

PRACTICAL ISSUES RELATED TO RESEARCH/PUBLICATION WORKFLOW

Lars Relund Nielsen

February 20, 2019

Department of Economics and Business

“ *An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.*

D. Donoho

”

- Reproducible results: The verification of a scientific experiment by other researchers using an independent experiment.
- Reproducible research idea: The ultimate product of academic research is the paper along with the full computational environment used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.
- Reproducible research = automation (by running a script everything should be done).

- Reproducible results: The verification of a scientific experiment by other researchers using an independent experiment.
- Reproducible research idea: The ultimate product of academic research is the paper along with the full computational environment used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.
- Reproducible research = automation (by running a script everything should be done).

- Reproducible results: The verification of a scientific experiment by other researchers using an independent experiment.
- Reproducible research idea: The ultimate product of academic research is the paper along with the full computational environment used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.
- Reproducible research = automation (by running a script everything should be done).

REPRODUCIBLE RESEARCH DIAGRAM

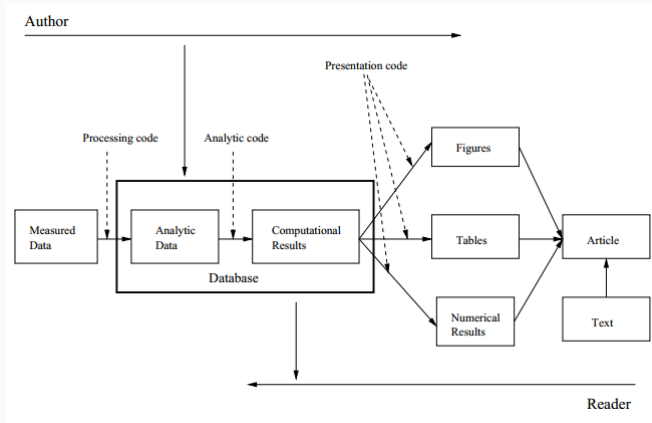


Figure 1: The research pipeline as a model for reproducible research Peng and Eckel, “Distributed reproducible research using cached computations”.

- Automatically regenerate documents when code, data, or assumptions change (good after receiving a review).
- Eliminate errors that occur when copy/transfer results into documents.
- Preserve your own “meta” comments about why analysis was performed in a certain fashion.
- Documentation for the analytic and computational processes from which conclusions are drawn (literal programming).
- You may get a greater network and may be cited more.

- Automatically regenerate documents when code, data, or assumptions change (good after receiving a review).
- Eliminate errors that occur when copy/transfer results into documents.
- Preserve your own “meta” comments about why analysis was performed in a certain fashion.
- Documentation for the analytic and computational processes from which conclusions are drawn (literal programming).
- You may get a greater network and may be cited more.

- Automatically regenerate documents when code, data, or assumptions change (good after receiving a review).
- Eliminate errors that occur when copy/transfer results into documents.
- Preserve your own “meta” comments about why analysis was performed in a certain fashion.
- Documentation for the analytic and computational processes from which conclusions are drawn (literal programming).
- You may get a greater network and may be cited more.

- Automatically regenerate documents when code, data, or assumptions change (good after receiving a review).
- Eliminate errors that occur when copy/transfer results into documents.
- Preserve your own “meta” comments about why analysis was performed in a certain fashion.
- Documentation for the analytic and computational processes from which conclusions are drawn (literal programming).
- You may get a greater network and may be cited more.

- Automatically regenerate documents when code, data, or assumptions change (good after receiving a review).
- Eliminate errors that occur when copy/transfer results into documents.
- Preserve your own “meta” comments about why analysis was performed in a certain fashion.
- Documentation for the analytic and computational processes from which conclusions are drawn (literal programming).
- You may get a greater network and may be cited more.

- R: Program computing and producing graphics (you may even use c or c++ for fast computations).
- RStudio: Programming environment/editor for R
- Using RStudio you can weave you computations together with your paper using knitr, LaTeX and R Markdown (can even make a Word file).
- Git: A distributed revision control system.
- Github: A web-based Git repository hosting service.

- R: Program computing and producing graphics (you may even use c or c++ for fast computations).
- RStudio: Programming environment/editor for R
- Using RStudio you can weave you computations together with your paper using knitr, LaTeX and R Markdown (can even make a Word file).
- Git: A distributed revision control system.
- Github: A web-based Git repository hosting service.

- R: Program computing and producing graphics (you may even use c or c++ for fast computations).
- RStudio: Programming environment/editor for R
- Using RStudio you can weave you computations together with your paper using knitr, LaTeX and R Markdown (can even make a Word file).
- Git: A distributed revision control system.
- Github: A web-based Git repository hosting service.

- R: Program computing and producing graphics (you may even use c or c++ for fast computations).
- RStudio: Programming environment/editor for R
- Using RStudio you can weave you computations together with your paper using knitr, LaTeX and R Markdown (can even make a Word file).
- Git: A distributed revision control system.
- Github: A web-based Git repository hosting service.

- R: Program computing and producing graphics (you may even use c or c++ for fast computations).
- RStudio: Programming environment/editor for R
- Using RStudio you can weave you computations together with your paper using knitr, LaTeX and R Markdown (can even make a Word file).
- Git: A distributed revision control system.
- Github: A web-based Git repository hosting service.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Automate your work flow. Create a script that run your code and embed the results in your paper.
- Use a revision control system when your write your code (this may even be useful when write your paper).
- Create graphics automatically.
- Use literal programming (Roxygen or doxygen)
- Make your code open source with easy access (e.g. using github or CoinOR)
- Make your test instances public available.

- Wikipedia (web), Comments on reproducibility (web).
- The paper by Nestler related to OR¹.
- The paper by Peng and Eckel².
- A paper by Stodden, Guo, and Ma comparing different journals policies about reproducible research³.
- A paper about CoinOR⁴.

¹Scott Nestler. *Reproducible (Operations) Research - A primer on reproducible research and why the O.R. community should care about it*. 2011.

²Roger D Peng and Sandrah P Eckel. “Distributed reproducible research using cached computations”. In: *Computing in Science and Engineering* 11.1 (2009), pp. 28–34.

³Victoria Stodden, Peixuan Guo, and Zhaokun Ma. “Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals”. In: *PloS one* 8.6 (2013).

⁴R. Lougee-Heimer. “The Common Optimization INTERface for Operations Research: Promoting open-source software in the operations research community”. In: *IBM Journal of Research and Development* 47.1 (Jan. 2003), pp. 57–66. DOI: 10.1147/rd.471.0057.