The FAIRification of pharmacoepidemiological research

A practical introduction to reproducible analytical workflows using git and R

**Authors**: Janick Weberpals1, Shirley V. Wang1

**Author affiliation**: 1Division of Pharmacoepidemiologyand Pharmacoeconomics, Department of Medicine, Brigham and Women’s Hospital, Harvard Medical School, Boston, MA

**Correspondence:**

Janick Weberpals, RPh, PhD

Instructor in Medicine

Division of Pharmacoepidemiology and Pharmacoeconomics,

Department of Medicine, Brigham and Women’s Hospital, Harvard Medical School,

1620 Tremont Street, Suite 3030-R, Boston, MA 02120, USA

Phone: 617-278-0932

Fax: 617-232-8602

Email: [jweberpals@bwh.harvard.edu](mailto:jweberpals@bwh.harvard.edu)

**Word count:** xxx words

**Tables:**

**Figures:**

**Supplementary material:**

**Keywords:**

**Funding:**

**Disclosures/COI:** Janick Weberpals reports no conflicts of interest.

**Analytical code sharing statement:**Figures and exemplary analyses were conducted using R version 4.1.2. Detailed information on used packages and versions can be found in the Supplementary Methods and code used in this study is available at .xxx.

**Data sharing statement:** …

**Acknowledgments:**

# Abstract

**Objective**

**Study Design and Settings**

**Results**

**Conclusions**

# 1. Introduction

Transparency and reproducibility in conducting healthcare database studies in pharmacoepidemiology is one of the most critical scientific requirements for meaningful research. While many advances have been made in the documentation and reporting of study protocols and results1–4, the transparency around the actual implementation and analytical steps is still significantly lacking, especially when analyses are implemented via line programming steps in statistical programming languages like R5 or Python. Although in other quantitative disciplines such as computational biology6–8 or health informatics9, the sharing of programming code is frequently observed, this practice is not as common in the (pharmaco-)epidemiological community.

Here, the principles of *FAIR* *data*10 can be extended to the implementation of pharmacoepidemiological study workflows by making all components of a study **F**indable, **A**ccessible, **I**nteroperable, and **R**eproducible. Version control systems (VCS), such as *Git*, can thereby provide easy and powerful tools to keep track of the versioning of analytical code, tables, figures and other resources, track and resolve errors, collaborate with peers, and share resources instantaneously upon submission or publication of a manuscript. By design, VCS workflows automatically enable researchers to comment, track and compare major changes made to documents and hence, increase the ability to comprehend the evolution of a project while maintaining a complete audit trail of all documents.11 Without VCS, this is often naturally done by assigning uninformative file names such as "*final\_analysis\_1\_rev\_more\_changes.R*" which is bad practice as this is highly error prone, it is untransparent for collaborating project members and outside personnel and changes are difficult to track and reconcile. Particularly in disciplines like pharmacoepidemiology, where analyses can have far reaching impact on healthcare decisions, analytical code needs to be accessible and reproducible by anyone, not only the analyst who originally carried out the analysis.

To increase the adoption of transparent and repdroducible workflows using VCS, this tutorial aims to give a practical and tailored introduction for pharmacoepidemiologists on how to set up, structure, and implement workflows using Git as the most frequently used VCS in the scientific community.12 We will first shortly introduce Git and how it can be used to for collaboration and dissemination of study results using project repositories. Then we will provide a technical step-by-step guidance on how to integrate Git in an analytical workflow using line commands. We will additionally showcase examples using the R open-source programming language, although the basic principles can be applied to any major coding language.

# 2. Git basics

Git is a free and open source distributed VCS software which was developed in 2005 by the Linux developer community primarily with the intent to handle large software projects simply and efficiently.13

# References

1. Wang SV, Pinheiro S, Hua W, *et al.* STaRT-RWE: structured template for planning and reporting on the implementation of real world evidence studies. *BMJ* 2021: m4856. doi:[10.1136/bmj.m4856](https://doi.org/10.1136/bmj.m4856).

2. Schneeweiss S, Rassen JA, Brown JS, *et al.* Graphical Depiction of Longitudinal Study Designs in Health Care Databases. *Annals of Internal Medicine* 2019; **170**: 398. doi:[10.7326/m18-3079](https://doi.org/10.7326/m18-3079).

3. Berger ML, Sox H, Willke RJ, *et al.* Good practices for real-world data studies of treatment and/or comparative effectiveness: Recommendations from the joint ISPOR-ISPE Special Task Force on real-world evidence in health care decision making. *Pharmacoepidemiology and Drug Safety* 2017; **26**: 1033–1039. doi:[10.1002/pds.4297](https://doi.org/10.1002/pds.4297).

4. Wang SV, Pottegård A, Crown W, *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* 2022; **32**: 44–55. doi:[10.1002/pds.5507](https://doi.org/10.1002/pds.5507).

5. R Core Team. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2023. Available at: <https://www.R-project.org/>.

6. Almugbel R, Hung L-H, Hu J, *et al.* Reproducible Bioconductor workflows using browser-based interactive notebooks and containers. *Journal of the American Medical Informatics Association* 2017; **25**: 4–12. doi:[10.1093/jamia/ocx120](https://doi.org/10.1093/jamia/ocx120).

7. Mammoliti A, Smirnov P, Nakano M, *et al.* Orchestrating and sharing large multimodal data for transparent and reproducible research. *Nature Communications* 2021; **12**. doi:[10.1038/s41467-021-25974-w](https://doi.org/10.1038/s41467-021-25974-w).

8. Russell PH, Johnson RL, Ananthan S, Harnke B, Carlson NE. A large-scale analysis of bioinformatics code on GitHub. Qin Z (ed.). *PLOS ONE* 2018; **13**: e0205898. doi:[10.1371/journal.pone.0205898](https://doi.org/10.1371/journal.pone.0205898).

9. Bakken S. The journey to transparency, reproducibility, and replicability. *Journal of the American Medical Informatics Association* 2019; **26**: 185–187. doi:[10.1093/jamia/ocz007](https://doi.org/10.1093/jamia/ocz007).

10. García-Closas M, Ahearn TU, Gaudet MM, *et al.* Moving Toward Findable, Accessible, Interoperable, Reusable Practices in Epidemiologic Research. *American Journal of Epidemiology* 2023; **192**: 995–1005. doi:[10.1093/aje/kwad040](https://doi.org/10.1093/aje/kwad040).

11. Ram K. Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine* 2013; **8**. doi:[10.1186/1751-0473-8-7](https://doi.org/10.1186/1751-0473-8-7).

12. Blischak JD, Davenport ER, Wilson G. A Quick Introduction to Version Control with Git and GitHub. Ouellette F (ed.). *PLOS Computational Biology* 2016; **12**: e1004668. doi:[10.1371/journal.pcbi.1004668](https://doi.org/10.1371/journal.pcbi.1004668).

13. Chacon S, Straub B. *Pro git*. Springer Nature, 2014.