run the code, the language or platform used, and the conditions under which code shared can be reused. This part of the review included extracting information from external sources, such as Github. In addition to features relating to programming code, we extracted information on article characteristics for all articles such as the affiliation (pharmaceutical industry vs. not), funding (industry vs. not) and the article research aim (grouped together as methodological vs. applied, where “applied” included descriptive analysis, comparative effectiveness and safety studies, validation studies and reviews).

Additionally, we extracted information on other open science indicators, including author-reported preregistration, preprinting or adherence to any named reporting guideline, and the provision of data (real or simulated) or clinical codelists (e.g., lists of disease, exposure or procedure codes used to identify patient attribute of interest, such as International Classification of Diseases 10th Revision [ICD-10] or National Drug [NDC]

codes) for any study parameter. The final data extraction form is provided in Table S4; this was piloted on a sample of articles ($N=20$). Information was extracted from both the main text and supporting information of eligible articles.

2.5 | Data Analysis

We performed descriptive analyses of the overall prevalence and trends over calendar time of code sharing and open science indicators in the eligible articles. We also described the prevalence of code sharing according to article characteristics. Data management and analysis was performed using R [version 4.3.1]. All data and programming code is shared online under an MIT open license at: <https://doi.org/10.5281/zenodo.13152263>.

2.6 | Development of Recommendations

We drafted a set of recommendations to address gaps in the reporting of code sharing identified in the literature review as follows: draft recommendations were developed by two authors (J.T. and A.S.) based on findings of the review and circulated to the working group. This was discussed during an in-person meeting in Halifax, NS, Canada in August 2023 and consequently refined. Further feedback was sought from stakeholders during a series of semi-structured interviews conducted during the autumn of 2023 (where permitted, interviewees are included in the Acknowledgements section) and through engagement with the ISPE society as described below.

2.7 | Society Engagement

We presented our findings and draft recommendations at a symposium at ICPE, Halifax 2023 with opportunities for audience participation and feedback, and all current ISPE members were invited to review this manuscript as part of the ISPE review process for funded manuscripts.

3 | Results

3.1 | Literature Review

The database search identified 1136 articles published in *PDS* between 2017 and 2022, of which 968 met the inclusion criteria after full-text screening (Figure S1). A dataset including all eligible articles and extracted information is available, under an MIT open license, at <https://doi.org/10.5281/zenodo.13152263>.

3.2 | Characteristics of Studies

Table 1 summarizes the characteristics of eligible articles. Most were “Original Articles” (90.8%), with 85.3% applied and 14.7% methodological in focus.

Of the eligible articles, 71.0% reported the programming language(s) used, with many reporting multiple languages. Of the 687 articles reporting this information, the key languages

TABLE 1 | Characteristics of included papers and of those which shared code.

Characteristic		N (%)
Total		968 (100)
Year		
	2017	168 (17.4)
	2018	163 (16.8)
	2019	180 (18.6)
	2020	166 (17.1)
	2021	164 (16.9)
	2022	127 (13.1)
Publication type (PDS) ^a		
	Brief Report	62 (6.4)
	Original Article	879 (90.8)
	Review	27 (2.8)
Article aim ^a		
	Applied	826 (85.3)
	Methodological	142 (14.7)
Simulation study	Yes	37 (3.8)
COVID-19 research	Yes	17 (1.8)
Pharmaceutical industry affiliation	Yes	197 (20.4)
Industry funding reported	Yes	140 (14.5)
Reported programming language	Yes	687 (71.0)
Type of programming language ^b		
	SAS	307 (44.6)
	R	167 (24.3)
	Stata	162 (23.6)
	SPSS/Excel	95 (13.8)
Shared code (Total)		46 (100)
Instructions to run code present	Yes in comments	17 (37.0)
	Yes, there is a README with instruction	13 (28.3)
	Yes, this is a published package with documentation	9 (19.6)
	No	9 (19.6)

(Continues)

TABLE 1 | (Continued)

Characteristic		N (%)
Programming code coverage	Statistical analysis	24 (52.2)
	Methods	10 (21.7)
	Illustrative code snippet	13 (28.3)
	Data management	18 (39.1)
Synthetic data provided	Yes	9 (19.6)
License for code	Yes	9 (19.6)

^aPDS article type refers to formal categories used by PDS. Article aim was assessed by data extractors, with the goal of differentiating applied and methodological research. Not all review articles were published as a PDS review (some were original research), which explains the discrepancies between these categories.

^bMany articles used more than one programming language, and the categories reported are not exclusive. A full list of each individual programming language used is provided in the Supporting Information S1.

were SAS (44.5%), R (24.3%), and Stata (23.6%). Less common analysis tools consisted of platforms (e.g., Aetion or Palantir Foundry) and other scripted languages (e.g., Python). A complete list of the languages used can be found in the Supporting Information Table S8.

3.3 | Programming Code Sharing

Overall, 62 (6.4%) papers acknowledged code sharing in either the main text or within the supporting information. Of those, 46 (74.2%) had programming code which could be accessed without contacting the authors, reflecting 4.8% of the total number of eligible articles. The characteristics of these articles is provided in Table 1, and the prevalence of code sharing stratified by article characteristics is presented in Table 2. Code sharing was more common in simulation studies (40.5%) and methodological articles (20.4%). There were only small differences in the prevalence of code sharing according to whether the study reported industry funding (2.9%), or consisted of COVID-19 related research (11.8%).

Figure 1 describes where code was shared, with supporting information documents (50.0%) or Github (32.6%) being the most common, although this was not always well signposted. Nine studies (19.6%) provided synthetic data, and 9 (19.6%) provided a license detailing how the programming code shared could be reused (Table 3). Of the 9 papers that provided licensed code, the most commonly provided license was GPL v3.0 (<https://www.gnu.org/licenses/gpl-3.0.txt>) (N = 5).

Across the period from 2017 and 2022, code sharing increased from 1.8% to 9.5% (Figure 2 and Table S5).

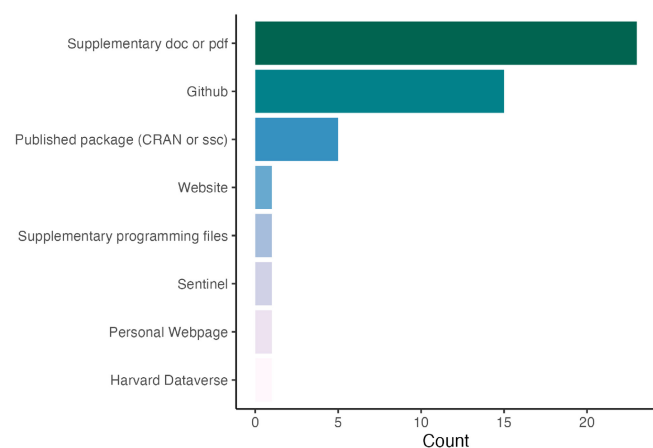
3.4 | Other Open Research Practice Indicators

Overall, as presented in Table 3, we observed low reporting of data sharing, preregistration, and adherence to reporting guidelines (adherence to specific reporting guidelines are presented

TABLE 2 | Code sharing prevalence according to article characteristics.

Article characteristic		Total (N)	Shared code (n, %)
Article type	Applied	826	17 (2.1)
	Methodological	142	29 (20.4)
Simulation study	Yes	37	15 (40.5)
COVID-19 research	Yes	17	2 (11.8)
Industry funding	Yes	140	4 (2.9)
Programming language ^a	R	167	23 (13.8)
	Stata	162	11 (6.8)
	SAS	307	12 (3.9)

^aThere is overlap with some studies using more than one programming language.

**FIGURE 1** | Location of code, when shared.

in Figure S2). However, in studies where codelists were used, at least one list was shared in 73.7% of articles. Between 2017 and 2022, whilst we observed an increase in the proportion of articles reporting adherence to a reporting guideline and data sharing, the proportion of sharing codelists remained consistently high (Figure 3, Table S6).

The prevalence of code sharing according to other open science practices is presented in Table S7. This showed that data sharing was more common in articles that also shared programming code, compared to those that did not (6.5% vs. 2.0%) (Table S7).

4 | Discussion

Of 968 articles published in *PDS* which involved analysis of data between 2017 and 2022, just 46 (4.8%) shared programming

code. Code sharing was reasonably stable between 2018 and 2021 but increased to just over 9.5% in 2022. Other open research practices were uncommon, apart from the publication of clinical codelists, with almost 3/4 of eligible articles sharing codelists for at least one study parameter. Our work highlights that there is currently no consensus around whether or how to share code in pharmacoepidemiology research.

4.1 | Considerations for Code Sharing in Pharmacoepidemiology

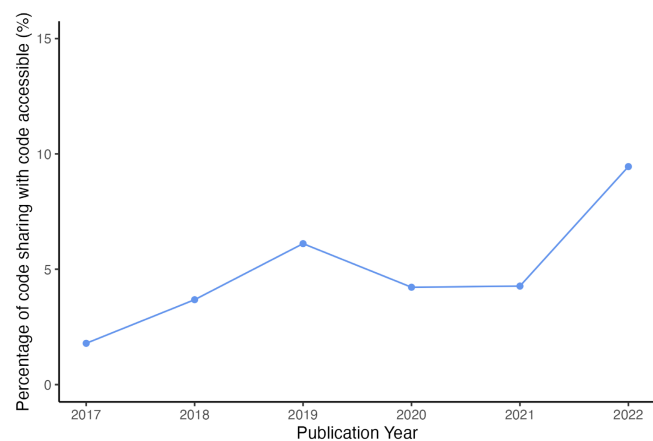
The advantages and disadvantages of code sharing have been widely discussed across the scientific literature [4, 9, 12] (Table 4). During the conduct of this work, two considerations emerged as particularly important potential barriers to code sharing in pharmacoepidemiology: the utility of making code available given the complexity of most study protocols, and concerns around IP or the potential loss of a commercial advantage. Regarding the first concern, it is important to emphasize that code sharing should not replace clear textual or graphical reporting of the study design and implementation, and researchers who share code should still adhere to best practice reporting guidelines [19, 20]. In conjunction with a detailed protocol, clinical codelists, documentation and data (where possible), code sharing can serve to enhance transparency; in isolation it may not meaningfully do so—particularly if the programs shared are large and complex to navigate. In our review, we found a consistently higher prevalence of code sharing for simulation studies and papers with a methodological focus. Such studies usually do not encounter the complexity of raw data from clinical practice and all the many formatting and manipulation decisions necessary to turn them into analyzable data. Because of this complexity and volume of code, researchers conducting applied work may not consider code sharing to be useful. The counterargument to this is that sharing code is unlikely to decrease transparency. Despite these divergent opinions, there was agreement that improved documentation of code when shared, and improved reporting of key study parameters [13], is important. This forms the basis for our recommendation that researchers should provide code with sufficient documentation (Recommendation #2, Table 5).

The second concern regarding IP rights is challenging. Although appropriate licensing, and authorship, might partially address concerns for academic researchers, it may not be sufficient for companies for whom authorship on a scientific publication is not a key incentive or where the programming code is the underlying IP, such as for analysis platforms or consultancies. These concerns mirror broader discussions around the benefits and risks of open-source software development more generally, with different companies taking different views on open versus proprietary models of software development. It is also worth noting that IP rights may also be dictated by data owners, which could put legal limits on code sharing for users accessing such databases for research purposes. In these situations, transparent reporting of key study parameters and means to support the computational reproducibility of studies, for example, by retaining records of the order in which functions are called for interactive interfaces, becomes particularly important. Specific examples of how researchers may increase transparency of

TABLE 3 | Other transparency practices across all included articles.

Characteristic		N (%)
Total		968 (100)
Preregistration	Yes, as reported by the authors	54 (5.6)
Data sharing		
	Data access procedures described/"available on request"	75 (7.7)
	Yes	21 (2.2)
Codelist sharing		
	Applicable	677 (69.9)
	At least one	499 (73.7, N = 677) ^a
	Published elsewhere	10 (6.4, N = 677)
Reporting guideline	Yes, as reported by the authors	44 (4.5)
Preprinting	Yes, as reported by the authors	0 (0)

^aThe denominator were those 677 articles where code lists were applicable.

**FIGURE 2** | Proportion of articles sharing programming code over time.

analytic workflows when code cannot be shared includes: the use of graphical study design diagrams [25], sharing of code lists [13], making protocols or statistical analysis plans publicly available [26, 27], and provision of pseudocode or well-documented, packaged software.

Recent years have also seen efforts to streamline RWD analytics with standardized, well-documented and unit-tested packages which can improve both the quality and transparency of analyses [28]: such code is sometimes open-source, and sometimes not. This raises an interesting distinction between studies which generate code, and those that rely at least in part on the reuse of existing code. A more narrow definition of code

sharing might consider that this is only be applicable for code that is written for a certain study, and not for code which is re-used: for example, we wouldn't consider the lack of source code for a certain procedure in proprietary programming tools that have undergone extensive validation tests, like SAS, Stata, or Action, to prevent full code sharing. Equally, simply listing use of an open source package would not be considered code sharing (although there are separate arguments for why using open source can support transparency and computational reproducibility efforts). For example, our review covered several papers which reported use of Sentinel tools. These are publicly available [29, 30], but as our definition of code sharing did not consider that use of an open source language in and of itself represented code sharing, these were generally classified as not sharing code. We recognize that this represents a gray area, although without specifically reporting which code modules were used and providing permanent references to them, the relevant code can be hard for others to find despite being public. Code sharing might also be defined as the sharing of all code necessary to reproduce the study results, although it is challenging to determine whether this has been provided without access to the underlying data and/or a more granular mapping of code to the reported outputs. The definition of code sharing we used was consciously broad, and as we considered virtually any sharing of programming code as code sharing, including partial code sharing and sharing of "code snippets," our prevalence estimates are likely higher than those in studies using more restrictive definitions. Overall, this highlights that there can be considerable complexity in how "code sharing" is defined, which will be relevant for journal editors or funders considering potential code sharing mandates [31, 32], and for the monitoring of code sharing going forward.

Potential barriers to code sharing vary depending on the context researchers work in. For example, academic researchers typically do not face challenges around IP, but may struggle with a lack of time, incentives, or prevailing research culture (including self-consciousness and fear of public scrutiny of code). These may be tackled through the provision of dedicated training to improve the quality and researchers' confidence in their code, and through ensuring that workflows which enhance transparency are appropriately valued by employers (e.g., by making these part of academic promotion criteria) and funders. Whilst these concerns are important to address, there are many potential benefits to code sharing which are also worth highlighting. These include direct benefits, such as facilitating independent error detection and validation, and indirect benefits, such as adoption of git for versioning code, code reviews and easier collaboration with team members. Collectively, alongside the broader adoption of open science practices, these can positively contribute to existing workflows, particularly for researchers working in contexts where there is a lack of structured quality assurance processes [4]. There is also an argument that code sharing is particularly important for publicly or federally funded research to ensure that public resources are used effectively, and funders are increasingly mandating or encouraging sharing of both data and code [33, 34]. This is one of the principles that has informed the open source approach taken by the EMA funded DARWIN-EU [35].

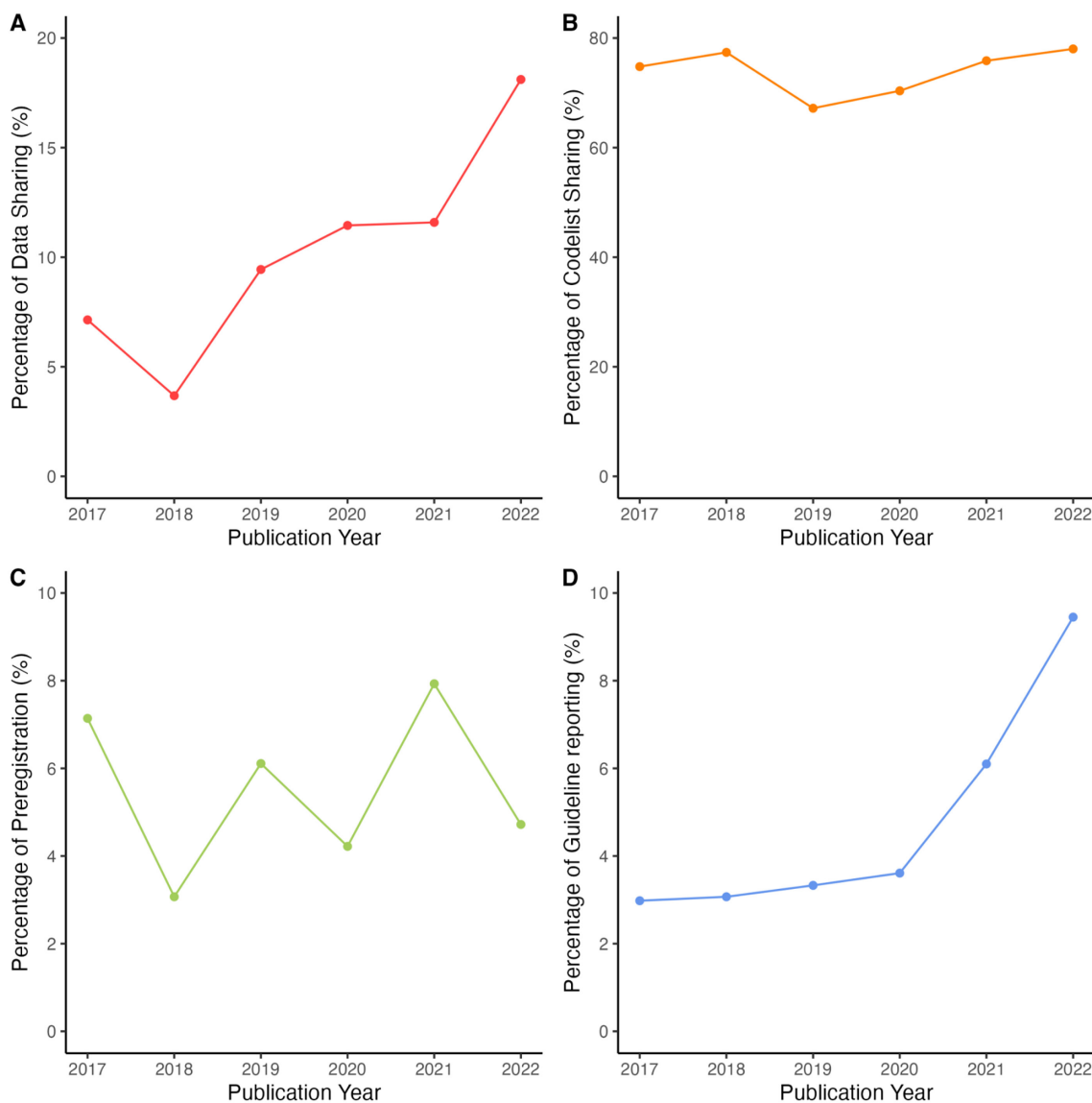


FIGURE 3 | Proportion of articles following other transparency practices. (A) Data sharing, (B) Codelist sharing (at least one), (C) Preregistration, and (D) Reported adherence to a reporting guideline.

Taking into account the breadth of perspectives represented in our working group, we have developed a series of recommendations for how researchers who choose to share their code can do so most effectively (Table 5). We hope this can act as a starting point for continued discussions around the utility of code sharing in pharmacoepidemiology.

4.2 | Comparison to Previous Studies

Recently, Hamilton and colleagues conducted a systematic review and meta-analysis of data and code sharing in the medical research literature between 2012 and 2018. They found a persistently low prevalence of code sharing, at <0.5% in all calendar years considered [36]. Some of the articles contributing to this assessment used automated tools for detecting programming code sharing based on keyword searches, and this might have missed instances of code sharing where this was not clearly signposted leading to a slight underestimation of

prevalence. In our study, approximately half of the articles that shared programming code did so in their supporting information and did not always clearly signpost this in the methods. This is an interesting finding, and indicates that there may be a broader willingness to share code, but that researchers may not have access to or familiarity with code sharing platforms that facilitate easier code sharing. Code sharing is also reported to be rare in other scientific disciplines, including in a sample of psychology articles (1%, 95% CI = 0%–1% between 2014 and 2018) [37] and cancer biology (4%, 95% CI = 2%–6% in 2019) [38]. The increase in code sharing we observed in 2022 might, at least in part, reflect changing community standards over time as interest in open science practices has increased among epidemiologists: it remains to be seen whether this increase is part of a sustained trend.

The FAIR standards (Findable, Accessible, Interoperable, Re-usable) standards developed for data sharing have been applied to research software and epidemiological workflows,

TABLE 4 | Considerations for code sharing in pharmacoepidemiology.

Reasons given for sharing code	Reasons given for not sharing code
<p>Programming code sharing is “good practice.” Programming code sharing is perceived to be good practice in pharmacoepidemiology, or within a specific sub-field in this discipline.</p> <p>Provide insight into how a protocol/SAP was interpreted. Even when the write-up of a study adheres to best-practice reporting guidelines, there may be ambiguity in how certain parameters should be defined. Programming code can therefore provide an additional level of clarification.</p> <p>Increase the uptake of a newly proposed or complex methodology. When researchers develop or propose a new methodology, providing programming code that allows other researchers to implement this can facilitate further use and evaluation of new methods. This is particularly the case when the provided code is provided as a formal package, together with documentation and tutorials.</p> <p>Facilitate direct replication, error detection and correction. Making the programming code available can facilitate direct replication efforts, as well as the detection (and consequent correction) of programming errors that would not otherwise have been detected.</p> <p>Enable trust in researchers. By allowing members of the public to see exactly how their data has been processed, code sharing can facilitate trust between the public and researchers.</p> <p>Protection against fraud. It is challenging to generate the amount of programming code required for an analyses of complex electronic health records without such an analysis ever taking place, and requirements to share programming code may therefore prevent the publication of fraudulent studies (such as the Surgisphere papers).</p> <p>Improve scientific workflows. If researchers were encouraged to share programming code, they may end-up adopting improved workflows that facilitate easier code sharing, such as formal version control using git, the creation of documentation and programming code review.</p>	<p>Lack of utility to others. The complexity of the data processing required for a typical pharmacoepidemiological study using real-world data sources can be large, typically involving tens of thousands of lines of code. Such code bases can be complex to interpret and re-use, and the act of making them publicly available may not, in and of itself, represent the most effective way of transparently communicating the design and implementation of a certain study.</p> <p>Commercial and IP considerations. Programming code may have commercial value for an organization, for example, for consultancies who sell data analyses services. Researchers who contract out part or all aspects of a research project may also not hold the right to publish the programming code, as these may be retained by the contract research organization who conducted the research.</p> <p>Resource considerations. The additional time required in making programming code ‘publication-ready’ could be significant, and if the aim of sharing code is to promote re-use, researchers raised additional concerns that this would require the creation of formal tutorials and/or result in maintenance requests from others seeking to re-use all or parts of the code.</p> <p>Knowledge barriers. Not all researchers may have the knowledge required to share programming code effectively whilst ensuring that patient confidentiality and organizational security is maintained.</p> <p>Lack of incentives. Existing incentives for code sharing are limited, and researchers might perceive that there are relatively limited benefits involved in sharing programming code.</p>

also resulting in a series in recommendations for how best to share research code, including in epidemiology [39, 40]. Some of our recommendations: for example, that programming code should be shared with a permanent digital identifier and with some provision of metadata, echo these recommendations. However, full adherence to FAIR principles would require the use of open-source software for analyses, as well as efforts to ensure that code is portable across operating systems. This would likely be challenging for many researchers to achieve and would pose a significant time burden. We have therefore

not provided prescriptive recommendations, and recognize that progress toward FAIR code sharing is likely to be incremental [40].

4.3 | Strengths and Limitations

There are some important limitations to this work. First, our literature review only covered a single journal, which although leading in the field may mean that our findings may

TABLE 5 | Recommendations for code sharing in pharmacoepidemiology.

Recommendation	Rationale
1 Reporting of open science practices for pharmacoepidemiological studies should be standardized	The reporting of open science practices varied, and very few papers had specific, standardized statements reporting on these practices at the end of the manuscript. Occasionally, programming code was found in the supporting information without being signposted anywhere in the main body text. Standardized reporting at the end of a paper—indicating whether or not code is accessible, and, if it is accessible, how others can access it—may make information on open research practices easier to find and, where relevant, re-use. This can ultimately make sharing more impactful. Reporting is likely most useful for open science practices that have the potential to increase the transparency of or confidence in a given study: such as preregistration of a study protocol, code sharing, code list sharing and data sharing. For other practices, such as preprinting, simply reporting the existence of a preprinted version of a paper is unlikely to either increase confidence in or the transparency of a particular publication. Journals seeking to increase the transparency of the research record may consider combining preprinting with open-peer review and versioning of research articles, a model recently adopted by Wellcome Open Research.
2 Researchers should provide documentation/metadata and ensure that the code contains comments	Most code shared contained some comments or overarching documentation, but approximately 20% had neither. The provision of comments and documentation is important to enable re-use of code, although is also important for overall good programming practice, as highlighted by PHUSE guidance [21]. The extent of documentation should match the task at hand, with more extensive or standardized documentation likely to be particularly useful for code specifically developed and shared with re-use in mind, such as standardized analytical pipelines like DARWIN EU®, the FDA Sentinel system or OpenSAFELY.
3 Researchers should use permanent digital identifiers (DOIs) to link to appropriate versions of the code	Most programming code was shared in supporting information, but in instances where programming code was shared online, we occasionally encountered issues accessing the underlying data or code due to broken links. DOIs are persistent, unique digital identifiers that can be used to overcome so-called “link rot” (where a link becomes expired). When sharing programming code on Github, generating a DOI requires archiving a specific version of the code with Zenodo [22]. Ensuring that the links point towards a permanent, stable version of the code also prevents confusion in case of updates to the underlying code repository, for example, in response to reviewers' comments. Finally, it is important for researchers to double check the settings of archived code or data they are intending to make public, as we encountered instances of code and data being shared but not accessible without further contacting the authors due to the repository being set to private. This might reflect authors wanting to keep data or code private until publication, and consequently forgetting to change the settings. Journals could assist with such issues by verifying any external links to data and code at the proof stage.

(Continues)

TABLE 5 | (Continued)

Recommendation	Rationale
4 Researchers should assign a license to shared code	Published programming code was rarely associated with a license. Researchers may not be aware that they retain full copyright to programming code that is shared on webpages such as Github, unless they specifically assign a license that enables re-use of the code. Which license is most appropriate will vary, and we recommend that research teams take time to choose and assign a license that suits their project's needs [23]. Resources for choosing an appropriate licenses are available, for example, https://choosealicense.com/ . Concerns about authorship and attribution where significant IP is shared may also be mitigated by licensing specific code, algorithms or analytical approaches, either separately or alongside other outputs, for example, a tutorial article describing functionality and application of an algorithm or method.
5 Computational reproducibility and re-use of the programming code can be promoted by: <ul style="list-style-type: none"> a sharing the code in a machine-readable format or on a code sharing platform b cataloguing the environment, for example in a requirements file c documenting any quality control or validation of the code d providing synthetic or “dummy” data 	We suggest that the following strategies may enhance computational reproducibility and promote re-use of code, although recognize that some of these proposals can be time-intensive and not all steps may be suitable for all research projects. Synthetic data in particular can be challenging to derive, although recent advancements in the use of AI for generating this represents an interesting development in this area [24]. In the absence of synthetic data, an intermediate step towards this recommendation is to describe the structure and formats of the input data being used. In addition, it is worth noting that the use of a code sharing platform to share code has many additional benefits including encouraging the use of version control tools, retention of a history of the code development and quality-control process through commits, and by offering built-in, automated test functionality.
6 Before programming code is shared, steps should be taken to ensure that it does not contain any sensitive information, for example, identifiable patient data	The risk of a personal data breach from sharing programming code is low, and existing good programming practice already emphasize the importance of not hard-coding patient information into scripts. However, researchers may include such information accidentally, or accidentally commit patient-level data to web-based code repositories. Research teams who are increasingly making programming code public should take steps to ensure that their existing standard operating procedures consider the risk of such accidental data breaches, that appropriate preventative measures, for example code review, are in place, and that there exists a detailed plan of action for any breaches.

not be generalizable to all pharmacoepidemiology research. However, the choice of *PDS* was considered likely to offer a reasonably representative sample in terms of describing trends and current practice. It is worth noting that *PDS* does not have restrictions on supplemental content, which might allow for greater sharing of methods and code compared to other journals. Secondly, due to resource constraints we did not double-extract data, which may have resulted in data extraction errors. We also did not extract information on whether or not a certain common data model (CDM) was used, so cannot comment on whether code sharing was more or less common among studies using a CDM. Using a CDM can make a study more transparent because the data structure is well described and, often, programming code must be shared across multiple

partners [41]. However, this does not necessarily imply that the programming code is shared externally, which is the target of our study. Large collaborations would still need to make an additional effort to make their code accessible from external researchers. Whilst we attempted to extract information on the coverage of analytical code (e.g., did the code cover data management, statistical analysis, or both?), this was challenging to assess. Specifically, during data extraction it became clear that these definitions were not straight-forward to apply to methodological or simulation studies: which were coincidentally the types of studies most likely to share code. In addition, even when code could be classified as either “data management” or “analysis,” it was not easy to assess whether code coverage was “complete,” in the sense that all of the code

necessary to reproduce that part of the research project was provided. However, it is worth flagging that code sharing is not necessarily a binary undertaking: and we observed several instances where researchers shared “some” of their code, as opposed to all or none. We recommend future researchers interested in studying this conduct extensive pilot work to explore how code can be classified and coverage assessed. The development of standards for documenting or visualizing programming code might aid in such endeavors [42], in the meantime, we caution against interpreting this variable in our publicly available dataset.

We also want to emphasize that our recommendations are a starting point for discussion and consciously broad, and not intended to be prescriptive in how programming code is written or shared. This is to ensure that they are applicable to a wide range of researchers working in different settings, but it also means that we have not provided specific guidance on whether to use, or how to use, certain tools such as Github or Docker. The provision of training and best-practice guidance on the use of these tools, and code development more generally, for pharmacoepidemiologists represent an important area of future work for our community [43], and continued monitoring of open science indicators will be important to determine if our recommendations need to be adjusted as practice evolves over time.

Beyond code sharing, our review highlighted limitations surrounding open science practices and study reporting that we were not able to expand on in detail. For example, the programming language was not reported in 29% of studies despite data analysis being conducted. Whilst we observed that many studies shared clinical codelists, a broad definition of “codelist” was applied that included studies where codes, often for the outcome, were listed in the main text (usually in the methods section). Further guidance encouraging researchers to provide codelists in machine-readable formats and documenting key metadata (e.g., around key decisions made when deriving the codelist) could facilitate better understanding and efficient reuse of these codelists [44]. Finally, we found that assessing author-reported use of preregistration, checklists and preprinting was insufficient to capture the extent of these practices and encountered several instances of authors using these practices without reporting on this in their papers: our estimates of these practices are therefore underestimated. Future initiatives aiming to study these practices may need to use data from external sources (such as preregistration databases and preprint servers) to gather more complete estimates of these practices. The low prevalence of reported preregistration may also, at least in part, reflect the fact that this is likely to lag other open science practices in the published literature, as it can only be completed before study initiation.

5 | Conclusion

In this evaluation of code sharing practices in pharmacoepidemiology, we found that programming code sharing was rare, practiced in only 4.8% of papers published in *Pharmacoepidemiology and Drug Safety* between 2017 and 2022. We identified a number of barriers to code sharing, some

of which warrant consideration by journals, funders, or regulators considering measures to either encourage or mandate code sharing in pharmacoepidemiology. Finally, we developed a series of recommendations which seek to improve the reporting of code sharing, and to improve the utility and the re-usability of publicly available code in situations when investigators opt to share programming code. We hope that our work can serve as a starting point for a continued discussion around the role of programming code sharing in improving the transparency of pharmacoepidemiological research and stimulate the development of standards for code sharing.

5.1 | Plain Language Summary

This study examined how often researchers share programming code in pharmacoepidemiology, through a review of articles published in *Pharmacoepidemiology and Drug Safety* between 2017 and 2022. We found that only a small percentage of studies (ranging from 1.8% in 2017 to 9.5% in 2022) shared their programming code, although this was more common for methodological and simulation-based papers. We propose six recommendations for researchers who want to share code, including the use of permanent digital identifiers, appropriate licenses, and following good software practices, for example, by providing metadata and documentation. Overall, the study advocates for better reporting of whether or not code is available for a certain publication, and how to access code when this is available.

Acknowledgments

This article was supported by a manuscript proposal grant from the International Society for Pharmacoepidemiology (ISPE). We are very grateful to thoughtful input from the following individuals during the development of this manuscript: Renee De Waal, University of Cape Town; Ben Goldacre, University of Oxford; Sinéad Langan, London School of Hygiene and Tropical Medicine; Adrienne Chen, University of Hong Kong, Andrew Bate, GSK; Joshua Gagne, Johnson & Johnson; and Arnold K Chan, TriNetX.

Ethics Statement

Ethical approval was not required for this study since all analyses were based on data either available in the public domain or fully simulated.

Conflicts of Interest

J.T. and A.S. were both employed by LSHTM on fellowships sponsored by GSK during the conduct of this work. J.L. and J.P. are employees of, and own shares in GSK. S.S. is a consultant to Aetion, Inc., a software manufacturer in which he owns equity and the principal investigator of investigator-initiated grants to the Brigham and Women’s Hospital from Boehringer Ingelheim unrelated to the topic of this study. I.D. has received unrestricted grants from GSK and AstraZeneca and holds stocks in GSK. D.R.M.L., P.A., K.D., R.G., C.M., D.P.A., and S.V.W. have no relationships to disclose.

References

1. Open Science Collaboration, “Estimating the Reproducibility of Psychological Science,” *Science* 349, no. 6251 (August 2015): aac4716.

2. T. M. Errington, M. Mathur, C. K. Soderberg, et al., "Investigating the Replicability of Preclinical Cancer Biology," *eLife* 7, no. 10 (December 2021): e71601.
3. S. V. Wang, S. K. Sreedhara, and S. Schneeweiss, "Reproducibility of Real-World Evidence Studies Using Clinical Practice Data to Inform Regulatory and Coverage Decisions," *Nature Communications* 13, no. 1 (August 2022): 1–11.
4. M. Mathur and M. P. Fox, "Toward Open and Reproducible Epidemiology [Internet]. OSF Preprints," 2022, <https://osf.io/bpkf7/>.
5. M. R. Munafò, B. A. Nosek, D. V. M. Bishop, et al., "A Manifesto for Reproducible Science," *Nature Human Behaviour* 1, no. 1 (January 2017): 1–9.
6. L. Besançon, N. Peiffer-Smadja, C. Segalas, et al., "Open Science Saves Lives: Lessons From the COVID-19 Pandemic," *BMC Medical Research Methodology* 21, no. 1 (June 2021): 117.
7. A. Trisovic, M. K. Lau, T. Pasquier, and M. Crosas, "A Large-Scale Study on Research Code Quality and Execution," *Scientific Data* 9, no. 1 (February 2022): 60.
8. Stop Hiding Your Code [Internet], "EveryONE," 2018, <https://everyone.plos.org/2018/04/18/stop-hiding-your-code/>.
9. B. Goldacre, C. E. Morton, and N. J. DeVito, "Why Researchers Should Share Their Analytic Code," *BMJ* 367 (November 2019): l6365.
10. B. E. Shepherd, M. Blevins Peratikos, P. F. Rebeiro, S. N. Duda, and C. C. McGowan, "A Pragmatic Approach for Reproducible Research With Sensitive Data," *American Journal of Epidemiology* 186, no. 4 (August 2017): 387–392.
11. N. J. DeVito, C. Morton, A. G. Cashin, G. C. Richards, and H. Lee, "Sharing Study Materials in Health and Medical Research," *BMJ Evidence-Based Medicine* 28, no. 4 (September 2022): 255–259.
12. A. Bate, "Guidance to Reinforce the Credibility of Health Care Database Studies and Ensure Their Appropriate Impact," *Pharmacoepidemiology and Drug Safety* 26, no. 9 (2017): 1013–1017.
13. S. V. Wang, S. Schneeweiss, M. L. Berger, et al., "Reporting to Improve Reproducibility and Facilitate Validity Assessment for Healthcare Database Studies V1.0," *Pharmacoepidemiology and Drug Safety* 26, no. 9 (2017): 1018–1032.
14. Wellcome [Internet], "Our New Policy on Sharing Research Data: What It Means for You," 2017, <https://wellcome.org/news/our-new-policy-sharing-research-data-what-it-means-you>.
15. "PCORI's Policy for Data Management and Data Sharing | PCORI [Internet]," 2018, <https://www.pcori.org/about/governance/pcoris-policy-data-management-and-data-sharing>.
16. "Data Management & Sharing (DMS) [Internet]," accessed January 23, 2024, <https://www.nimhd.nih.gov/programs/extramural/data-management-sharing.html>.
17. K. Abbasi, "A Commitment to Act on Data Sharing," *BMJ* 382 (July 2023): 1609.
18. GOV.UK [Internet], "Better, Broader, Safer: Using Health Data for Research and Analysis," accessed September 15, 2022, <https://www.gov.uk/government/publications/better-broader-safer-using-health-data-for-research-and-analysis>.
19. S. M. Langan, S. A. Schmidt, K. Wing, et al., "The Reporting of Studies Conducted Using Observational Routinely Collected Health Data Statement for Pharmacoepidemiology (RECORD-PE)," *BMJ* 363 (November 2018): k3532.
20. S. V. Wang, S. Pinheiro, W. Hua, et al., "STaRT-RWE: Structured Template for Planning and Reporting on the Implementation of Real World Evidence Studies," *BMJ* 372 (January 2021): m4856.
21. PHUSE, "Good Programming Practice Guide," accessed August 5, 2024, <https://advance.phuse.global/display/WEL/Good+Programming+Practice+Guidance>.
22. GitHub Docs [Internet], "Referencing and Citing Content," accessed January 8, 2024, <https://ghdocs-prod.azurewebsites.net/en/repositories/archiving-a-github-repository/referencing-and-citing-content>.
23. GitHub Docs [Internet], "Licensing a Repository," accessed January 8, 2024, <https://ghdocs-prod.azurewebsites.net/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository>.
24. L. Mosquera, K. El Emam, L. Ding, et al., "A Method for Generating Synthetic Longitudinal Health Data," *BMC Medical Research Methodology* 23 (2023): 67.
25. S. Schneeweiss, J. A. Rassen, J. S. Brown, et al., "Graphical Depiction of Longitudinal Study Designs in Health Care Databases," *Annals of Internal Medicine* 170, no. 6 (2019): 398–406, <https://doi.org/10.7326/M18-3079>.
26. L. S. Orsini, B. Monz, C. D. Mullins, et al., "Improving Transparency to Build Trust in Real-World Secondary Data Studies for Hypothesis Testing—Why, What, and How: Recommendations and a Road Map From the Real-World Evidence Transparency Initiative," *Pharmacoepidemiology and Drug Safety* 29, no. 11 (2020): 1504–1513, <https://doi.org/10.1002/pds.5079>.
27. S. V. Wang, A. Pottgård, W. Crown, et al., "HARmonized Protocol Template to Enhance Reproducibility of Hypothesis Evaluating Real-World Evidence Studies on Treatment Effects: A Good Practices Report of a Joint ISPE/ISPOR Task Force," *Pharmacoepidemiology and Drug Safety* 32, no. 1 (2023): 44–55, <https://doi.org/10.1002/pds.5507>.
28. D. S. Bové, H. Seibold, A. L. Boulesteix, et al., "Improving Software Engineering in Biostatistics: Challenges and Opportunities," accessed January 24, 2023, <https://doi.org/10.48550/arXiv.2301.11791>.
29. Sentinel System [Internet], "Public Repositories," accessed March 1, 2024, <https://dev.sentinelssystem.org/repos?visibility=public>.
30. Sentinel Initiative [Internet], "Software Packages Toolkits," accessed May 1, 2024, <https://www.sentinelinitiative.org/methods-data-tools/software-packages-toolkits#search-software-packages-toolkits>.
31. E. Loder, H. Macdonald, T. Bloom, and K. Abbasi, "Mandatory Data and Code Sharing Published by the BMJ," *BMJ* 384 (2024): q324, <https://doi.org/10.1136/bmj.q324>.
32. PCORI [Internet], "New Policy on Data Management and Data Sharing," accessed May 1, 2024, <https://www.pcori.org/blog/pcoris-new-policy-data-management-and-data-sharing-step-forward-open-science>.
33. European Commission [Internet], "Open Access and Data Management," accessed May 1, 2024, https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-data-management/data-management_en.htm.
34. Wellcome [Internet], "Data, Software and Materials Management and Sharing Policy," accessed May 1, 2024, <https://wellcome.org/grant-funding/guidance/data-software-materials-management-and-sharing-policy>.
35. European Medicines Agency [Internet], "Data Analysis and Real World Interrogation Network (DARWIN EU)," accessed May 1, 2024, <https://www.ema.europa.eu/en/about-us/how-we-work/big-data/data-analysis-real-world-interrogation-network-darwin-eu>.
36. D. G. Hamilton, K. Hong, H. Fraser, A. Rowhani-Farid, F. Fidler, and M. J. Page, "Prevalence and Predictors of Data and Code Sharing in the Medical and Health Sciences: Systematic Review With Meta-Analysis of Individual Participant Data," *BMJ* 382 (July 2023): e075767.
37. T. E. Hardwicke, R. T. Thibault, J. E. Kosie, J. D. Wallach, M. C. Kidwell, and J. P. A. Ioannidis, "Estimating the Prevalence of Transparency and Reproducibility-Related Research Practices in Psychology (2014–2017)," *Perspectives on Psychological Science* 17, no. 1 (January 2022): 239–251.
38. D. G. Hamilton, M. J. Page, S. Finch, S. Everitt, and F. Fidler, "How Often Do Cancer Researchers Make Their Data and Code Available and What Factors Are Associated With Sharing?" *BMC Medicine* 20, no. 1 (November 2022): 438.

39. M. Barker, N. P. Chue Hong, D. S. Katz, et al., "Introducing the FAIR Principles for Research Software," *Scientific Data* 9, no. 1 (October 2022): 622.
40. M. García-Closas, T. U. Ahearn, M. M. Gaudet, et al., "Moving Towards FAIR Practices in Epidemiological Research," *American Journal of Epidemiology* (February 2023): kwad040.
41. R. Gini, M. C. J. Sturkenboom, J. Sultana, et al., "Different Strategies to Execute Multi-Database Studies for Medicines Surveillance in Real-World Setting: A Reflection on the European Model," *Clinical Pharmacology and Therapeutics* 108 (2020): 228–235, <https://doi.org/10.1002/cpt.1833>.
42. R. Gini, D. Messina, W. Aarts, et al., "Representation of Study Scripts to Improve Transparency and Efficiency in Multidatabase Distributed Vaccine Studies," in *Poster Presented to: 39th International Conference on Pharmacoepidemiology and Therapeutic Risk Management* (Halifax, Nova Scotia, 2023).
43. J. Weberpals and S. V. Wang, "The FAIRification of Research in Real-World Evidence: A Practical Introduction to Reproducible Analytic Workflows Using Git and R," *Pharmacoepidemiology and Drug Safety* (January 2024).
44. J. Matthewman, K. Andresen, A. Suffel, et al., "Checklist and Guidance on Creating Codelists for Electronic Health Records Research," *NIHR Open Research* 4 (2024): 20, <https://doi.org/10.3310/nihropenres.13550.1>.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.