Infrastructure for Reproducible Research

Speaker: Doan Ho Anh Triet

triet.doan@gwdg.de

Institute of Computer Science, University of Göttingen
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

May 28, 2021

Outline

Hypothetical Scenarios

Submitted a Paper!

- Long delay
- Rerun code...

Found Cool Paper!

- No Code / No Data

Joined new lab!

- Code in a zip archive
- Critical Program
- Old student graduated...

Reproducibility Crisis

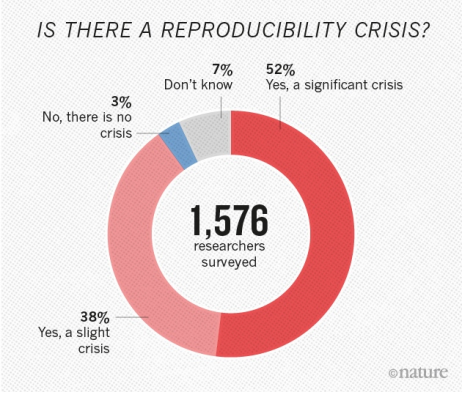


Figure 1: The crisis of Reproducibility. Nature 2016 survey [1].

Reproducibility Crisis

- 70 % Failed to reproduce other's experiments
- 50 % Failed to reproduce own experiments

Psychology [2]:

- 249 Dataset from American Psychology Association empirical articles
- 73% did not respond with their data over a 6-month period.

Cancer Research, 2012 study [3]

- 47 out of 53 medical research papers were irreproducible

Reproducibility Crisis - continued


BUSINESS INSIDER MARKETS

Shocking Paper Claims That Microsoft Excel Coding Error Is Behind The Reinhart-Rogoff Study On Debt

Mike Konczal, NewDeal2.0
Apr. 16, 2013, 12:40 PM 92,101

THE WALL STREET JOURNAL.

REAL TIME ECONOMICS

Reinhart, Rogoff Admit Excel Mistake, Rebut Other Critiques

Figure 2: Don't use Excel for analysis

Reinhart-Rogoff study on debt [4]

- Used to drive international austerity policies
- Incorrect result due to Excel Error
  - It's actually the opposite
- Found by a PhD student
  - Thomas Herndon

Applied Computer Science, 2014 study [5]

- 613 Papers
- 102 had code that could build and run

How did we get here?

Novelty

Researcher != Inventor

What is Replication?

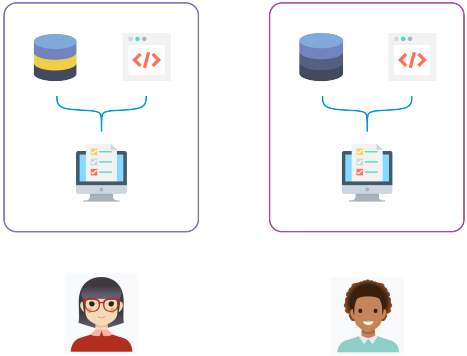


Figure 3: Replication. Different data, same results

- Ultimate standard for scientific evidence
- Ability to reproduce findings with different
  - investigators
  - data
  - analytical methods
  - laboratories
  - instruments

Replication Problems

Not all studies can be easily replicated:

- Unique Cases
  - Astronomic Observations
- Time Constrains
  - Studies that span decades
- Infrastructure
  - Big Data / HPC access
- Costs

Reproducibility Spectrum

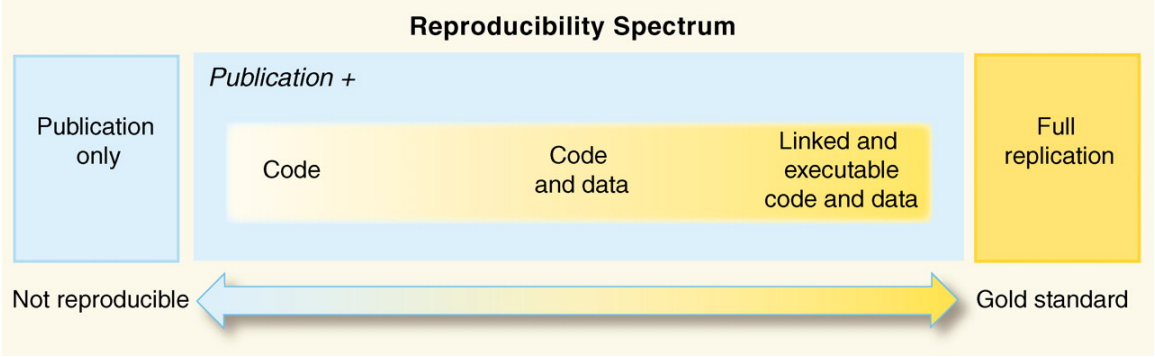


Figure 4: Reproducibility Spectrum [6]

What is Reproducibility?

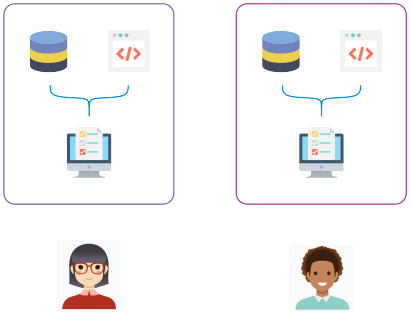


Figure 5: Reproduction. Same data, same code, same results.

- Bridges the gap between replication and stand-alone study
- Aims to validate finding using same:
  - data
  - methodology (code)

Benefits of Reproducible Research

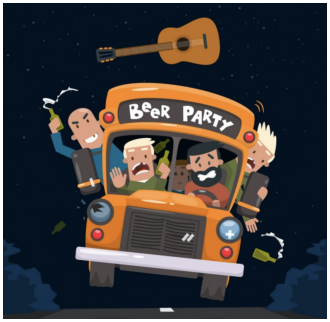


Figure 6: Reproducible research also increases the bus factor

Individual

- Reproduce your Research
- Onboard new researchers on the group
- Share research with others

Community

- Dissemination of results
- Increase public trust in science
- Assess the procedure of the analysis, not only the final outcome

Reproduction VS Replication

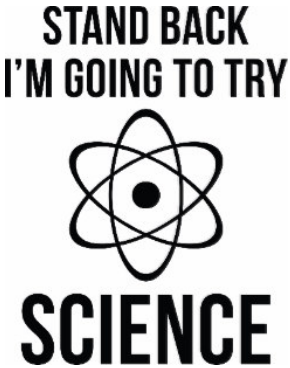
Replication:

- Focus:
  - Validity of Scientific Claim
- Asks
  - Is the claim true
- Reproduce Results
  - new investigators
  - new data / methods
- Ultimate Standard for strengthening scientific Evidence

Reproduction:

- Focus:
  - Validity of data analysis
- Asks
  - Can we trust this analysis
- Reproduce Results
  - new investigators
  - same data / methods
- A minimum standard for any scientific work

Reaching Reproducible Research



Focus on project:

- Organization
- Documentation
- Automation
- Dissemination

Project Plan

Figure 7: Start with a project plan

Questions:

- Final Product?
- Data Use?
- Maintenance?

Possible Scenarios:

- Thesis, article, report, figures
- Myself, colleagues, external
- 6 months, longer

Tools:

- LaTeX, Jupyter Notebooks, processing scripts
- Git, License?, Docs?
- Colleagues, OSS (Open Source Software) communities

File Organization

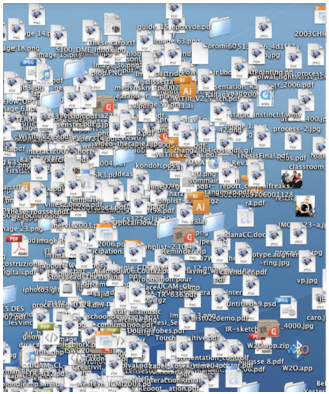


Figure 8: Does this look like your Desktop?

There will be:

- Lots of files
- Lots of changes
  - Files will change
  - The analysis will change
  - Results will change
  - Figures will change

File organization and naming can minimize the chaos

Regarding File Names

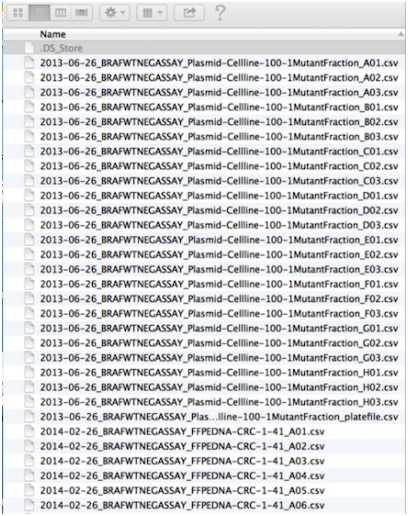


Figure 9: Well Defined file names

Three Principles of file naming:

1. Machine Readable
   - regex and globbing friendly
   - deliberate use of delimiters
2. Human Readable
   - contains info on content,
   - connects to concept of slug from semantic URLs
3. Plays well with default Ordering
   - put something numeric first with zero padding,
   - use ISO 8601 for dates YYYY-MM-DD

File name examples

Bad	Good
report.docx	2018-08-13_report-for-sla.tex
We use spaces!.doc	we_should_not.md
fig 1.jpg	fig01_scatter-len-vs-time.png
1_analysis.m	01.analysis.py

Folder Structure

```
.
├── AUTHORS.md
├── LICENSE
├── README.md
├── bin                <- Your compiled model code can be stored here (not tracked by git)
├── config             <- Configuration files, e.g., for doxygen or for your model if needed
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
├── docs               <- Documentation, e.g., doxygen or scientific papers (not tracked by git)
├── notebooks          <- Ipython or R notebooks
├── reports            <- For a manuscript source, e.g., LaTeX, Markdown, etc., or any project reports
│   └── figures        <- Figures for the manuscript or reports
└── src                <- Source code for this project
    ├── data           <- scripts and programs to process data
    ├── external       <- Any external source code, e.g., pull other git projects, or external libraries
    ├── models         <- Source code for your own model
    ├── tools          <- Any helper scripts go here
    └── visualization  <- Scripts for visualisation of your results, e.g., matplotlib, ggplot2 related.
```

Figure 10: A proposed directory structure.

Creating project structure

- To create such a folder structure, we can use the cookiecutter utility along with the Reproducible Science template.
- There is also a Data Science template, but it is a bit more complicated to start with.

Need for a License

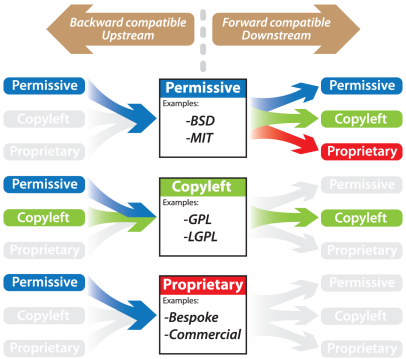


Figure 11: License Propagation, from the "A quick guide to software licensing for the scientist-programmer" [7]

Always Put a license

- Removes Ambiguity
- Simplifies reuse and legal matters
- Recommended to use permissive licenses
- Choose a License can help you decide.
  - The appendix contains detailed tables with the differences.

Version Control Systems (VCS)

Figure 12: Final.doc

- A VCS keeps track of changes in files over time
- Advantages
  - Enables collaboration
  - Keeps track of the changes made as well as appropriate timestamps
  - Can access you work everywhere!
  - Help you and others to have a tidy project structure
- Git is the most popular VSC nowdays
  - Distributed VCS
  - Made by Linux Torvalds (the Linux guy)

Version Control Systems (VCS)

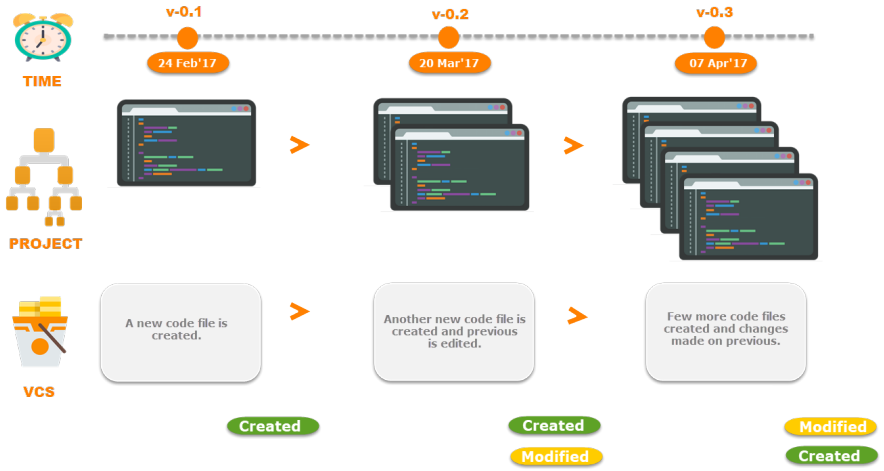


Figure 13: Workings of a VCS system. Every there are changes in a file, a new timestamp is generated.

Git Basics

- Next steps would familiarize you with basic commands
  - Configure Repo
  - Add files under git control
  - Show Changes to files
  - Commit Changes
- Pro Git is a good book to start with.

Git Basics

- Configure your name and email:

```
git config --global user.name "Triet Doan"
git config --global user.email "triet.doan@gwdg.de"
git config --list # Check your configuration
```

- Initialize a git repository at your project page

```
ls -a       # Show all directory files
git init    # Create git repo
ls -a       # See the new .git folder
```

Git Basics

- Check status of repository:

```
git status  # Check status of repository
```

- Add all untracked files

```
git add .   # Add all files
```

- Check status of again:
  - What has changed?

```
git status  # Check status of repository
```

Git Basics

- Commit the changes
  - Just because git is tracking the file, doesn't mean the changes are saved
  - The `-m` flag adds the commit message. This should be informative
  - Check status again after the commit and see what changed.

```
git commit -m "Initial Commit"
```

- See the commit history of the project

```
git log
```

Git Basics

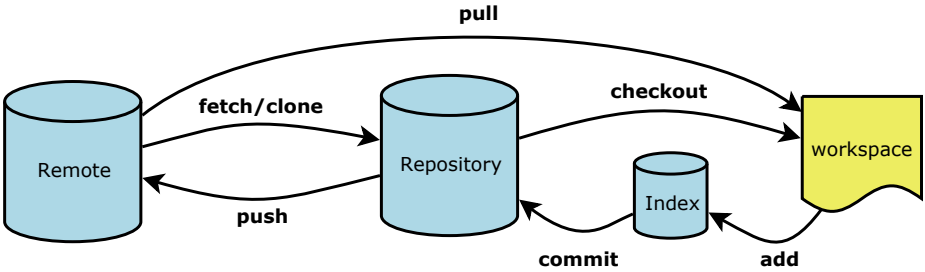


Figure 14: Git workflow

RAW Data

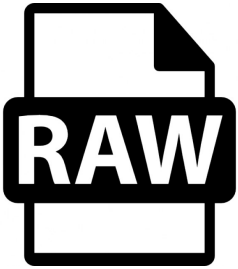


Figure 15: RAW data are sacred

- Raw data are stored under `data/raw`
- You shall not modify the raw data
- Not modified, not under version control
- Add a `README.md` with metadata information
  - Data Source
  - Field Types
  - URL
  - Collected Date

Basic Principles of Automation

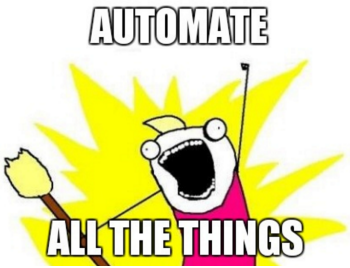


Figure 16: Call to automation!

- Core Principle of Automation:
  - DON'T REPEAT YOURSELF!
- Automate
  - Environment Creation
  - Analysis
  - Artifact Creation
- Simply put, don't do changes by hand...
  - and don't use Spreadsheets
  - European Spreadsheet Risks Interest Group

Makefile

GNU Make

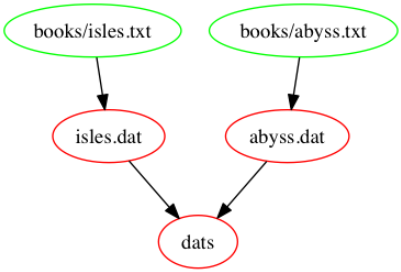
- A build automation tool
- Deals with targets and dependencies to create a DAG (Directed Acyclic Graph).
- Filename: `Makefile`
- Basic format:

```
target-1: requirements
<TAB>   commands that will create target-1

target-2: requirements target-1
<TAB>   commands that will create target-2
```

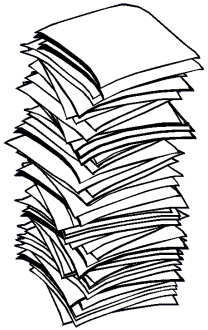
- A good and easy course on Automation and Make by Software Carpentry.

An examlple Makefile

```
# Count words.
.PHONY : dats
dats : isles.dat abyss.dat

isles.dat : books/isles.txt
    python countwords.py books/isles.txt isles.dat

abyss.dat : books/abyss.txt
    python countwords.py books/abyss.txt abyss.dat

.PHONY : clean
clean :
    rm -f *.dat
```



More information on Make



Many Types of Document

- Manuals
- Notes on the proceedure
- Publication Paper
- Presentation Slides
- ...

Many possible formats

- Plain Text is the most "Reproducible"

Markdown → HTML

```
Title (header 1, actually)
==========================

This is a Markdown document.

## Medium header (header 2, actually)

It's easy to do *italics* or __make things bold__.

> All models are wrong, but some are useful. An
approximate answer to the right problem is worth a
good deal more than an exact answer to an
approximate problem. Absolute certainty is a
privilege of uneducated minds--and fanatics. It is,
for scientific folk, an unattainable ideal. What
you do every day matters more than what you do
once in a while. We cannot expect anyone to know
anything we didn't teach them ourselves.
Enthusiasm is a form of social courage.

Code block below. Just affects formatting here but
we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ...
the average is computed as $\frac{1}{n} \sum_{i=1}
^{n} x_{i}$. Or display equations like this:

$$
\begin{equation*}
|x| =
\begin{cases} x & \text{if $x \geq 0$,} \\\\
-x & \text{if $x \le 0$.}
\end{cases}
\end{equation*}
$$
```

Title (header 1, actually)

This is a Markdown document.

Medium header (header 2, actually)

It's easy to do *italics* or **make things bold**.

> All models are wrong, but some are useful. An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem. Absolute certainty is a privilege of uneducated minds-and fanatics. It is, for scientific folk, an unattainable ideal. What you do every day matters more than what you do once in a while. We cannot expect anyone to know anything we didn't teach them ourselves. Enthusiasm is a form of social courage.

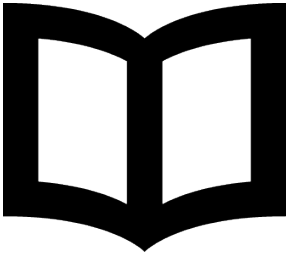
Code block below. Just affects formatting here but we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ... the average is computed as $\frac{1}{n} \sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

README files and Pandoc



Include README files

- Information on Project Name
- Dates
- Maintainer's contanct info
- Data Origin
- Goal of the project

Pandoc

- Universal Markup Converter
- Supports Word Formats
- Ebooks
- TEX and pdf

Provenance

Provenance

- The chronology of the ownership, custody or location of a historical object
- Simply put:
  - Know which code modified / created what.

A number of tools exist, for Python, one can use recipy

Publication, Sharing, Archiving

- Not the same!
- Shared
  - Found online
  - Personal Transfer
- Published
  - Citable
- Archived
  - Long Term Preservation

Why share?

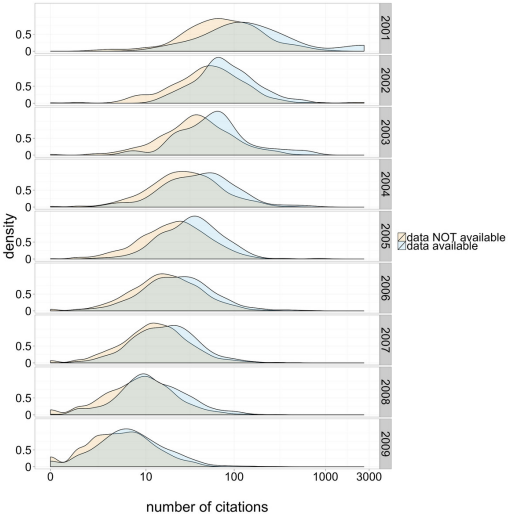
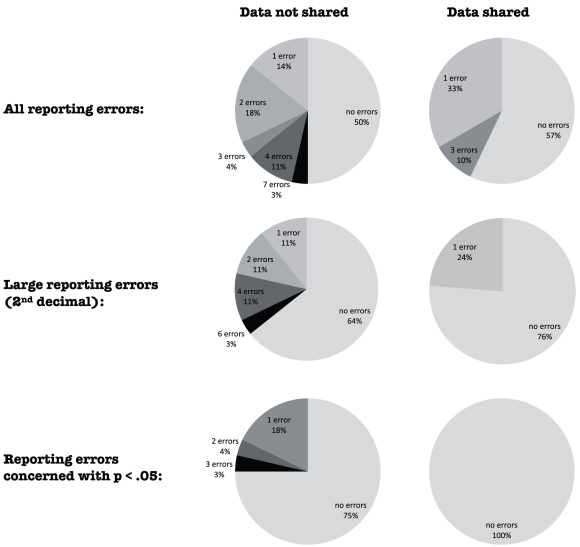


Figure 17: Sharing increases citations [8]

- Increased Visibility
- Increased Citations

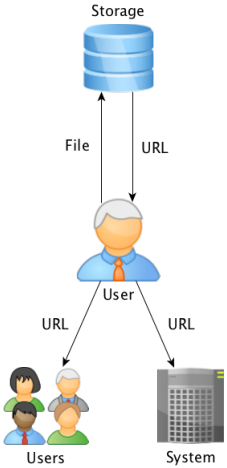
Why share?

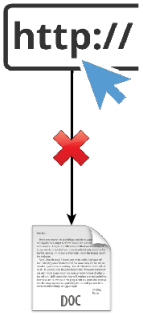


- Better Research
- Willingness to Share Research Data Is Related to the Strength of the Evidence and the Quality of Reporting of Statistical Results [9]

Figure 18: And strengthens science [9]

Problems with URL





FAIR principles

Figure 19: FAIR principles

Persistent identifiers

- Digital Object Identifier (DOI)
  - https://doi.org/10.1109/5.771073
- Handle
  - http://hdl.handle.net/20.1000/113

For students of Göttingen University

- Göttingen Research Online (GRO) for publishing data
- eResearch Alliance service catalog

Conclusion

Aim for reproducibility from the start

- Reproducibility can be achieved on many levels
- Can be abstracted in 4 “levels”
  - Documentation
  - Organization
  - Automation
  - Dissemination
- Requires more “initial investment”
- Worth it in the long run


thank you
danke
謝謝
рахмат
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
sukriya
kop khun krap
terima kasih
tapadh leat
mochchakkeram
спасибо
Баярлалаа
dank je
grazie
arigatô
dakujem
мерси
감사합니다
ありがとう
ευχαριστώ
faafetai lava
kiitos
dankie
dhanyavad
hvala
mauruuru
köszönöm
blagodaram
vinaka
спасибі
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
djiere dieuf
дякую
mamnun
murakoze
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
takk
trugarez
shukriya
merce
dhanyavadagalu
diolch
xièxie
tanemirt
rahmet
najis tuke
kam sah hamnida
rahmat
didi madloba
sagolun
mèsi
sobodi
děkuji
enkosi
bayarlalaa
gracje
nanni
nandri
mersi
kia ora
barka
welalin
tack
spas
dankon
akun
tau

General

- Software Carpentry Lessons: Lessons on Unix Shell, Git, Python, Make, and other things.
- Data Carpentry: Similar to the above, but focusing more on data related lessons for various disciplines.
- DataTau: A news aggregator, focusing on data science news.
- Awesome Data Science: Part of the "Curated list of awesome lists", it contains a number of useful link, for anything data science related.
- Overleaf Documentation: For those who want to start learning LaTeX
- Data Science at the Command Line: A free book covering everything on the topic. Even if you don't end up doing all that, it's really helpful to know what you can do.
- HackerRank: A way to learn programming with a number of languages available, through small exercises.

Reproducible Research + Open Science

- Reproducible Research: Walking the Walk: A SciPy 2014 workshop on reproducible research. Parts of the presentation were based on this
- Down the rabbit hole. A 101 on reproducible workflows with Python: A PyCon 2018 talk, on which I based some parts of the theoretical intro and the code examples. Etherpad, Slides
- Opening Science: The Evolving Guide on How the Web is Changing Research, Collaboration and Scholarly Publishing.
- Open Science MOOC: An Open Science Course. Still in progress, but worth to wait for.
- FOSTER: Contains a number of free courses on Open Science.
- OpenCon: The open science conference. At the resources you can find videos and material of previous conferences.
- Publons: A way to get recognition for doing peer review.
- Orcid: A unique identifier for your scientific career.

Some Blogs / sites worth following

- Study Hacks: Cal Newport's blog. A lot of material related to work and study.
- Five Thirty Eight: A website doing journalism with data science. They also post all their datasets online.
- The Pudding: The Pudding explains ideas debated in culture with visual essays.
- Green Tea and Velociraptors: Jon posts a number of things regarding open science
- The Thesis Wisperer: Articles regarding PhD life and research
- The Research Whisperer: Just like the above but with more money related stuff.
- Overcoming Bias: On why believe and do what we do, why pretend otherwise, and how to do better.
- You are not so smart: A celebration of self delusion
- The Morning Paper: An overview of one computer science paper a day.
- O'Reilly Ideas: The 4-short links daily posts contain a lot of great resources

For ~~procrastination~~ Creative Thinking

- Phd Comics
- XKCD
- Abstruse Goose
- Saturday Morning Breakfast Cereal
- The Oatmeal
- Zen Pencils

Bibliography I

M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature News*, vol. 533, no. 7604, p. 452, 2016.

J. M. Wicherts, D. Borsboom, J. Kats, and D. Molenaar, "The poor availability of psychological research data for reanalysis." *American Psychologist*, vol. 61, no. 7, p. 726, 2006.

C. G. Begley and L. M. Ellis, "Drug development: Raise standards for preclinical cancer research," *Nature*, vol. 483, no. 7391, p. 531, 2012.

C. M. Reinhart and K. S. Rogoff, "Growth in a time of debt," *American Economic Review*, vol. 100, no. 2, pp. 573–78, 2010.

C. Collberg, T. Proebsting, G. Moraila, A. Shankaran, Z. Shi, and A. M. Warren, "Measuring reproducibility in computer systems research," *Department of Computer Science, University of Arizona, Tech. Rep*, vol. 37, 2014.

Bibliography II

- R. D. Peng, "Reproducible research in computational science," *Science*, vol. 334, no. 6060, pp. 1226–1227, 2011.
- A. Morin, J. Urban, and P. Sliz, "A quick guide to software licensing for the scientist-programmer," *PLoS computational biology*, vol. 8, no. 7, p. e1002598, 2012.
- H. A. Piwowar and T. J. Vision, "Data reuse and the open data citation advantage," *PeerJ*, vol. 1, p. e175, 2013.
- J. M. Wicherts, M. Bakker, and D. Molenaar, "Willingness to share research data is related to the strength of the evidence and the quality of reporting of statistical results," *PloS one*, vol. 6, no. 11, p. e26828, 2011.