



HAL
open science

Caring about reproducibility in scientific research

Timothée Flutre

► **To cite this version:**

Timothée Flutre. Caring about reproducibility in scientific research. Master. EvoGEM, ENS Ulm, Paris, France. 2023, pp.39. hal-04279303

HAL Id: hal-04279303

<https://hal.inrae.fr/hal-04279303>

Submitted on 10 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Master M2 EvoGEM

Caring about reproducibility in scientific research

Timothée Flutre

UMR Génétique Quantitative et Evolution
Université Paris-Saclay — INRAE — CNRS — AgroParisTech

10/11/2023

Outline

Global context

Spectrum of solutions

Outline

Global context

Spectrum of solutions

Statistical Inference: the Big Picture (Kass, 2011)

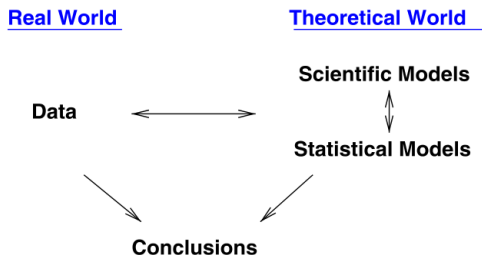
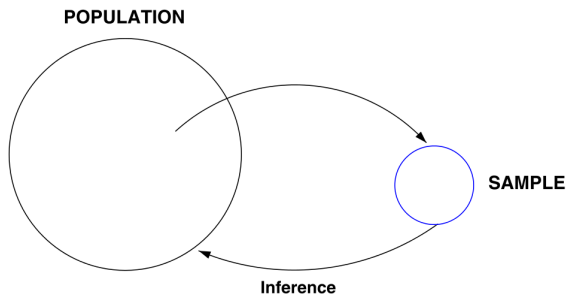


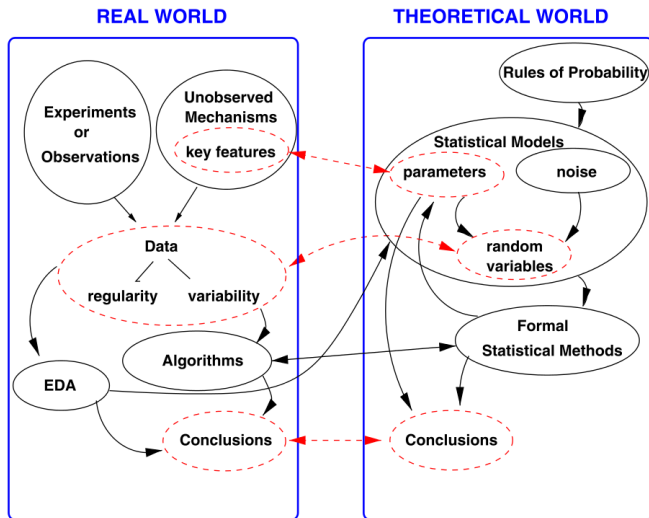
FIG. 1. *The big picture of statistical inference. Statistical procedures are abstractly defined in terms of mathematics but are used, in conjunction with scientific models and methods, to explain observable phenomena. This picture emphasizes the hypothetical link between variation in data and its description using statistical models.*

Statistical Inference: the Big Picture (Kass, 2011)

Standard conception:



Statistical Inference: the Big Picture (Kass, 2011)



A gloomy title

Why Most Published Research Findings Are False

John P. A. Ioannidis

Published: August 30, 2005 • DOI: 10.1371/journal.pmed.0020124

A gloomy title

Why Most Published Research Findings Are False

John P. A. Ioannidis

Published: August 30, 2005 • DOI: 10.1371/journal.pmed.0020124

- ▶ Did you hear about this paper?

A gloomy title

Why Most Published Research Findings Are False

John P. A. Ioannidis

Published: August 30, 2005 • DOI: 10.1371/journal.pmed.0020124

► Did you hear about this paper?

Total citations Cited by 2352



A gloomy title

Why Most Published Research Findings Are False

John P. A. Ioannidis

Published: August 30, 2005 • DOI: 10.1371/journal.pmed.0020124

- ▶ Did you hear about this paper?

Total citations Cited by 2352



- ▶ Did you read it?

Statistical model

PPV: positive predictive value

- ▶ posterior probability that a given research finding is true

Statistical model

PPV: positive predictive value

- ▶ posterior probability that a given research finding is true

Depends on:

- ▶ the prior probability of it being true (R)
- ▶ the power of the study ($1 - \beta$)
- ▶ the probability of claiming a relationship when none truly exists (α)

Statistical model

PPV: positive predictive value

- ▶ posterior probability that a given research finding is true

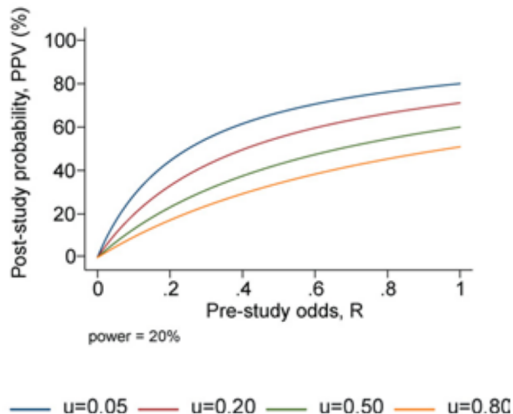
Depends on:

- ▶ the prior probability of it being true (R)
- ▶ the power of the study ($1 - \beta$)
- ▶ the probability of claiming a relationship when none truly exists (α)

But also on:

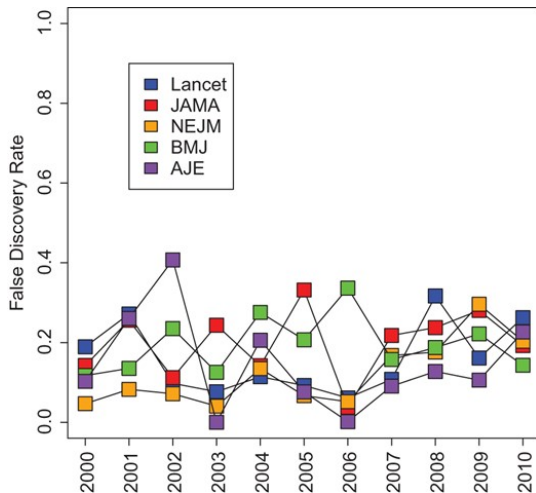
- ▶ bias (u)
- ▶ number of independent studies of equal power (n)

Example



Ioannidis (2005)

First, is it really that bad?



Jager & Leek (2014)

But it's not only about weak power and high bias...

DERIVING CHEMOSENSITIVITY FROM CELL LINES: FORENSIC BIOINFORMATICS AND REPRODUCIBLE RESEARCH IN HIGH-THROUGHPUT BIOLOGY


BY KEITH A. BAGGERLY¹ AND KEVIN R. COOMBES²

University of Texas

High-throughput biological assays such as microarrays let us ask very detailed questions about how diseases operate, and promise to let us personalize therapy. Data processing, however, is often not described well enough to allow for exact reproduction of the results, leading to exercises in “forensic bioinformatics” where aspects of raw data and reported results are used to infer what methods must have been employed. Unfortunately, poor documentation can shift from an inconvenience to an active danger when it obscures not just methods but errors. In this report we examine several related papers purporting to use microarray-based signatures of drug sensitivity derived from cell lines to predict patient response. Patients in clinical trials are currently being allocated to treatment arms on the basis of these results. However, we show in five case studies that the results incorporate several simple errors that may be putting patients at risk. One theme that emerges is that the most common errors are simple (e.g., row or column offsets); conversely, it is our experience that the most simple errors are common. We then discuss steps we are taking to avoid such errors in our own investigations.

A hopeful title, and no less crucial advice

How to Make More Published Research True

John P. A. Ioannidis 

Published: October 21, 2014 • DOI: 10.1371/journal.pmed.1001747

Read it here!

› To make more published research true, practices that have improved credibility and efficiency in specific fields may be transplanted to others which would benefit from them—possibilities include the adoption of large-scale collaborative research; replication culture; registration; sharing; reproducibility practices; better statistical methods; standardization of definitions and analyses; more appropriate (usually more stringent) statistical thresholds; and improvement in study design standards, peer review, reporting and dissemination of research, and training of the scientific workforce.

Some terminology (1/2)

From a U.S. NSF subcommittee:

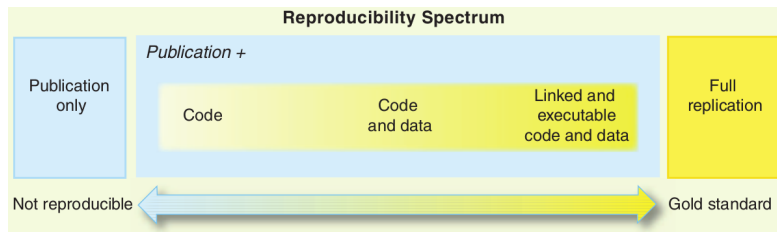
Reproducibility refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator. That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results. Reproducibility is a minimum necessary condition for a finding to be believable and informative.

Some terminology (2/2)

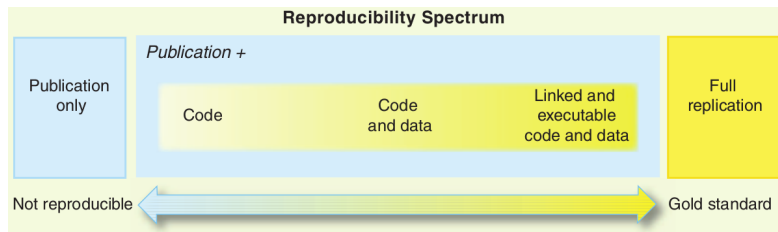
Goodman et al (2016):

- ▶ methods reproducibility: ability to implement, as exactly as possible, the experimental and computational procedures, with the same data and tools, to obtain the same results
 - ▶ capture the original meaning of “reproducibility”
 - ▶ here, I focus on this notion
- ▶ results reproducibility: production of corroborating results in a new study, having followed the same experimental methods
 - ▶ refers to what was previously described as “replication”

Reprod. Research in Computational Science (Peng, 2011)

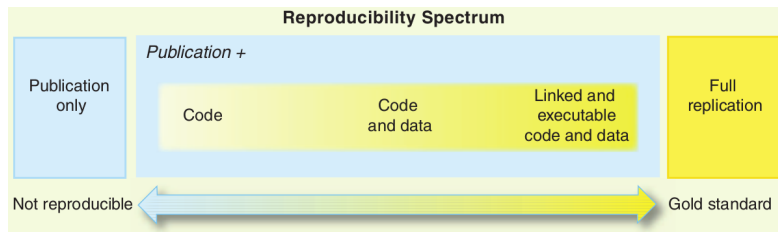


Reprod. Research in Computational Science (Peng, 2011)



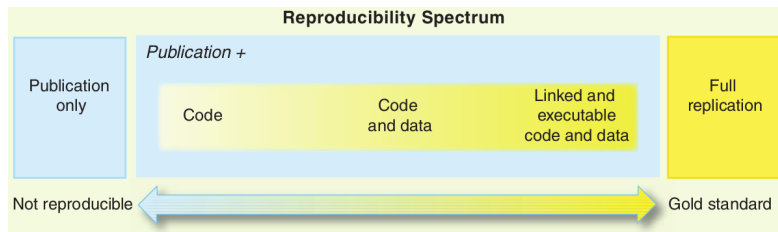
- ▶ Would you define yourself as a “computational scientist”?

Reprod. Research in Computational Science (Peng, 2011)



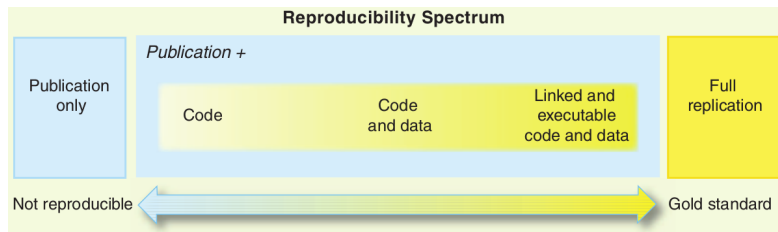
- ▶ Would you define yourself as a “computational scientist”?
 - ▶ Do you use Excel?

Reprod. Research in Computational Science (Peng, 2011)



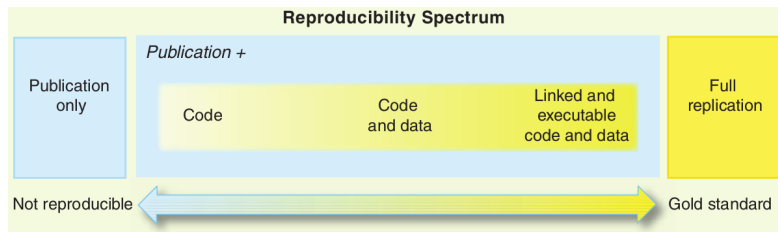
- ▶ Would you define yourself as a “computational scientist”?
 - ▶ Do you use Excel? R and/or Python?

Reprod. Research in Computational Science (Peng, 2011)



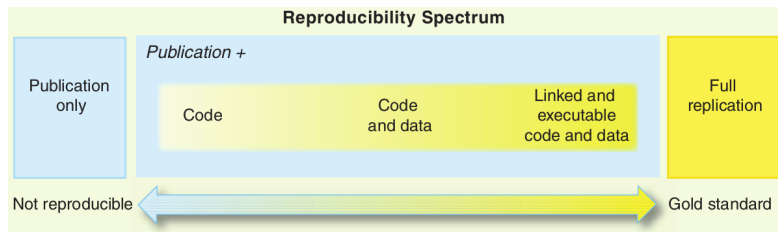
- ▶ Would you define yourself as a “computational scientist”?
 - ▶ Do you use Excel? R and/or Python? Julia?

Reprod. Research in Computational Science (Peng, 2011)



- ▶ Would you define yourself as a “computational scientist”?
 - ▶ Do you use Excel? R and/or Python? Julia? C/C++?

Reprod. Research in Computational Science (Peng, 2011)



- ▶ Would you define yourself as a “computational scientist”?
 - ▶ Do you use Excel? R and/or Python? Julia? C/C++?
- ▶ Where are you along the spectrum?

Let us start with code...

Claerbout & Karrenbach (1992):

An article about a computational result is advertising, not scholarship. The actual scholarship is the full software environment, code and data, that produced the result.

Let us start with code...

Claerbout & Karrenbach (1992):

An article about a computational result is advertising, not scholarship. The actual scholarship is the full software environment, code and data, that produced the result.

Buckheit & Donohoe (1995):

Publishing figures or results without the complete software environment could be compared to a mathematician publishing an announcement of a mathematical theorem without giving the proof.

... and now data.

Where Have All the Crop Phenotypes Gone?

Dani Zamir 

Published: June 25, 2013 • DOI: [10.1371/journal.pbio.1001595](https://doi.org/10.1371/journal.pbio.1001595)

... and now data.

Where Have All the Crop Phenotypes Gone?

Dani Zamir 

Published: June 25, 2013 • DOI: 10.1371/journal.pbio.1001595

Currently, virtually none of the data generated from the hundreds of phenotypic studies conducted each year are being made publically available as raw data; thus there is little we can learn from past experience when making decisions about how to breed better crops for the future.

... and now data.

Where Have All the Crop Phenotypes Gone?

Dani Zamir 

Published: June 25, 2013 • DOI: 10.1371/journal.pbio.1001595

Currently, virtually none of the data generated from the hundreds of phenotypic studies conducted each year are being made publically available as raw data; thus there is little we can learn from past experience when making decisions about how to breed better crops for the future.

Zamir (Science, 2014):

without the corresponding potentially commercially valuable phenotypic data. For example, in rice (*Oryza sativa*), which feeds roughly half the world population, 3000 variants from 89 countries were sequenced revealing 18.9 million single-nucleotide polymorphisms (SNPs) (4). But what good are 3000 genomes if the associated phenotypic data, and sometimes seed stocks, are kept proprietary? A wake-up call is needed for scientists, granting agencies, journal editors, and referees: What we eat are phenotypes, and seriously addressing global food security demands that, at least in the domain of crop plants, phenotypic data should be shared between scientists in the same manner as for sequences (3).

An extreme view: what do you think?



The NEW ENGLAND
JOURNAL of MEDICINE

HOME

ARTICLES & MULTIMEDIA ▾

ISSUES ▾

SPECIALTIES & TOPICS ▾

FOR AUTHORS

EDITORIAL

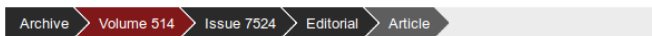
Data Sharing

Dan L. Longo, M.D., and Jeffrey M. Drazen, M.D.

N Engl J Med 2016; 374:276-277 | [January 21, 2016](#) | DOI: 10.1056/NEJMe1516564

"research parasites" [...] people who had nothing to do with the design and execution of the study but use another group's data for their own ends, possibly stealing from the research productivity planned by the data gatherers, or even use the data to try to disprove what the original investigators had posited

Incentives, to be improved



NATURE | EDITORIAL



Code share

Papers in Nature journals should make computer code accessible where possible.

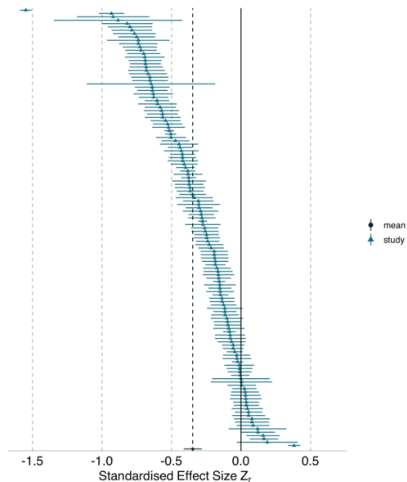
29 October 2014

Publication is conditional upon the agreement of the authors to make freely available any materials and information described in their publication that may be reasonably requested by others.

Data Availability

PLOS journals require authors to make all data underlying the findings described in their manuscript fully available without restriction, with rare exception¹.

But that's not it!



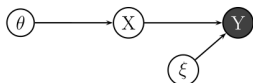
<https://ecoevorxiv.org/repository/view/6000/>

You shall document your preprocessing

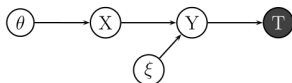
Blocker & Meng (Bernoulli, 2013):

Decisions made in preprocessing constrain all later analyses and are typically irreversible. Hence, data analysis becomes a collaborative endeavor by all parties involved in data collection, preprocessing and curation, and downstream inference.

Preprocessor's model



Downstream analyst's model



Effort towards more formalization

Devezer et al (RSOS, 2021):

Formal approach to solving methodological problems.

0. **Conception.** An informal problem statement and a proposed solution to that problem, often expressed non-technically.
1. **Definitions.** Identification of variables, population parameters, and constants involved in the problem, and statistical model building using these quantities, with explicitly stated model assumptions.
2. **Formal problem statement.** Mathematical propositions or algorithms positing methodological claims.
3. **Formal result.** Mathematical or simulation-based proofs that interrogate the validity of the statements in step 2.
- 4a. **Demonstrations.** If the statements are valid, examples showing their relevance in application.

After all, why is reproducibility important

Which reason(s) would you choose?

After all, why is reproducibility important

Which reason(s) would you choose?

- ▶ reproducing work is also the first step to extending it
- ▶ we are forgetful, error-prone (or dishonest)
- ▶ helps communication with your collaborators
- ▶ bigger visibility in the community
- ▶ we are mostly funded by public money
- ▶ ...

Outline

Global context

Spectrum of solutions

Multi-dimensional space of "projects"

- ▶ **question**: mono- or inter-disciplinary; closer to applied or to basic research
 - ▶ won't be discussed here, but obviously crucial...

Multi-dimensional space of "projects"

- ▶ **question**: mono- or inter-disciplinary; closer to applied or to basic research
 - ▶ won't be discussed here, but obviously crucial...
- ▶ **data**: few small files or many large files; text or binary; custom or standardized formats; databases

Multi-dimensional space of "projects"

- ▶ **question**: mono- or inter-disciplinary; closer to applied or to basic research
 - ▶ won't be discussed here, but obviously crucial...
- ▶ **data**: few small files or many large files; text or binary; custom or standardized formats; databases
- ▶ **code**: in interpreted languages (e.g. R, Python, Julia, Perl, Bash) or compiled languages (e.g. C, C++, Fortran)

Multi-dimensional space of "projects"

- ▶ **question**: mono- or inter-disciplinary; closer to applied or to basic research
 - ▶ won't be discussed here, but obviously crucial...
- ▶ **data**: few small files or many large files; text or binary; custom or standardized formats; databases
- ▶ **code**: in interpreted languages (e.g. R, Python, Julia, Perl, Bash) or compiled languages (e.g. C, C++, Fortran)
- ▶ **model**: informal (any word processor is enough, e.g. Writer, Word) or formal (many equations, much easier in LaTeX)

Multi-dimensional space of "projects"

- ▶ **question**: mono- or inter-disciplinary; closer to applied or to basic research
 - ▶ won't be discussed here, but obviously crucial...
- ▶ **data**: few small files or many large files; text or binary; custom or standardized formats; databases
- ▶ **code**: in interpreted languages (e.g. R, Python, Julia, Perl, Bash) or compiled languages (e.g. C, C++, Fortran)
- ▶ **model**: informal (any word processor is enough, e.g. Writer, Word) or formal (many equations, much easier in LaTeX)
- ▶ **collaborators**: professional scientist or anyone else; beginner or experienced; "open curious" or already fully proficient (N.B.: to specify author contributions, take a look at CRediT)

Initial steps toward reproducible research

Karl Broman's tutorial: <http://kbroman.org/steps2rr/>

1. Everything with a script
2. Organize your data and code
3. Automate the process
4. Turn scripts into reproducible reports
5. Turn repeated code into functions
6. Package functions for reuse
7. Use version control

Initial steps toward reproducible research

Karl Broman's tutorial: <http://kbroman.org/steps2rr/>

1. Everything with a script
2. Organize your data and code
3. Automate the process
4. Turn scripts into reproducible reports
5. Turn repeated code into functions
6. Package functions for reuse
7. Use version control

Teach yourself: Software Carpentry, MOOC RR, forum RR

A text editor, not a word processor

- ▶ word processors: LibreOffice Writer, Microsoft Word, etc
- ▶ text editors: Emacs, Vim, Notepad++, Gedit, Geany, etc

A text editor, not a word processor

- ▶ word processors: LibreOffice Writer, Microsoft Word, etc
- ▶ text editors: Emacs, Vim, Notepad++, Gedit, Geany, etc

Why **knowing/mastering a good text editor is crucial?**

- ▶ essential tool of any data analysis (source code and scripts)
- ▶ light-weight markup languages convert plain text to PDF/HTML/odt/docx/etc, check out pandoc
- ▶ delivers full power of text manipulation tools, e.g. `diff` to compare, `grep` to search, `awk` to extract, ...
- ▶ allows efficient use of version control systems such as `git`

Reproducible reports

Literate programming (Knuth, 1984): a computer program is given as an explanation of how it works in a natural language, interspersed (embedded) with traditional source code, from which compilable source code can be generated.

Reproducible reports

Literate programming (Knuth, 1984): a computer program is given as an explanation of how it works in a natural language, interspersed (embedded) with traditional source code, from which compilable source code can be generated.

```
1 ~~~
2 title: "Test"
3 author: "Timothée Flutre"
4 date: "2023-11-08"
5 output: html_document
6 ~~~
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ~~~
11
12 Draw from a Normal distribution:
13 ```{r}
14 x = rnorm(n=100, mean=0, sd=1)
15 ~~~
16
17 Plot a histogram:
18 ```{r}
19 hist(x)
20 ~~~
```

Test

Timothée Flutre

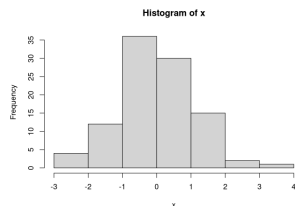
2023-11-08

Draw from a Normal distribution:

```
x = rnorm(n=100, mean=0, sd=1)
```

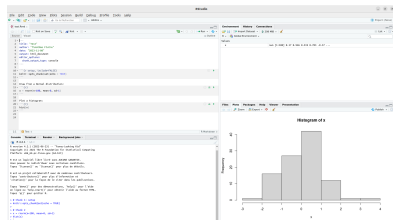
Plot a histogram:

```
hist(x)
```

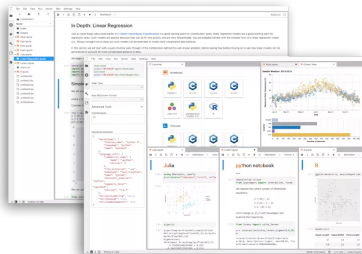


Notebook interfaces

<https://posit.co/products/open-source/rstudio/>



<https://jupyter.org/>



Wrap your code into functions, and use unit tests

Start by writing a test (e.g., setting 2 to 0), here in R:

```
1 input <- c(1, 2, 3, 4)
2 expected <- c(1, 0, 3, 4)
```

Wrap your code into functions, and use unit tests

Start by writing a test (e.g., setting 2 to 0), here in R:

```
1 input <- c(1, 2, 3, 4)
2 expected <- c(1, 0, 3, 4)
```

Then implement the function:

```
1 setTwoToZero <- function(input){
2   output <- input
3   output[input == 2] <- 0
4   return(output)
5 }
```

Wrap your code into functions, and use unit tests

Start by writing a test (e.g., setting 2 to 0), here in R:

```
1 input <- c(1, 2, 3, 4)
2 expected <- c(1, 0, 3, 4)
```

Then implement the function:

```
1 setTwoToZero <- function(input){
2   output <- input
3   output[input == 2] <- 0
4   return(output)
5 }
```

Check that the test passes; until then, fix the implementation:

```
1 observed <- setTwoToZero(input)
2 library(testthat)
3 testthat::expect_equal(observed, expected)
```

Gather all your R functions into a package

Research funding being more and more project-based, it is frequent to need the same kind of analysis in different projects. But don't copy-paste your code, gather it into an R package instead.

- ▶ read H. Wickham's great book (freely available online)

Then, version it with git, host it somewhere (e.g., locally with GitLab, online with SourceSup, GitHub, etc), and use the R package devtools to install it easily on any computer.

Make a R package in 1 min

```
1 library(usethis)
2 library(devtools)
```

```
1 pkg_name <- "mypkg"
2 pkg_dir <- path.expand("~/Documents")
3 path_to_pkg <- file.path(pkg_dir, pkg_name)
4 pkg_fields <- list()
5 pkg_fields[["Package"]] <- pkg_name
6 pkg_fields[["Version"]] <- "0.1.0"
7 pkg_fields[["Title"]] <- "My First package"
8 pkg_fields[["Description"]] <- "This package is my first one."
9 pkg_fields[["Authors@R"]] <- "person('Timothee', 'Flutre', '', 'timothee.
    flutre@inrae.fr', c('aut', 'cre'))"
10 pkg_fields[["License"]] <- "AGPL-3"
11 usethis::create_package(path=path_to_pkg, fields=pkg_fields)
```

```
1 devtools::document(pkg=path_to_pkg)
2 built_pkg <- devtools::build(pkg=path_to_pkg)
3 devtools::check(pkg=path_to_pkg)
4 devtools::install_local(path=built_pkg)
```

And now add your functions in a R subdirectory, along with roxygen2 documentation.

Project organization

Follow the spirit of Noble (PLoS Comput Biol, 2009):

- ▶ someone unfamiliar with your project should be able to look at your computer files and understand in detail what you did and why;
- ▶ everything you do, you will probably have to do it over again.

Project organization

Follow the spirit of Noble (PLoS Comput Biol, 2009):

- ▶ someone unfamiliar with your project should be able to look at your computer files and understand in detail what you did and why;
- ▶ everything you do, you will probably have to do it over again.

1. choose a short project name and create a directory
2. give brief explanations in a text file named README
3. describe sharing/modifying rights in a text file named COPYING or LICENSE
4. list authors in a text file named AUTHORS
5. create subdirectories doc/, data/, src/ and results/

Keep track of code changes, and collaborate

Karl Broman on version control systems (VCS):

- ▶ not strictly necessary for *reproducibility*, but can be hugely useful for *sanity*
- ▶ requires a big initial investment in time and effort, but become a natural part of your daily workflow after a month or so
- ▶ huge *short-term* advantages in collaborative projects (keeping in sync, merging simultaneous changes)

Keep track of code changes, and collaborate

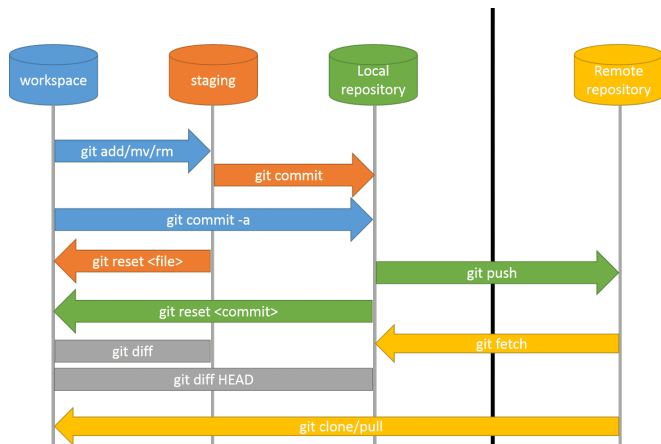
Karl Broman on version control systems (VCS):

- ▶ not strictly necessary for *reproducibility*, but can be hugely useful for *sanity*
- ▶ requires a big initial investment in time and effort, but become a natural part of your daily workflow after a month or so
- ▶ huge *short-term* advantages in collaborative projects (keeping in sync, merging simultaneous changes)

My advice: use **git** (free software; multi-platform; used all over the globe by programmers, scientists, journalists, etc)

- ▶ follow a tutorial online
- ▶ download git and install it
- ▶ *read* chapters 1 to 3 of the official book

Git workflow



Diego C. Martin

Two case studies

1. **project "light"**: a few small data files (most in plain text); laptop; classical models; available implementations for interpreted languages
 - ▶ ok to re-run the whole analysis from time to time
 - ▶ write notebooks in text file in Rmd format and use RStudio
 - ▶ example: github.com/timflutre/tuto-reproducible-research

Two case studies

1. **project "light"**: a few small data files (most in plain text); laptop; classical models; available implementations for interpreted languages
 - ▶ ok to re-run the whole analysis from time to time
 - ▶ write notebooks in text file in Rmd format and use RStudio
 - ▶ example: github.com/timflutre/tuto-reproducible-research
2. **project "heavy"**: many large files (notably in binary formats); cluster; possibly new models and implementations
 - ▶ can't re-run the whole analysis because of intensive computations
 - ▶ write notebooks as above, but develop also "real" software

Example of a “heavy” project

Usually, (very) large data are kept on a computer cluster, in a shared directory, e.g., `/bigdata`, itself organized in several subdirectories:

- ▶ `external_public/`, `external_private/`, `internal/`

Example of a “heavy” project

Usually, (very) large data are kept on a computer cluster, in a shared directory, e.g., /bigdata, itself organized in several subdirectories:

▶ external_public/, external_private/, internal/

1. create a git repository as before
2. clone it into your \$HOME on the cluster
3. do NOT copy the large data from /bigdata into
~/work/project-heavy/data/
4. instead make a symbolic link

```
1 ln -s /bigdata ~/work/project-heavy/data/
```

State-of-the art for dealing with code dependencies

Guix: a *purely functional* package management tool

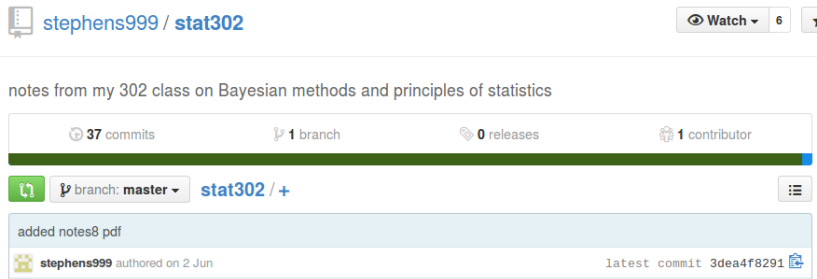
- ▶ support for transactional package upgrade and rollback, per-user installation, and garbage collection of packages
- ▶ maximize *build reproducibility* (bit-by-bit identical programs)
- ▶ linked to Software Heritage

For scientists:

- ▶ high-performance computing: GuixHPC
- ▶ mailing list: guix-science

Are you teaching?

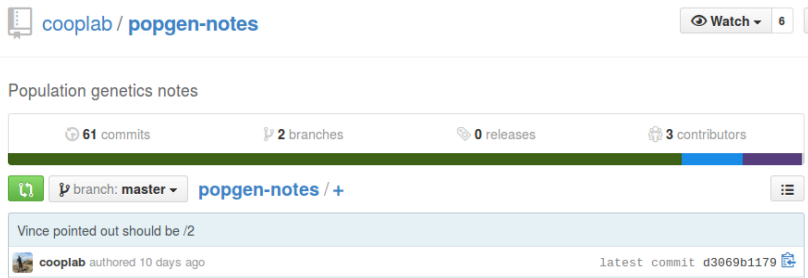
Example in **statistics** from Matthew Stephens: class notes versioned with git and hosted on GitHub



The screenshot shows the GitHub repository page for 'stephens999 / stat302'. At the top right, there is a 'Watch' button with a dropdown arrow, a '6' in a box, and a '1' in a box. Below this, the repository name 'stephens999 / stat302' is displayed. The main content area shows the repository description: 'notes from my 302 class on Bayesian methods and principles of statistics'. Below the description, there are statistics: '37 commits', '1 branch', '0 releases', and '1 contributor'. A green progress bar is visible below these statistics. Below the progress bar, there is a green 'Refresh' button, a 'branch: master' dropdown, the repository name 'stat302 / +', and a 'Menu' button. Below this, a commit message 'added notes8 pdf' is shown, along with the author 'stephens999' and the date 'authored on 2 Jun'. At the bottom right of the commit section, it says 'latest commit 3dea4f8291' with a link icon.

Are you teaching? Cont'd

Example in **population genetics** from Graham Coop: class notes versioned with git and hosted on GitHub



The screenshot shows the GitHub interface for the repository 'cooplab / popgen-notes'. At the top right, there is a 'Watch' button with a dropdown arrow and the number '6'. Below the repository name, the title 'Population genetics notes' is displayed. A summary bar shows '61 commits', '2 branches', '0 releases', and '3 contributors'. Below this, there is a navigation bar with a green 'Refresh' button, a dropdown menu for 'branch: master', the repository name 'popgen-notes / +', and a hamburger menu icon. The main content area shows a commit message 'Vince pointed out should be /2' by user 'cooplab' from 10 days ago, with the latest commit hash 'd3069b1179' and a link to view the commit.

Are you writing an article?

Example in **plant biology** from Rubén Rellán Álvarez: article (text, figures) and software versioned with git and hosted on GitHub

 rr-lab / glo_roots

 Watch 3

Growth and Luminescence Observatory for Roots (GLO-Roots) http://www.rrlab.org/glo_roots


 116 commits

 3 branches

 2 releases

 2 contributors



 branch: master


glo_roots / +



bump ...

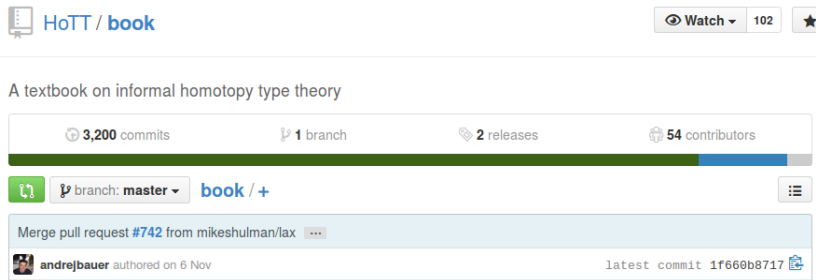


rellan authored on 7 May

latest commit 14089af658 

Are you writing a book?

Example in **mathematics** from the "Homotopy Type Theory" group which started at Princeton (IAS) and now writes a book versioned with git and hosted on GitHub



The screenshot shows the GitHub repository page for **HoTT / book**. At the top right, there are buttons for "Watch" (with 102 watchers) and a star icon. Below the repository name, the description reads: "A textbook on informal homotopy type theory". A progress bar shows the repository's activity, with a green segment for commits and a blue segment for releases. Below the progress bar, there are statistics: 3,200 commits, 1 branch, 2 releases, and 54 contributors. The current branch is **master**, and the repository name is **book / +**. A recent pull request is visible: "Merge pull request #742 from mikeshulman/lax", authored by **andrejbauer** on 6 Nov. The latest commit is **1f660b8717**.

Take-home message

Whatever the exact tools you are using, it is the spirit that is important!

Take-home message

Whatever the exact tools you are using, it is the spirit that is important!

- ▶ How well does your behavior favor others to understand and build on your work?
- ▶ When and how will you start?