version control with R

R buddy, FS2023

Content

- what is version control?
- install version control
- guided tutorial
- exercises
- Take home/ Reflection

what is version control?
and why should I care about?

why care?

SORTEE

Society for Open, Reliable, and Transparent Ecology and Evolutionary Biology

The replication crisis is good for science

Published: April 8, 2019 12.44pm CEST

d scrutiny. PORTRAIT IMAGES ASIA BY NONWARIT/shutterstock.com

is in the midst of a crisis: A surprising fraction of published studies fail to
when the procedures are repeated.

nple, take the study, published in 2007, that claimed that tricky math
is requiring careful thought are easier to solve when presented in a fuzzy
ıen researchers found in a small study that using a fuzzy font improved
ance accuracy, it supported a claim that encountering perceptual
es could induce people to reflect more carefully.

Open Science Isn't Always Open to All Scientists

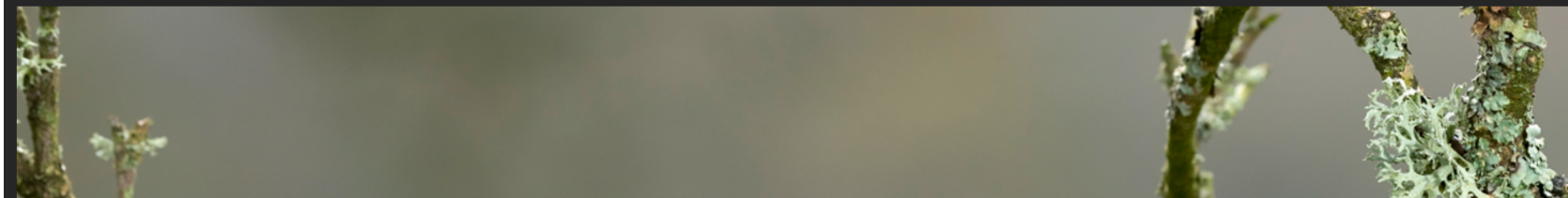
BY CHRISTIE BAHLAI, LEWIS J. BARTLETT, KEVIN R. BURGIO, AURIEL M. V. FOURNIER, CARL N. KEISER, TIMOTHÉE POISOT, KAITLIN STACK WHITNEY

Current efforts to make research more accessible and

Psychology's replication crisis inspires ecologists to push for more reliable research

New society aims for transparency and culture change in ecology

9 DEC 2020 • BY CATHLEEN O'GRADY

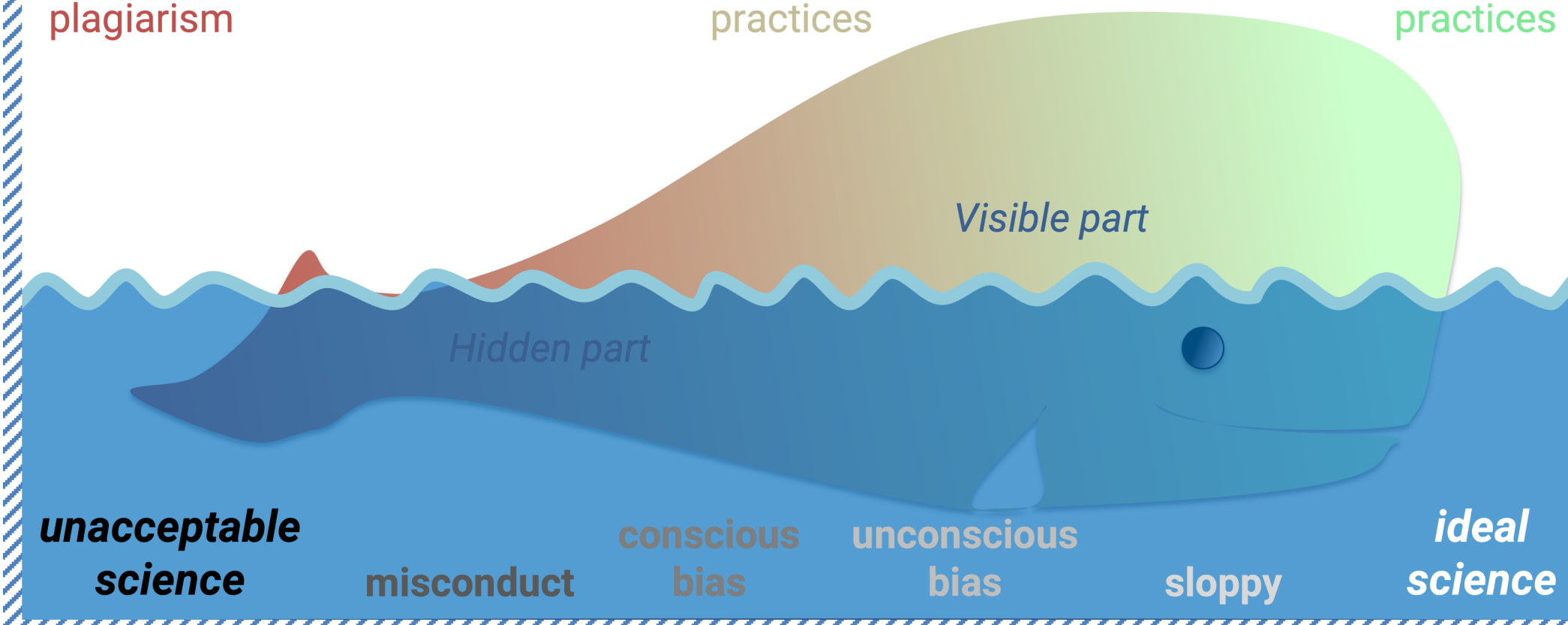


THE RESEARCH WHALE

fabrication
falsification
plagiarism

questionable
research
practices

good
research
practices



unacceptable science

misconduct

conscious bias

unconscious bias

sloppy

ideal science

>WHY OPEN CODE IN RESEARCH?

SORTEE

RESEARCHERS STUDENTS

- **Learning** new computational methods
- **Sharing** verified resources
- **Collaborating** efficiently
- **Reproducing** statistical analyses
- Improving **employability**

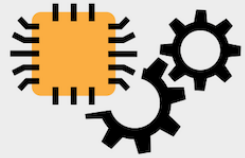
FUNDERS

- Reducing **costs**
- Improving research **efficiency**
- Increasing **transparency and integrity**
- Facilitating project **evaluations**
- **Showcasing** best projects

JOURNALS PUBLISHERS

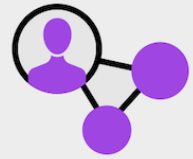
- Increasing **impact** of papers
- Improving journal's **metrics/ranking**
- Improving **peer-review** process
- Ensuring **integrity**
- Becoming **role models** in open science

PRACTICIONERS AGENCIES



- Facilitating **industry applications**
- Encouraging **collaborations**
- **Leading change** to best practice
- Supporting **R&D** information transfer
- **Repurposing** code

GENERAL PUBLIC



- **Building trust** in science
- Encouraging **public access**
- Delivering robust, science-based **solutions**
- Promoting public **computational literacy**
- Reusing code in **citizen-science**

better science

https://betterscience.ch/#/

- Highlight extra-academic work
- Take your time to think
- Put quality before quantity
- Prioritise thoughtfully
- Enable a healthy academic culture

University Library of Bern UB

https://www.ub.unibe.ch/services/open_science/services/index_eng.html

< Open Science

Services

Please contact us if you have any questions about Open Science. openscience@unibe.ch

Universitätsbibliothek Bern

Bern Open Science Blog https://wkblog.ub.unibe.ch/

Information and Training | Support

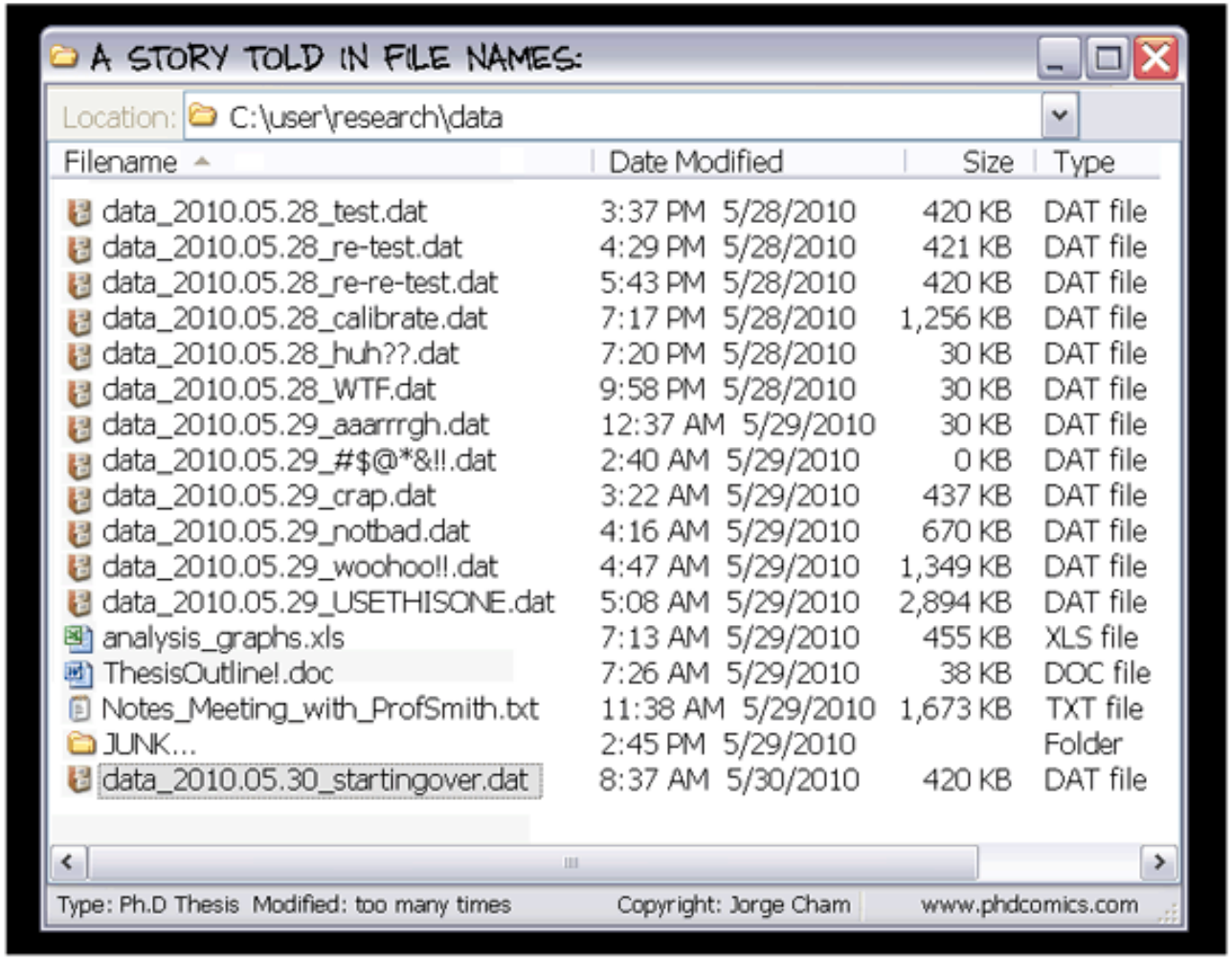
Start | Über uns | Kontakt

ChatGPT und Autorschaft

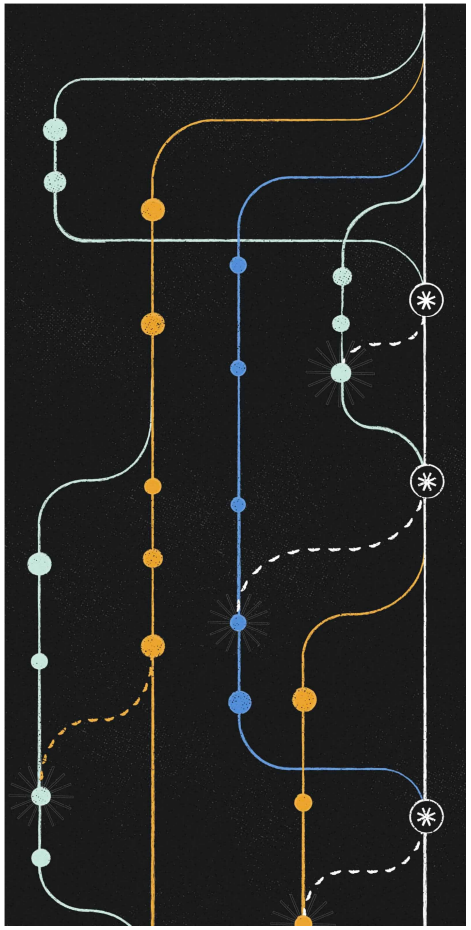
Published on 3. April 2023 by Julia Wermelinger Category: Beiträge

Zusammen mit vielen neuen Studierenden startete auch ChatGPT in sein erstes

purposes of version control

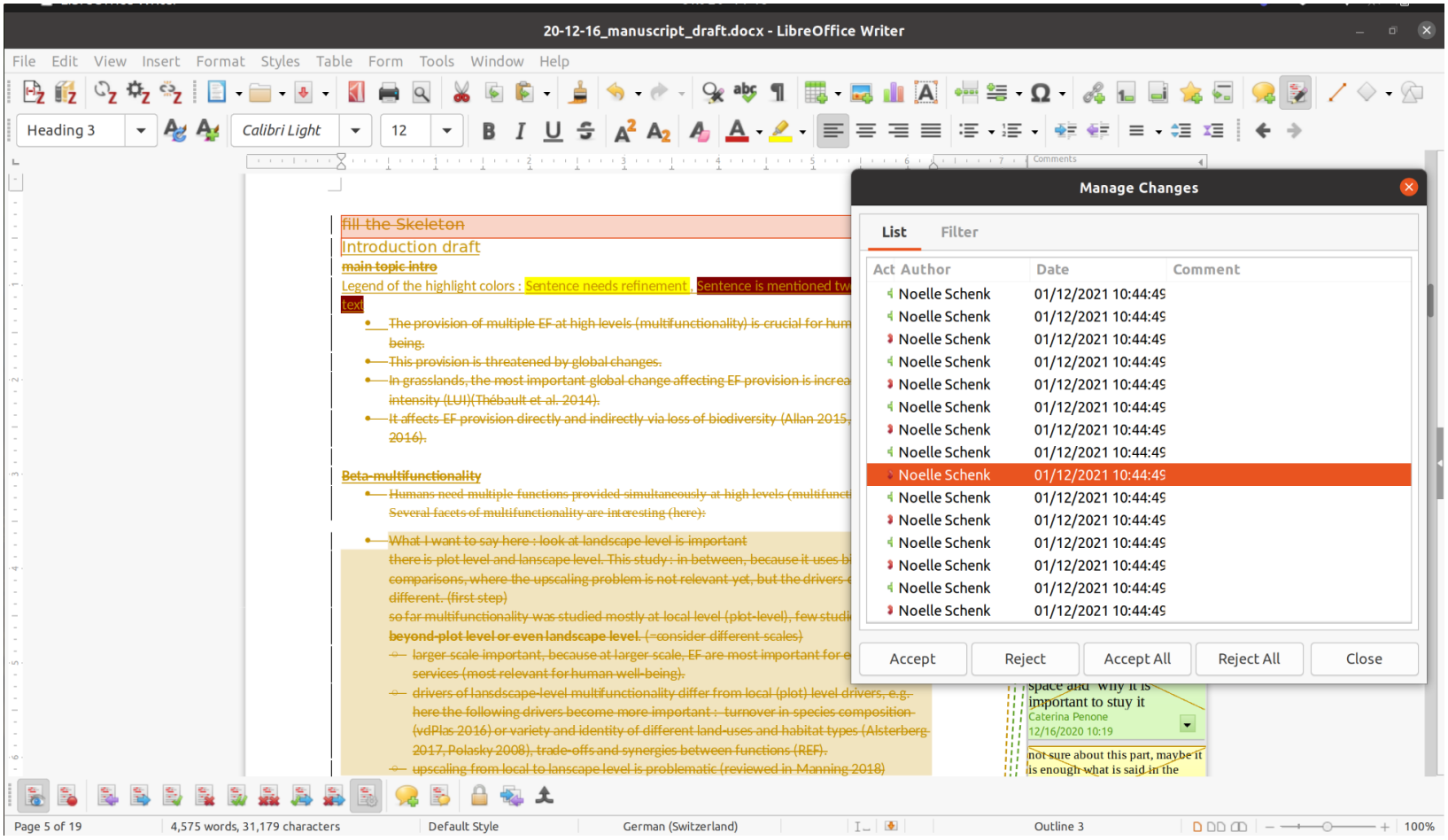


purposes of version control



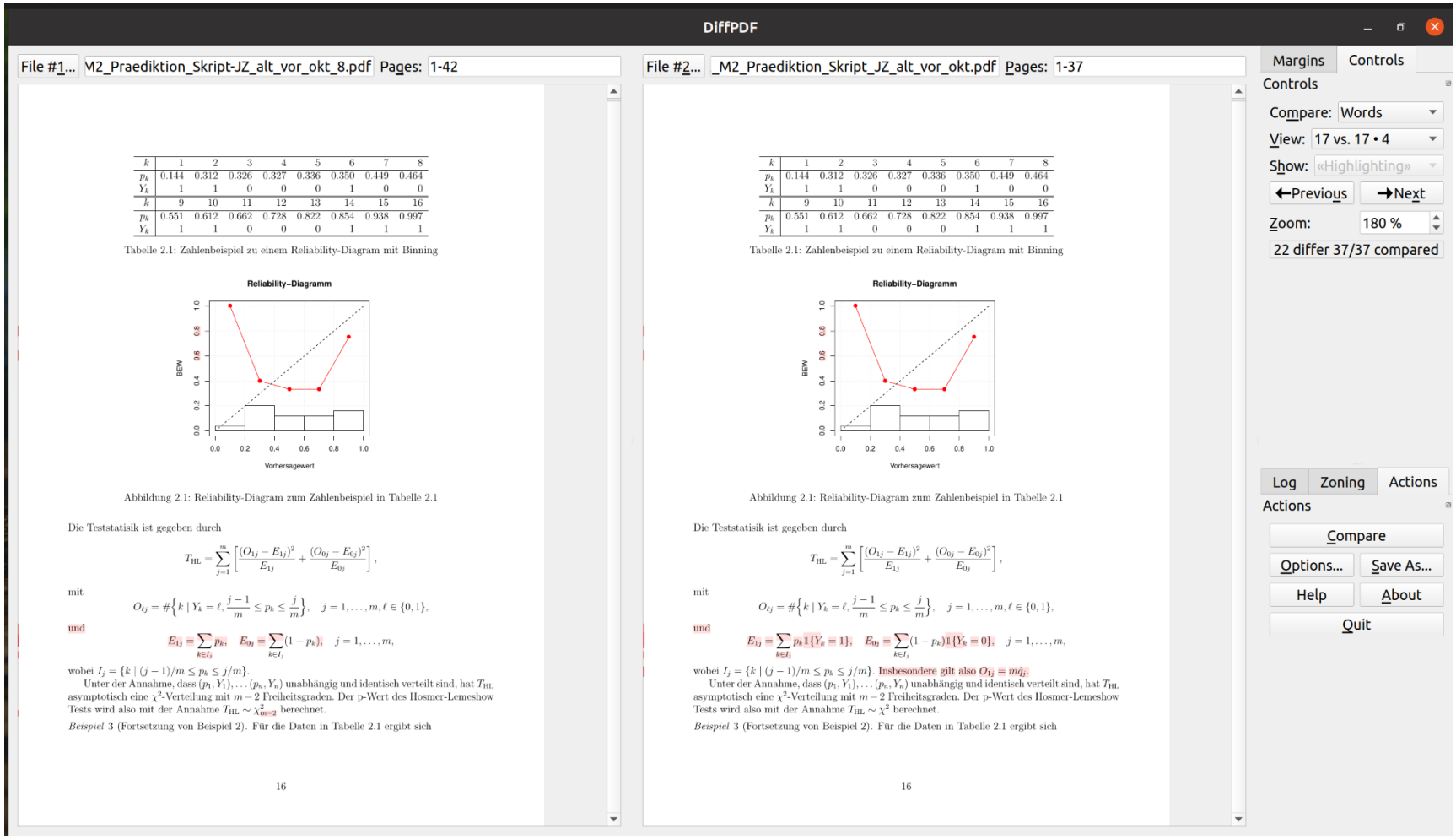
- keep track of changes
- keep scripts clear and structured
- understand previous changes
- compare different versions
- facilitate collaboration
- short- and long-term undo

what is version control?



- Track Changes is a “one time version control”
- Edit > Track Changes > Compare document
- No commenting, just two documents

what is version control?



- Comparing two documents (e.g. pdfs) creates a “diff” file
- Shows what’s different between files
- Version control for code : saves only first version and all following “diffs”

Version Control for code

git

SUBVERSION

Etymology 1 [edit]

From Middle English *get* (“[illegitimate] offspring”). A southern variant of Scots *get* (“illegitimate child, brat”), related to *beget*.[1]

Noun [edit]

git (*plural* **gits**)

1. (*Britain*, *slang*, *derogatory*) A silly, incompetent, stupid, annoying or childish person (usually a man). [quotations ▼]

Usage notes [edit]

- *Git* is usually used as an insult, more severe than *twit* but less severe than a true profanity like *wanker* or *arsehole*, and may often be used affectionately between friends. *Get* can also be used, with a subtle change of meaning. "You cheeky get!" is slightly less harsh than "You cheeky git!".
- *Git* is frequently used in conjunction with another word to achieve a more specific meaning. For instance a "smarmy git" refers to a person of a slimy, ingratiating disposition; a "jammy git" would be a person with undeserved luck. The phrase "grumpy old git", denoting a cantankerous old man, is used with particular frequency.
- In parts of northern England, Northern Ireland and Scotland, *get* is still used in preference to *git*. In the Republic of Ireland, *get*, rather than *git* is used.
- The word has been ruled by the Speaker of the House of Commons to be unparliamentary language.[2][3]

Translations [edit]

± A silly, incompetent, stupid, annoying or childish person [show ▼]

Version Control for code

git

SUBVERSION

Etymology 1 [edit]

From Middle English *get* (“[illegitimate] offspring”). A southern variant of Scots *get* (“illegitimate child, brat”), related to *beget*.[1]

Noun [edit]

git (*plural* **gits**)

1. (*Britain*, *slang*, *derogatory*) A silly, incompetent, stupid, annoying or childish person (usually a man). [quotations ▼]

Usage notes [edit]

- *Git* is usually used as an insult, more severe than *twit* but less se
  Get can also be used, with a subtle change of meaning. "You che
- *Git* is frequently used in conjunction with another word to achiev
  disposition; a "jammy git" would be a person with undeserved luc
- In parts of northern England, Northern Ireland and Scotland, *get* i
- The word has been ruled by the Speaker of the House of Commo

Translations [edit]

± A silly, incompetent, stupid, annoying or childish person

```
GIT - the stupid content tracker

"git" can mean anything, depending on your mood.

 - random three-letter combination that is pronounceable, and not
   actually used by any common UNIX command.  The fact that it is a
   mispronunciation of "get" may or may not be relevant.
 - stupid. contemptible and despicable. simple. Take your pick from the
   dictionary of slang.
 - "global information tracker": you're in a good mood, and it actually
   works for you. Angels sing, and a light suddenly fills the room.
 - "goddamn idiotic truckload of sh*t": when it breaks

This is a stupid (but extremely fast) directory content manager.  It
doesn't do a whole lot, but what it _does_ do is track directory
contents efficiently.
```

Version Control for code

plot overviewbars of multiple models : added option for dodged barplo… Browse files

…ts for better overview

master

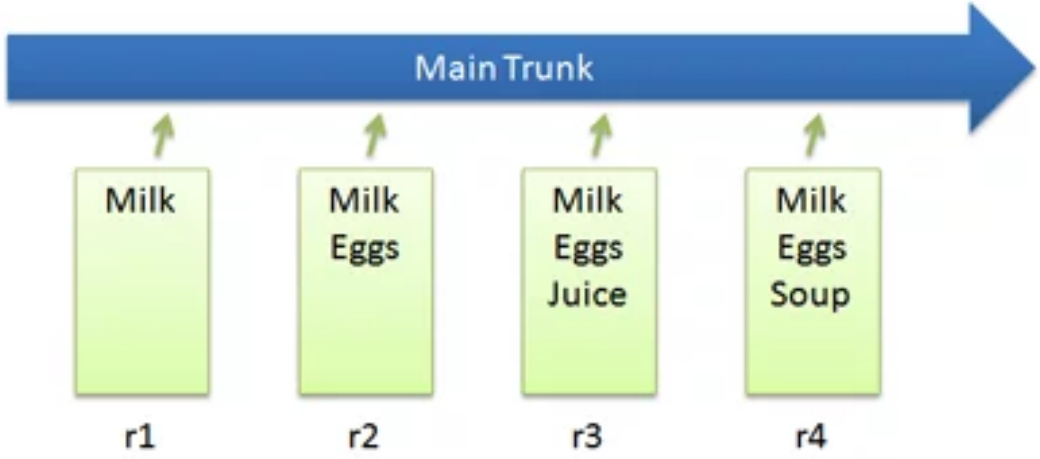
noschenk committed 19 days ago 1 parent 6eb43e4 commit aacc9b9df485ec99079b966b69e38c2773b8c5b1

```
271  273    #' @export
272  274    ### FUNCTION
273       - create_single_funs_overviewbar_plot <- function(singleF_restab, legend = F){
     275  + create_single_funs_overviewbar_plot <- function(singleF_restab, legend = F, pos = "stack"){
274  276      if(legend){
275  277      # return a plot with legend
276  278        print("not implemented yet.")
277  279      } else {
278  280        sf_ov <- ggplot(singleF_restab, aes(x = variable, y = value, fill = color)) +
279       -        geom_bar(stat = "identity", color = "black") +
     281  +        geom_bar(stat = "identity", color = "black", position = pos) +
280  282          scale_fill_identity("", labels = singleF_restab$ground,
281  283                              breaks = singleF_restab$color,  guide = "legend") +
282  284          # coord_flip() +
```

- Instead of saving versions of files, only save the “diff”
- Comment on every change
- Reconstruct any previous version as you need it

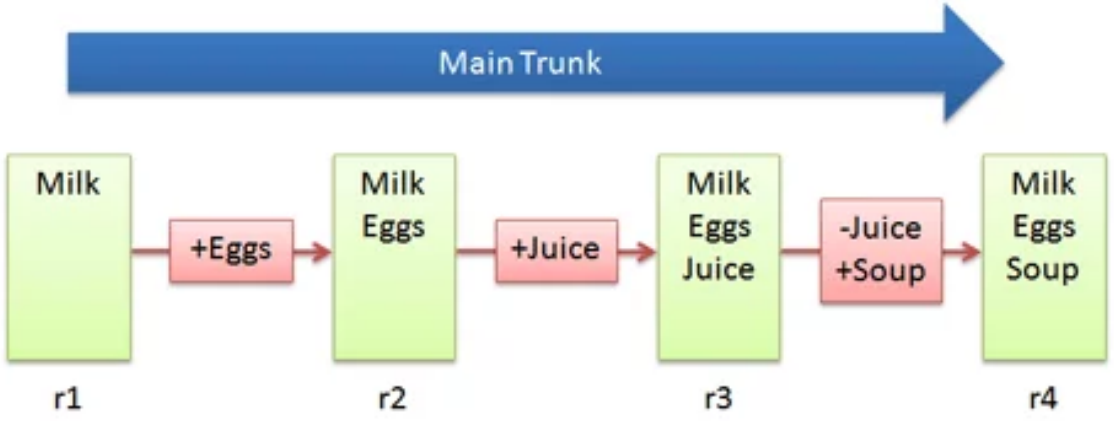
what is version control?

- easiest case : the “project” just consists of one file which experiences changes :

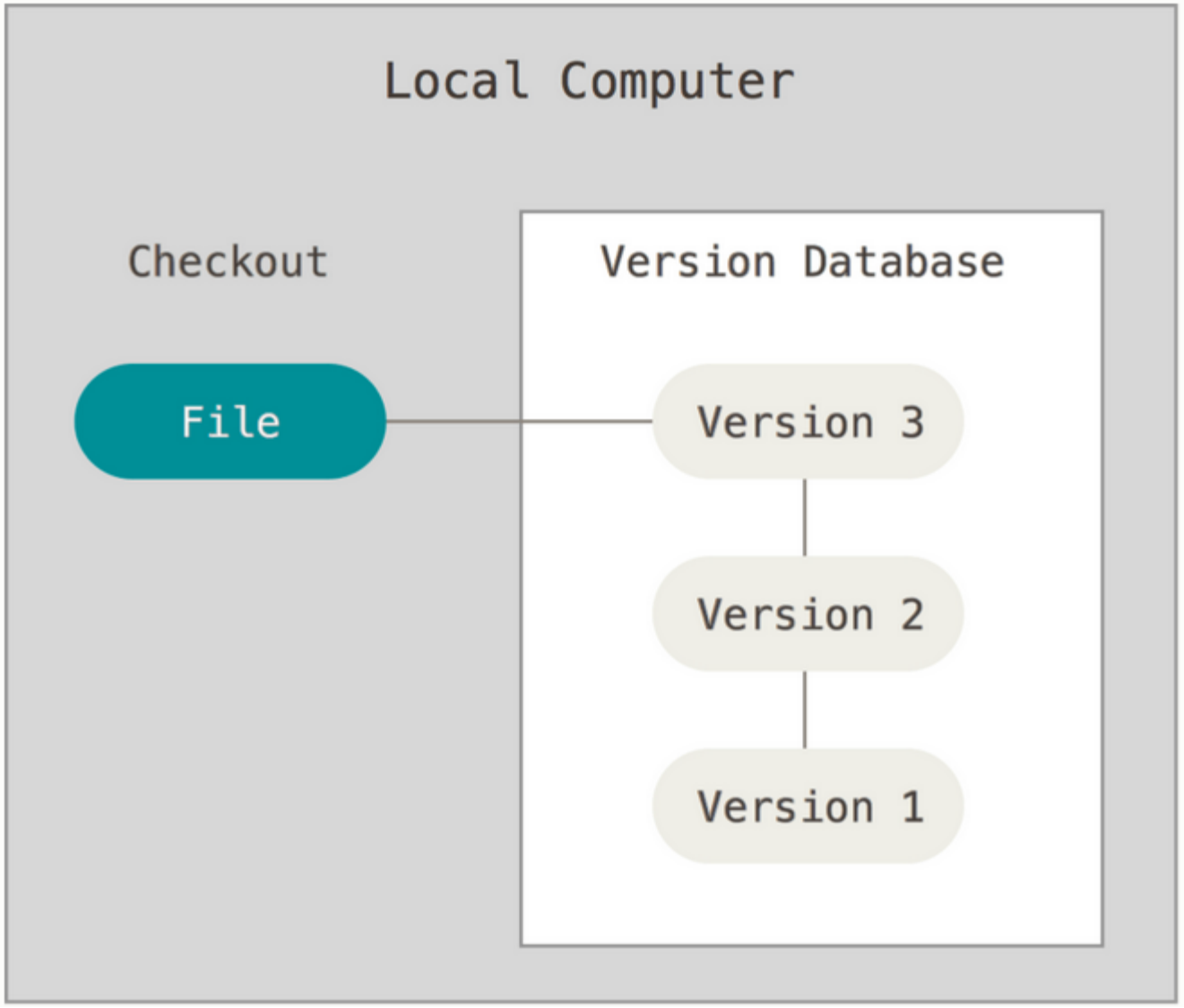


what is version control?

- easiest case : the “project” just consists of one file which experiences changes
- diffs : Are a precise description of the changes made to a file.
- The changes to your file are stored as diff : this does not use a lot of storage and is easier to look at (you don’t have to search where exactly in your file were the changes)

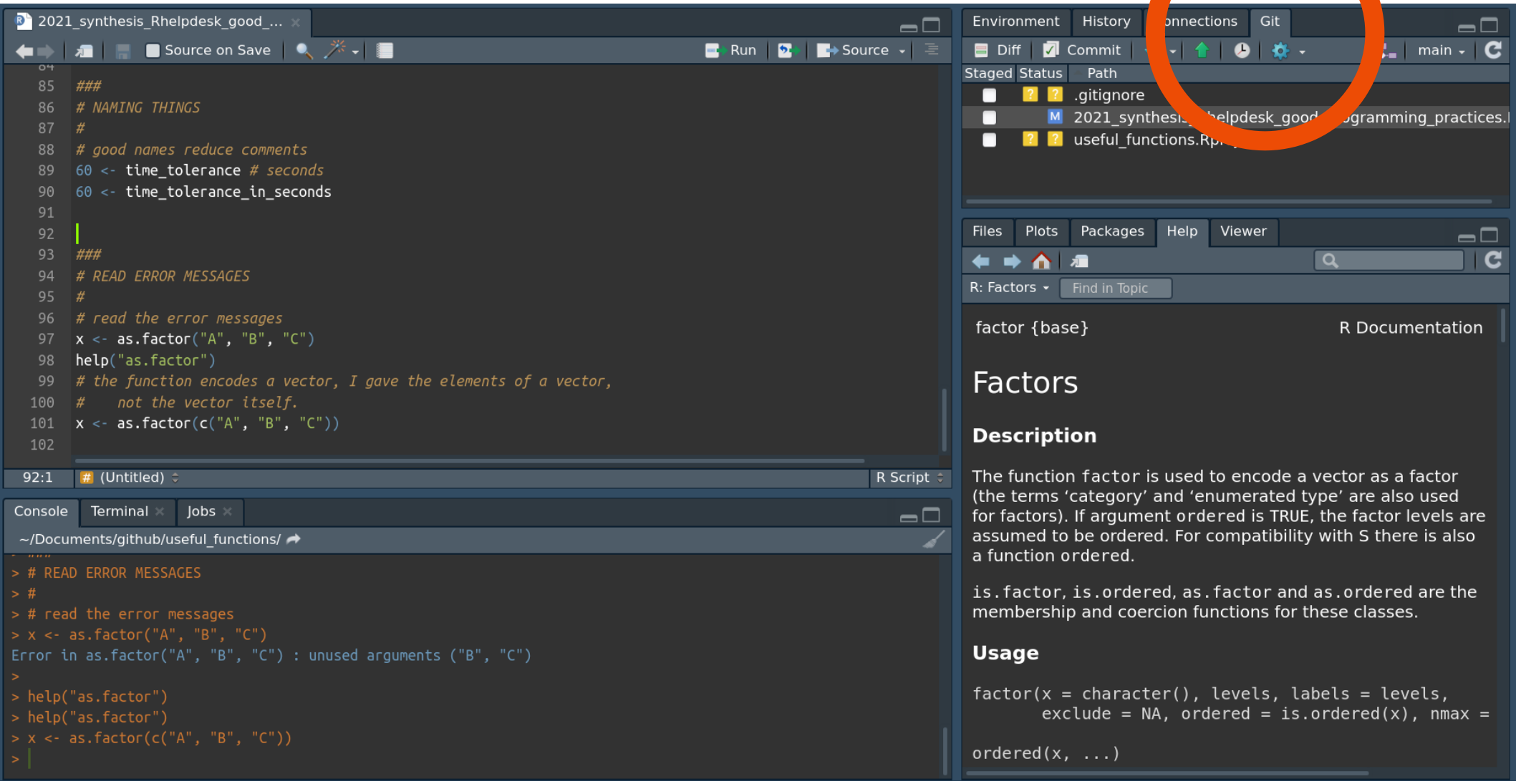


what is version control?



version control in RStudio

- git



Git

version control software

on local computer, integrates with RStudio

"does all the work"

https://git-scm.com/

GitHub

share unfinished code online

store repositories remotely

integrates with git and RStudio

update every day/ week

https://github.com/

alternatives

GitLab

sourceforge

gitbucket

bitbucket

phabricator

gitea

repository = the directory with your scripts in it

remotely = not in your local computer, but at a server in another place

zenodo

publish (finished) code

guarantees requirements for longterm code publishing
(nonproprietary, 10 years publish)

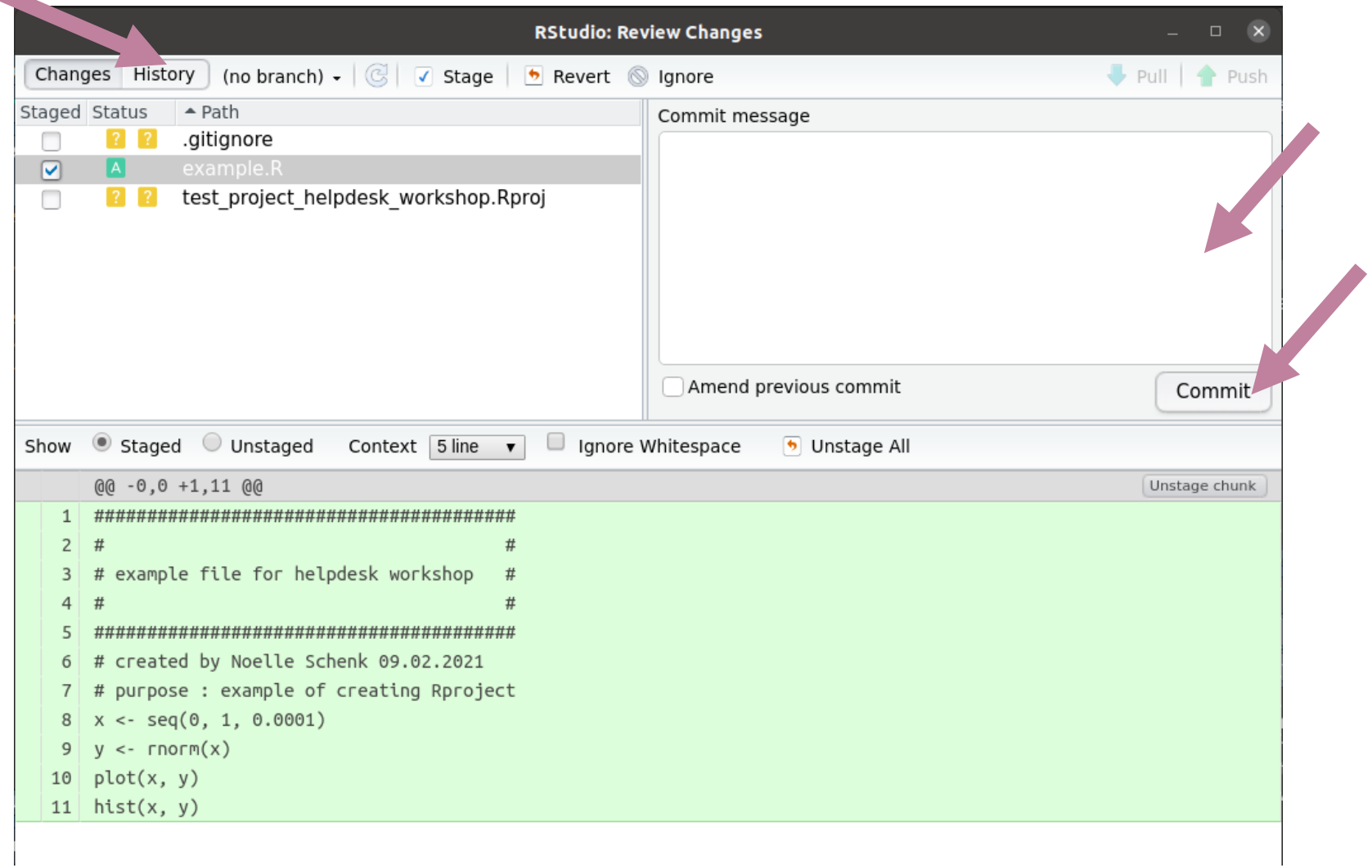
update every month/ year

integrates with GitHub

https://zenodo.org/

guiding through version control

- make a change, check the box, click commit, write commit message, commit, leave window and repeat



version control in RStudio

- git 

- manual to set up git and RStudio
  https://happygitwithr.com/big-picture.html
- requires your folder to be an Rproject
- looking at history : see the commits you have done in the past

Take home

Reflections

- Will you use version control in the future?

Take home

- Overview about version control
- motivation to use it
- sources to look up things

references and further reading

references and resources

- BES Guide for Reproducible Code : https://www.britishecologicalsociety.org/wp-content/uploads/2019/06/BES-Guide-Reproducible-Code-2019.pdf
- paper Wilson 2017 "Good enough practices in scientific computing" : https://doi.org/10.1371/journal.pcbi.1005510
- Florian Schneider resources , e.g. : basic R intro : http://fdschneider.de/r_basics/
- books of Hadley Wickham
  - R for Data Science : https://r4ds.had.co.nz/
  - Advanced R : https://adv-r.hadley.nz/
  - tidyverse : https://www.tidyverse.org/
- Introduction Course to Data Science by Brian Enquist : https://www.researchgate.net/publication/310798573_How_to_think_About_Your_Data_Introduction_to_Data_Science_Management_what_they_don%27t_-_but_should_-_teach_you_about_the_scientific_method

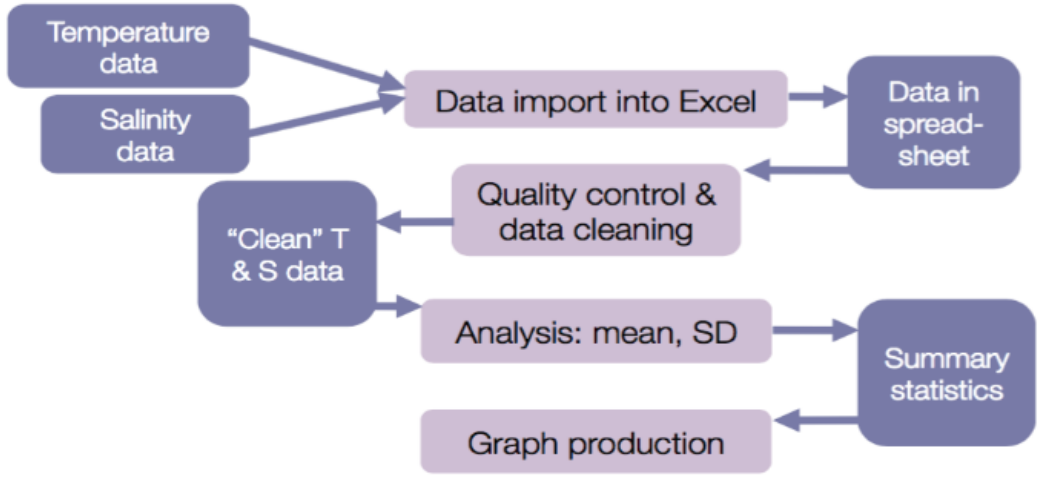
- additional material

Document your workflow

- "all science is a workflow, but we rarely document it and rarely can we reconstruct all parts of it" – *unknown person in the internet*

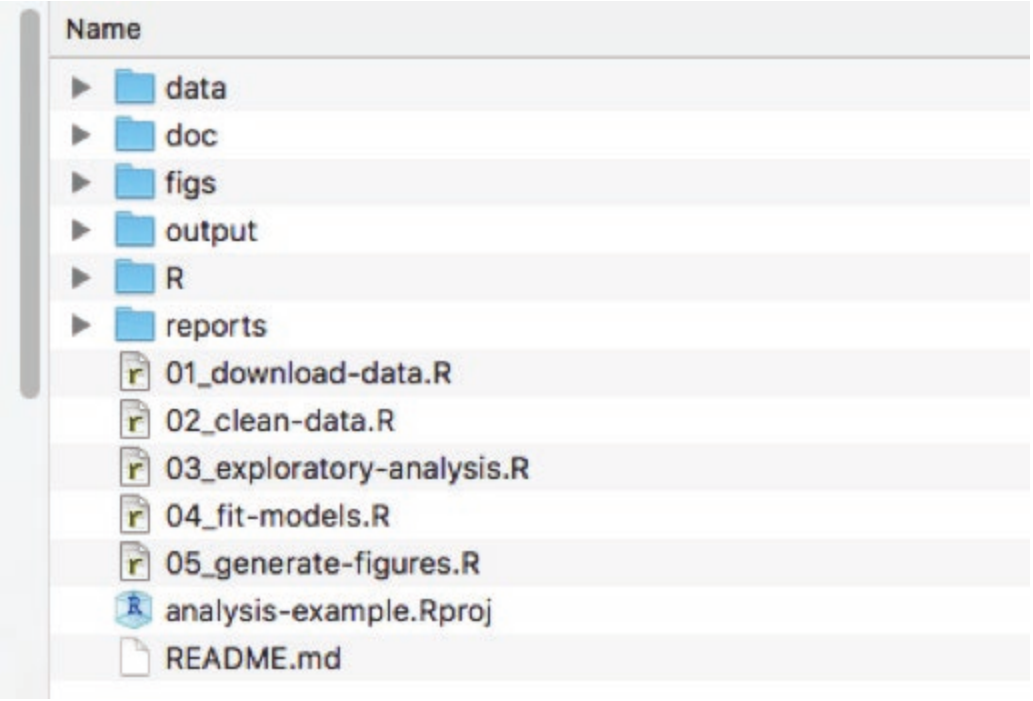
Workflow: how you get from the raw data to the final products of your research

Simple workflow: flow chart



folder structure

- each project to its own directory
- /data : raw data and metadata.
  - intermediate data : to separate subfolder in /data OR to /results
- /doc : manuscripts, documentations for code, lab notebook, ...
- /figs, /output or /results : statistical tables, figures and publication-ready figures (subdirectories)
- /R : R scripts
- README
- /old



BES reproducible code manual :
https://www.britishecologicalsociety.org/wp-content/uploads/2019/06/BES-Guide-Reproducible-Code-2019.pdf

use open file formats

- working reproducible → be able to open files in 10 years
- .csv and .txt (not .xslx, this is proprietary)
  - comma or semi-colon (;) separation, point decimal separator
- GIF, JPEG and PNG for images
- encoding : ASCII or UTF

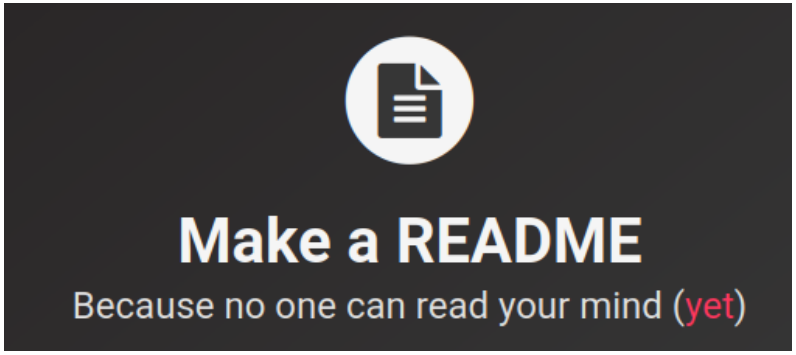
- Excel
  - can be used to open, edit and save .csv files
  - if your system language is german : uses semi-colon to separate columns and comma as decimal separator (4,3 instead of 4.3)
  - do not : merge cells, rely on formatting (e.g. coloured cells) or empty lines

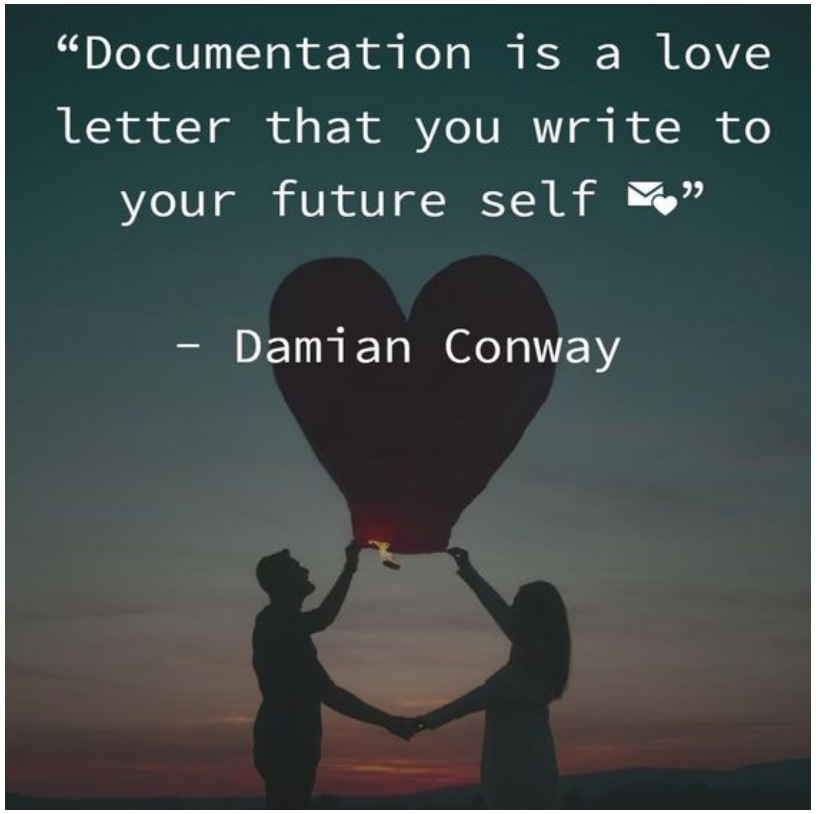
How many R scripts should I write?

- Many!
- one script for one purpose
    - data cleaning
    - analysis
    - making figure
    - ...
- → easier debugging, more flexible
- there are alternatives (debated)

README

- describe the project (shortly) and your name and date
- provide basic orientation in the project folder
  - describe all column names and variables, define units
  - describe measurements
  - link relevant papers or data
- describe file naming scheme
- include answers to questions that a person trying to reproduce your project might have





Awesome README files: https://github.com/matiassingers/awesome-readme

automated workflows

- what?
  - Code which can be run from input to output without you manually changing input or selecting subsections of the code.
- how ?
  - don’ t comment out parts of the script to adjust, but use if/else statements
  - use an executing script and a master script
  - if you need to do the same with several datasets : write code which is not dataset-specific, make a second script which runs the executing script for all datasets. Instead of having two scripts (D.R.Y.)


<DRY>
DRY
</DRY>
DON'T REPEAT YOURSELF

Don’t repeat yourself

- try to write code only once
- re-use good pieces of your work
- reduce the risk of errors : you always have to update code in *all* places
- watch out for copy-pasting
- use functions

functions

- should do what you mean
- no side effects
- describe parameters
- keep functions small
- few parameters
- return early
- for those who don’t know how to use functions yet : keep in mind to learn it

reduce mental overhead

- in german “mentaler Aufwand”
- Programming alone is already demanding, try to relieve your brain with a good overview.
- causes of mental overhead :
  - related code is not together
  - a lot of scrolling needed
  - little empty lines
  - no sections or visually distinguishable parts
  - very long lines (scrolling from left to right)
  - nested code (part of the code here, bt is dependent on manually running other part in different file)

```
# Load data ----------------------------


# Plot data ----------------------------
```

unit testing (advanced)

- entangled software **breaks upon tiny changes** (e.g. new R version, package updates, bug fixes, ...)
- write integrated test functions which check if your code works right
- unit testing in R :
- e.g. testthat

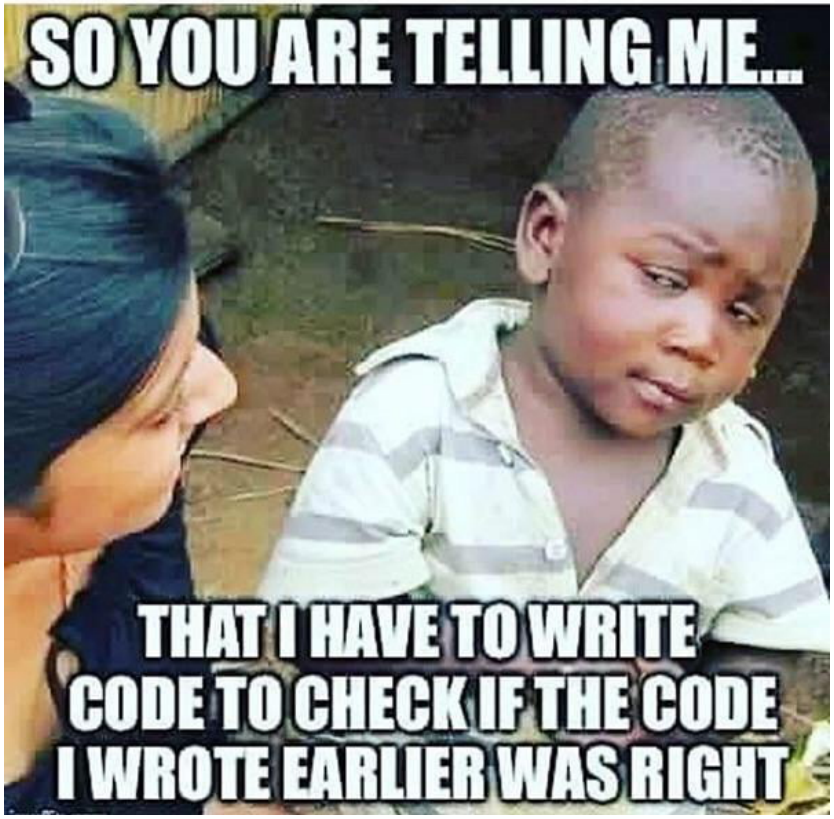
unit testing (advanced)

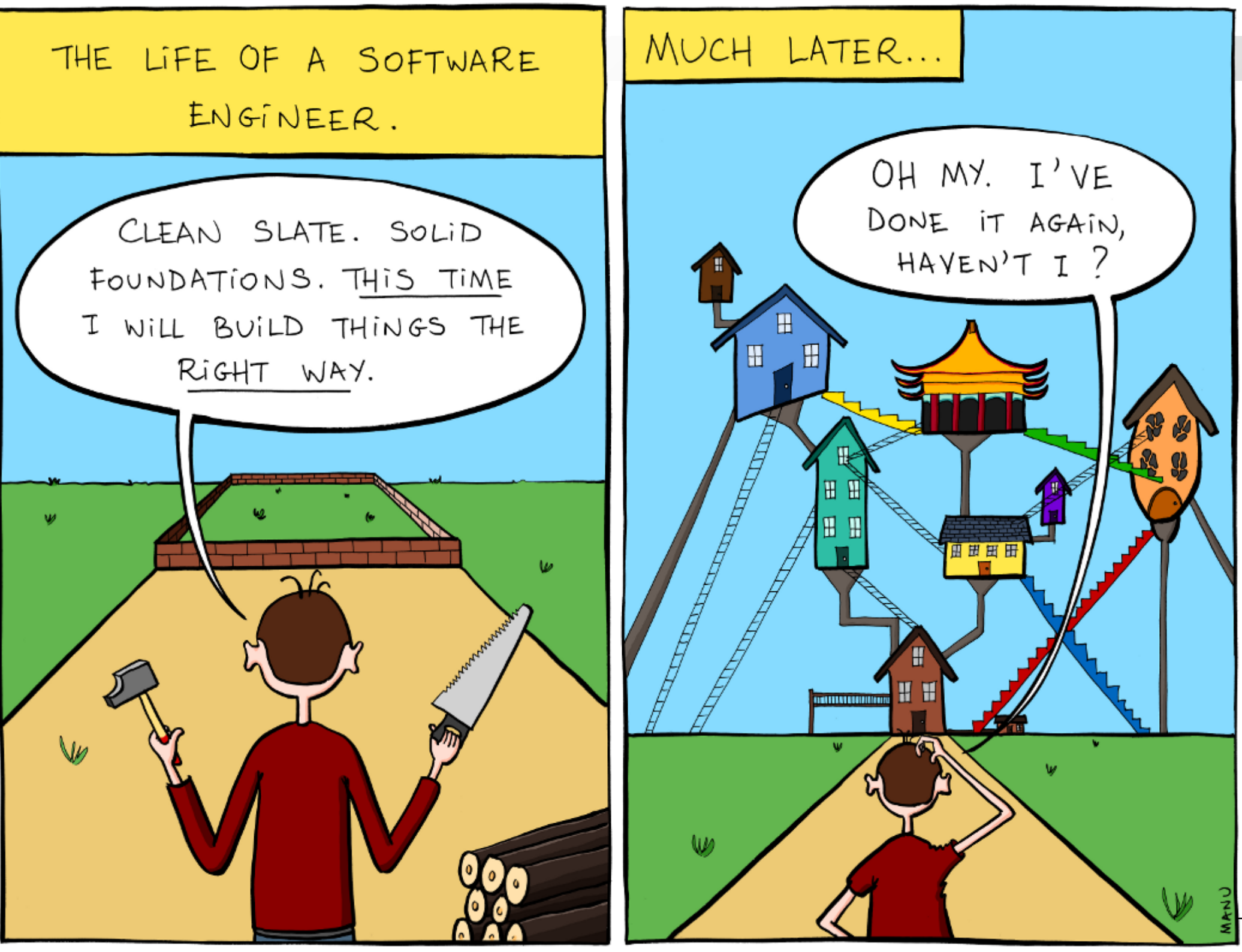
The Art of Software Testing, Third Edition

Editor(s): Glenford J. Myers, Tom Badgett, Corey Sandler

First published: 2 January 2012

Print ISBN: 9781118031964 | Online ISBN: 9781119202486 | DOI: 10.1002/9781119202486




THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?
MANU

code refractoring

- refractoring : going through your code and make it easier to understand without changing in- and output
  - difference to bug-fixing : there was no bug
- when should I refractor? all the time!
- things to address address :
  - cryptic names for variables: rename them
  - break long functions/ script bodies down to smaller units
  - replace algorithm if you learn a better one
  - combine duplicated pieces of code
  - add functions / loops after you learned them
  - TODO : make more advanced stuff slide, put this more basic
  - include new stuff, go over your code over again. once it works, its fine if it works, but if you want to improve you can do.