Improve your workflow for reproducible science

Mine Çetinkaya-Rundel
Duke University + RStudio

🔗 **bit.ly/improve-repro-workflow**

@minebocek
mine-cetinkaya-rundel
cetinkaya.mine@gmail.com


The results in Table 1 don’t seem to correspond to those in Figure 2!

61 45 4

94 12 3

```r
# set.seed
set.seed(20190314)

# generate 8 random numbers between 0 and 99
runif(8, 0, 99) %>% round()
```

20 44

more than **70** percent

have tried and **failed** to reproduce
another scientist's experiments

Monya Baker,. "1,500 scientists lift the lid on reproducibility." Nature News 533.7604 (2016): 452.

more than **50** percent

have tried and **failed** to reproduce
their **own** experiments

Monya Baker. "1,500 scientists lift the lid on reproducibility." Nature News 533.7604 (2016): 452.

Google Scholar yields

1070

results containing the term **reproducibility crisis** just in **2021**

Google Scholar Search, 28 October 2021.

1992

earliest reference **reproducibility research***

Jon Claerbout and Martin Karrenbach. "Electronic documents give reproducible research a new meaning."
SEG Technical Program Expanded Abstracts 1992. Society of Exploration Geophysicists, 1992. 601-604.

that I could find…

Electronic Documents Give Reproducible Research a New Meaning **RE1.3**

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides a concrete definition of reproducibility in computationally oriented research. Experience at the Stanford Exploration Project shows that preparing such electronic documents is little effort beyond our customary report writing; mainly, we need to file everything in a systematic way.

In 1990 we began experimenting with electronic documents that merge our scientific software with our word-processing software. A year later we manufactured a CD-ROM containing a new textbook, Joe Dellinger's doctoral dissertation, and two progress reports of the Stanford Exploration Project. We distributed these CD-ROMs[1] to sponsors and many friends at the 1991 SEG meeting.

In 1990, we set this sequence of goals:

- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).
- Export electronic documents to numerous other sites (sponsors) so they can readily reproduce a substantial portion of our Stanford research.

We met all these goals and set new ones:

- produce all new documents in this form, including lab reports in formal classes and "lab notebooks" of research progress.
- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's interior, but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commerical television. We include all our plot software as well as freely available software from many sources, including compilers and the LaTeX word processing system. Naturally we cannot include licensed software, but with the exception of Fortran and C compilers and the UNIX system itself, our publication includes source code for everything needed. The CD-ROM, at 680 megabytes, is so large we have had room for many executable programs on popular brands of workstations. The presence of these executables gives our readers a fast start.

Nearly everyone would rather read a paper book than the bitmapped page images on a screen that you see with an electronic document. But the illustrations in the electronic book are mostly in color, many are movies, and some are interactive. So the electronic book gives the reader a better understanding of the results. We typically use an interactive movie program to compare seismic sections where successive frames include processing with various parameters. The movie medium is much more informative than comparing seismic sections side by side. 3-D volumes are much better exhibited by movies than static paper illustrations. We are delivering a volume of software that is accessed like a book.

[1]SEP-CD-1 is available from Stanford University Press, $15 plus shipping, tel 415-723-1593

In 1990, we set this sequence of goals:

- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).
- Export electronic documents to numerous other sites (sponsors) so they can readily reproduce a substantial portion of our Stanford research.

Jon Claerbout and Martin Karrenbach. "Electronic documents give reproducible research a new meaning." SEG Technical Program Expanded Abstracts 1992. Society of Exploration Geophysicists, 1992. 601-604.

Electronic Documents Give Reproducible Research a New Meaning

RE1.3

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides a concrete definition of reproducibility in computationally oriented research. Experience at the Stanford Exploration Project shows that preparing such electronic documents is little effort beyond our customary report writing; mainly, we need to file everything in a systematic way.

In 1990 we began experimenting with electronic documents that merge our scientific software with our word-processing software. A year later we manufactured a CD-ROM containing a new textbook, Joe Dellinger's doctoral dissertation, and two progress reports of the Stanford Exploration Project. We distributed these CD-ROMs[1] to sponsors and many friends at the 1991 SEG meeting.

In 1990, we set this sequence of goals:

- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).
- Export electronic documents to numerous other sites (sponsors) so they can readily reproduce a substantial portion of our Stanford research.

We met all these goals and set new ones:

- produce all new documents in this form, including lab reports in formal classes and "lab notebooks" of research progress.
- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

[1]SEP-CD-1 is available from Stanford University Press, $15 plus shipping, tel 415-723-1593

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's interior, but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commerical television. We include all our plot software as well as freely available software from many sources, including compilers and the LaTeX word processing system. Naturally we cannot include licensed software, but with the exception of Fortran and C compilers and the UNIX system itself, our publication includes source code for everything needed. The CD-ROM, at 680 megabytes, is so large we have had room for many executable programs on popular brands of workstations. The presence of these executables gives our readers a fast start.

Nearly everyone would rather read a paper book than the bitmapped page images on a screen that you see with an electronic document. But the illustrations in the electronic book are mostly in color, many are movies, and some are interactive. So the electronic book gives the reader a better understanding of the results. We typically use an interactive movie program to compare seismic sections where successive frames include processing with various parameters. The movie medium is much more informative than comparing seismic sections side by side. 3-D volumes are much better exhibited by movies than static paper illustrations. We are delivering a volume of software that is accessed like a book.

Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's inte-

Jon Claerbout and Martin Karrenbach. "Electronic documents give reproducible research a new meaning." SEG Technical Program Expanded Abstracts 1992. Society of Exploration Geophysicists, 1992. 601-604.

NEWS · 28 FEBRUARY 2019 ·

Pioneering 'live-code' article allows scientists to play with each other's results

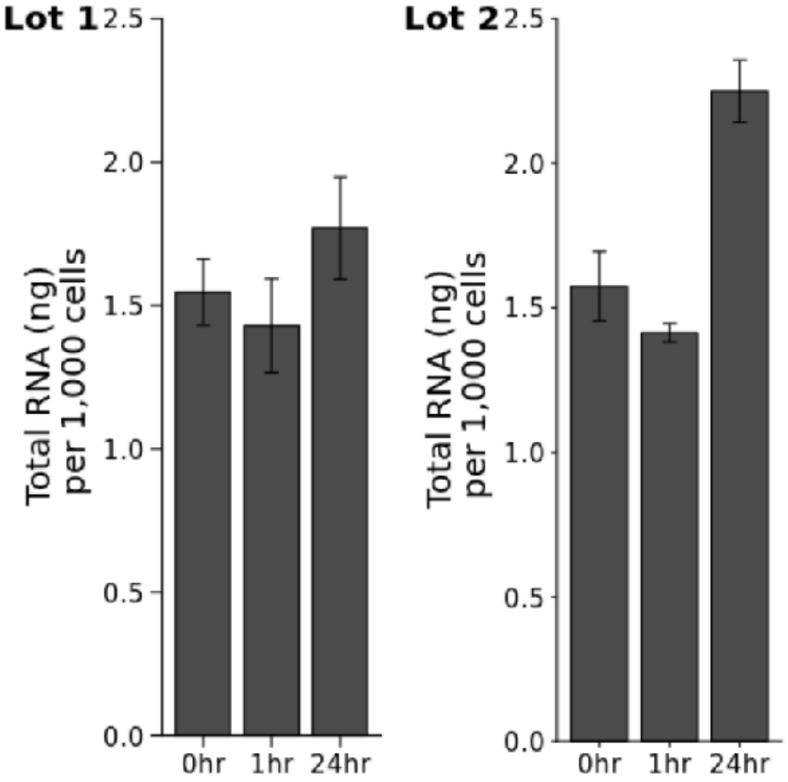
eLife's prototype lets scientists modify the software underlying figures to validate, build on, or better understand the work.

This is a Reproducible document. See the original article or source.

NOTE: Below is a reproducible version of Figure 1B. You can inspect the code, make changes and run the code by pressing SHIFT+ENTER. The data used can be downloaded here.

R Script

status: ready (run code with ⇧↵)



Total RNA levels following c-Myc overexpression

Michelle Lewis, et al. "Replication Study: Transcriptional amplification in tumor cells with elevated c-Myc." Elife 7 (2018): e30274.

setting the stage

Photo by Alexander Dummer on Unsplash].

replicability

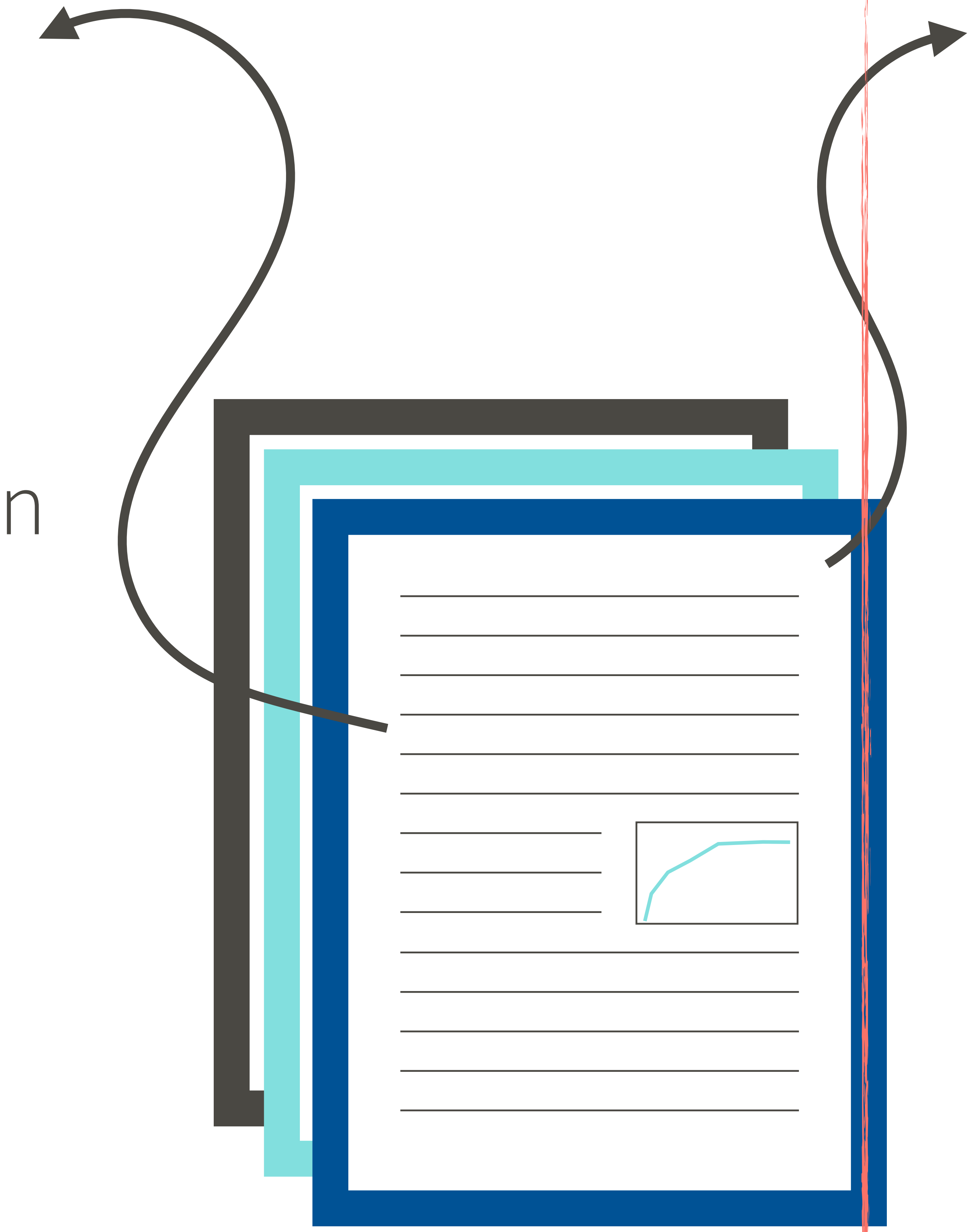
same research question

same results

new data


EDITORIAL

The ASA's Statement on p-Values: Context, Process, and Purpose



reproducibility

same research question

same results

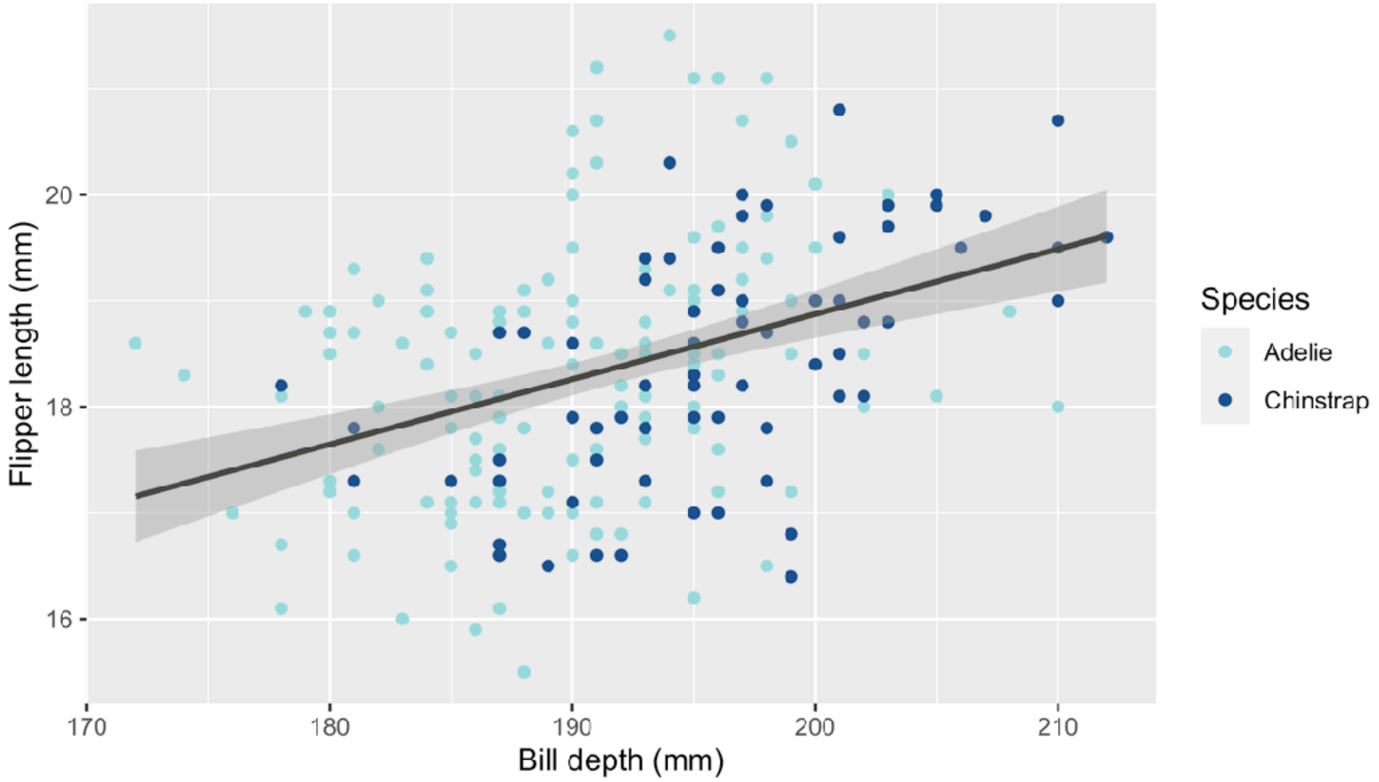
same data

e.g.

Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	-0.0820	0.00618	-13.3	1.23e-32

Figure 2. Relationship between bill depth and flipper length.

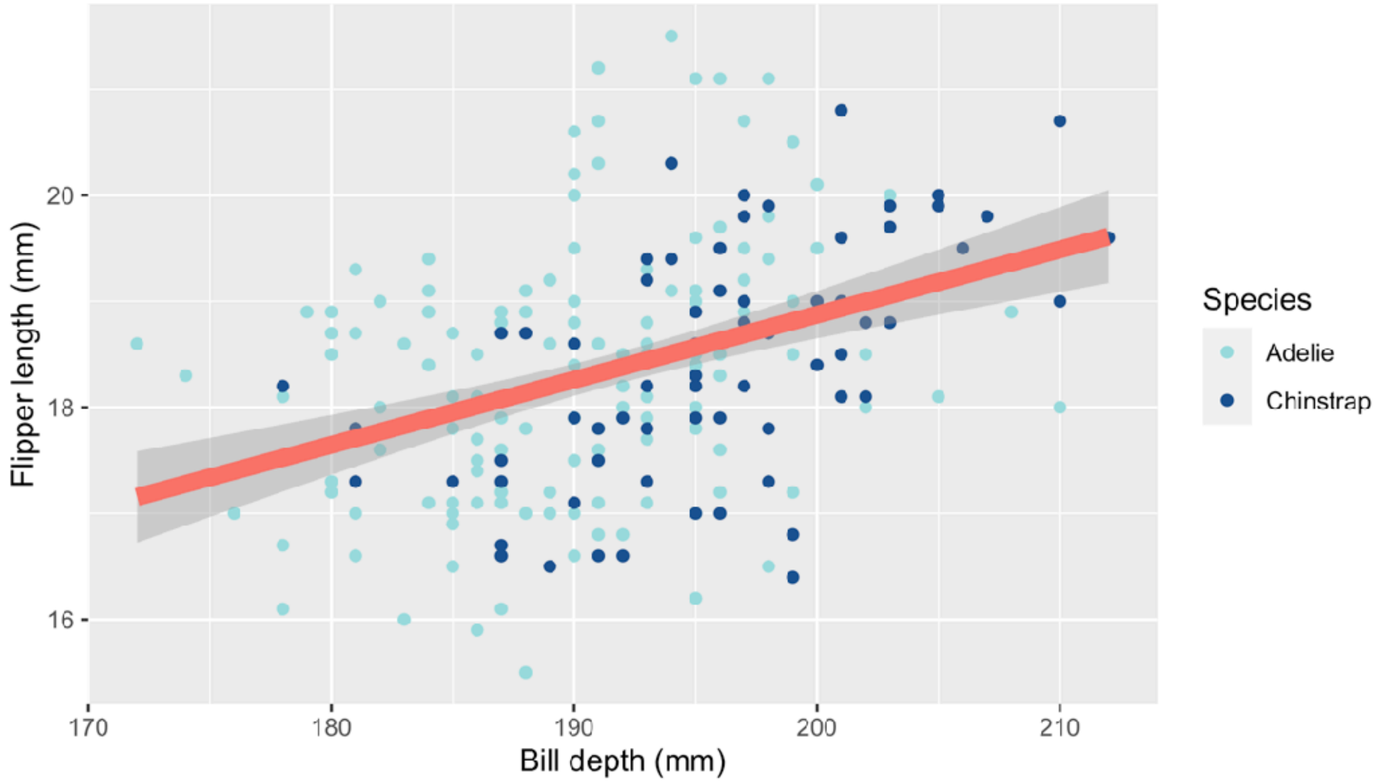


e.g.

Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	**-0.0820**	0.00618	-13.3	1.23e-32

Figure 2. Relationship between bill depth and flipper length.



analysis

```r
# fit model
model <- lm(bill_depth_mm ~ flipper_length_mm, data = penguins)

# print model summary
tidy(model)
```

report

Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	-0.0820	0.00618	-13.3	1.23e-32

analysis

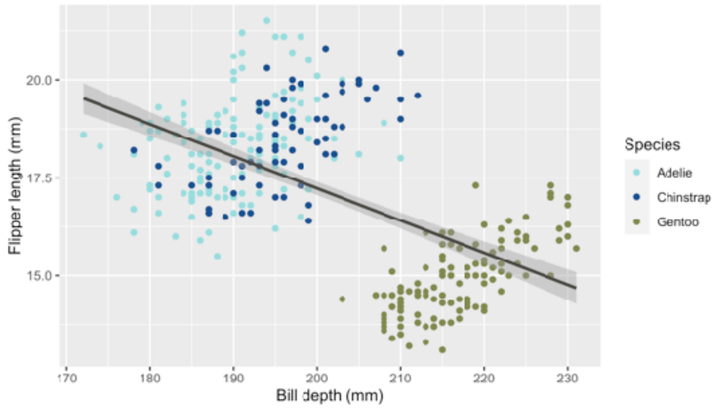
```
# visualize the relationship
ggplot(penguins) +
  geom_point(
    aes(x = bill_depth_mm, y = flipper_length_mm, color = species)
  ) +
  geom_smooth(
    aes(x = bill_depth_mm, y = flipper_length_mm),
    method = "lm"
  )
```

report

Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	-0.0820	0.00618	-13.3	1.23e-32

Figure 2. Relationship between bill depth and flipper length.



analysis

```r
# filter out Gentoos
penguins_nongentoo <- penguins %>%
  filter(species != "Gentoo")

# visualize the relationship
ggplot(penguins_nongentoo) +
  geom_point(
    aes(x = bill_depth_mm, y = flipper_length_mm, color = species)
  ) +
  geom_smooth(
    aes(x = bill_depth_mm, y = flipper_length_mm),
    method = "lm"
  )
```

report

Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	-0.0820	0.00618	-13.3	1.23e-32

Figure 2. Relationship between bill depth and flipper length.

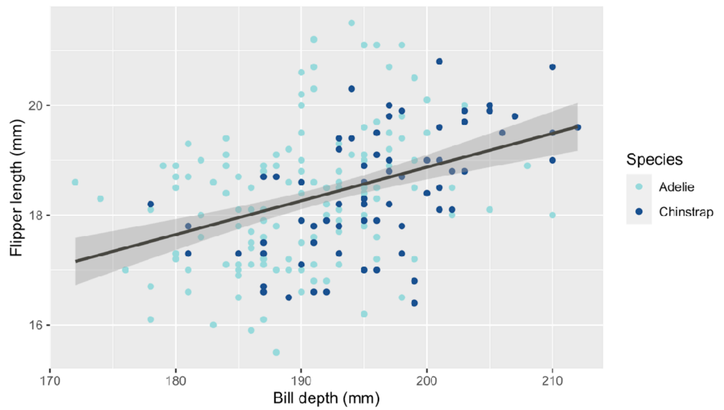
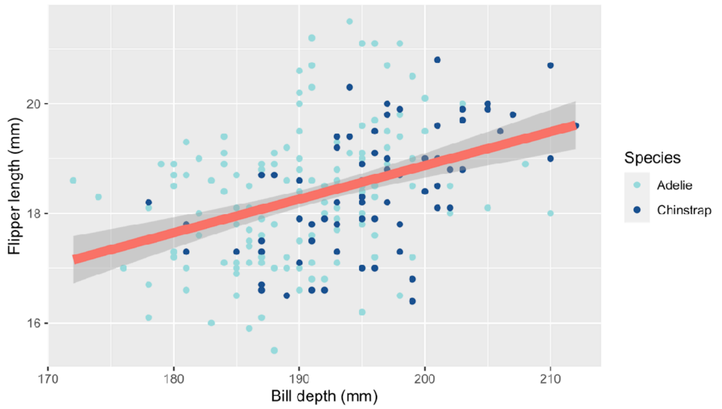
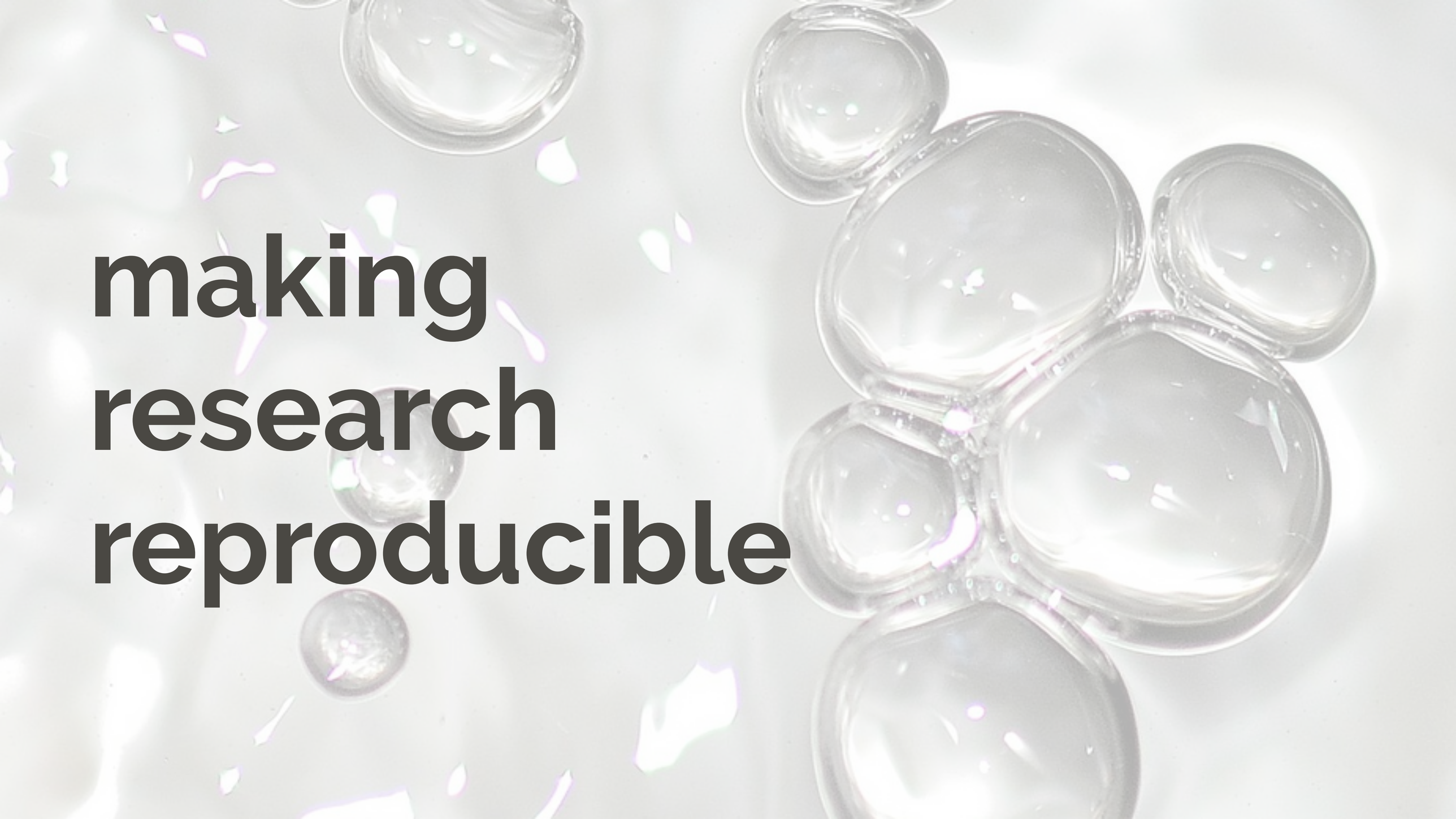


Table 1. Regression output for predicting bill depth from flipper length.

term	estimate	std.error	statistic	p.value
(Intercept)	33.6	1.25	27.0	1.39e-86
flipper_length_mm	-0.0820	0.00618	-13.3	1.23e-32

Figure 2. Relationship between bill depth and flipper length.




making
research
reproducible

make

raw data

code & documentation to reproduce the analysis

specifications of your computational environment

available and accessible

Peng, Roger. "The reproducibility crisis in science: A statistical counterattack." Significance 12.3 (2015): 30-32.
Gentleman, Robert, and Duncan Temple Lang. "Statistical analyses and reproducible research." Journal of Computational and Graphical Statistics 16.1 (2007): 1-23.

“The most important tool is the **mindset**, when starting, that the end product will be reproducible.”

– Keith Baggerly

nobody,
not even yourself,
can recreate any part
of your analysis

push button
reproducibility
in published work

“There’s **no one-size-fits-all solution** for computational reproducibility.”

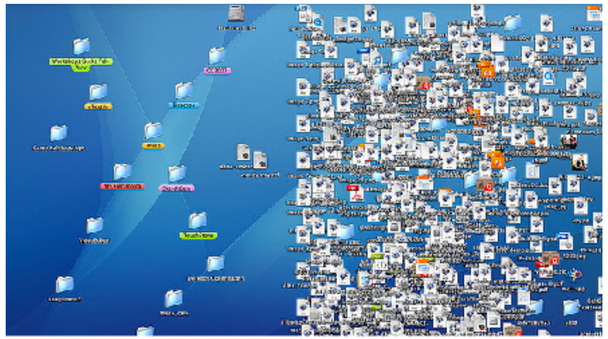
Perkel, Jeffrey M. "A toolkit for data transparency takes shape." Nature 560 (2018): 513-515.

8 principles

but the following might help…

organize your project

level of organization



simpler analysis

- raw-data
- processed-data
- manuscript
  - manuscript.Rmd

more complex analysis

- raw-data
- processed-data
- scripts
- figures
- manuscript
  - manuscript.Rmd

stick with the conventions of your peers

write READMEs liberally

raw-data

├─ README.md

├─ airlines.csv

├─ airports.csv

├─ flights.csv

├─ planes.csv

├─ weather.csv

processed-data

scripts

figures

manuscript

README

This folder contains the raw data
for the project.
All datasets were downloaded from
openflights.org/data.html
on 2019-04-01.

- airlines: Airline names
- airports: Airports metadata
- flights: Flight data
- planes: Plane metadata
- weather: Hourly weather data

keep data tidy & machine readable

Student	Exam Grade		
Name	1	2	Major
Barney Donaldson	89	76	Data Science, Public Policy
Clay Whelan	67	83	Public Policy
Simran Bass	82	90	Statistics
Chante Munro	45	72	Political Science, Statistics
Gabrielle Cherry	32	79	.
Kush Piper	98	sick	Statistics
Faizan Ratliff	82	75	Data Science
Torin Ruiz	70	80	Sociology, Statistics
Reiss Richardson	missed exam	34	Neuroscience
Ajwa Cochran	50	65	Data Science

record code + document non-code steps + write tests

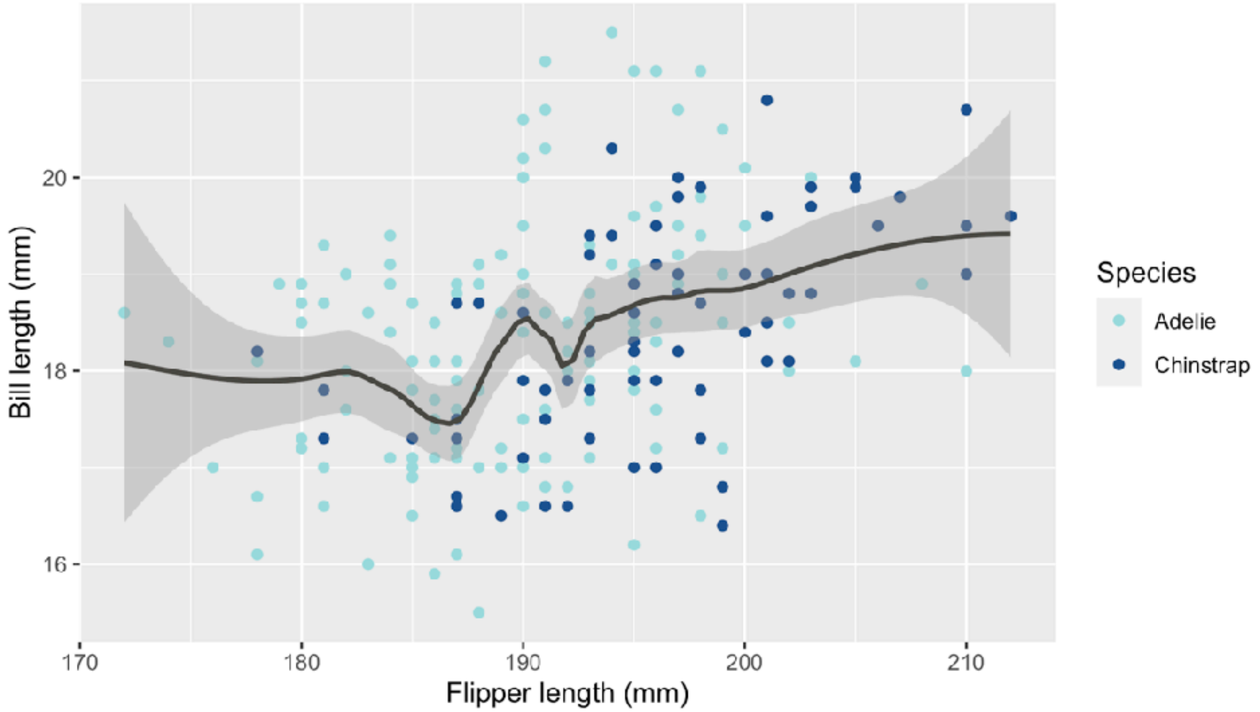
name	exam_1	exam_2	first_major	second_major	participation
Barney Donaldson	89	76	Data Science	Public Policy	ok
Clay Whelan	67	83	Public Policy	NA	ok
Simran Bass	82	90	Statistics	NA	ok
Chante Munro	45	72	Political Science	Statistics	low
Gabrielle Cherry	32	79	NA	NA	ok
Kush Piper	98	NA	Statistics	NA	ok
Faizan Ratliff	82	75	Data Science	NA	ok
Torin Ruiz	70	80	Sociology	Statistics	ok
Reiss Richardson	NA	34	Neuroscience	NA	low
Ajwa Cochran	50	65	Data Science	NA	low

Low participation

Broman, Karl W., and Kara H. Woo. "Data organization in spreadsheets." The American Statistician 72.1 (2018): 2-10.

comment your code

```r
# use loess smoothing
ggplot(penguins_nongentoo) +
  geom_point(
    aes(x = flipper_length_mm, y = bill_depth_mm, color = species)
  ) +
  geom_smooth(
    aes(x = flipper_length_mm, y = bill_depth_mm),
    method = "loess", span = 0.375
  )
```



use literate programming

tab1-fig2.Rmd

Knit | Run

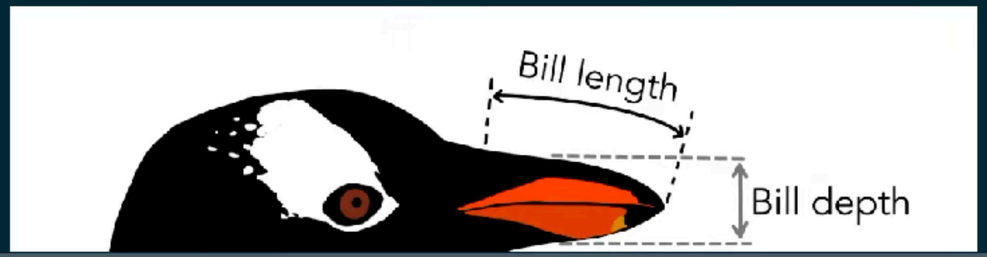
Normal | Format | Insert | Table

```yaml
---
title: "Table 1 matches Figure 2!"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
  bookdown::html_document2:
    fig_caption: yes
bibliography: references.bib
---
```

```{r chunk-options, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, message = FALSE)
```

```{r setup}
library(broom)          # for tidy model output
library(glue)           # for gluing strings to data
library(knitr)          # for kable
library(palmerpenguins) # for data
library(tidyverse)      # for data wrangling and visualisation
```

In this report we evaluate the relationship between relationship between bill depth and flipper length of penguins. The data come from @gorman2014.



Exploratory data analysis | R Markdown

improve-repro-workflow-reproducibilitea-2020 - RStudio

tab1-fig2.Rmd

```
---
title: "Table 1 matches Figure 2!"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
  bookdown::html_document2:
    fig_caption: yes
bibliography: references.bib
---
```

```
{r chunk-options, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, message = FALSE)
```

```
{r setup}
library(broom)          # for tidy model output
library(glue)           # for gluing strings to data
library(knitr)          # for kable
library(palmerpenguins) # for data
library(tidyverse)      # for data wrangling and visualisation
```

In this report we evaluate the relationship between relationship between bill depth and flipper length of penguins. The data come from @gorman2014.

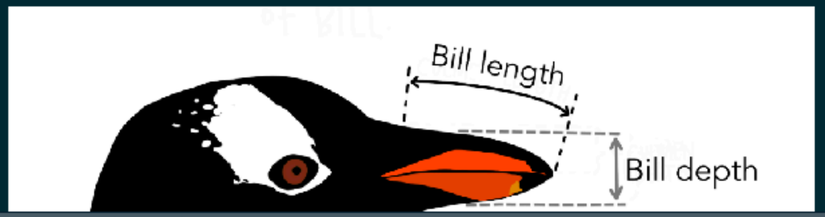
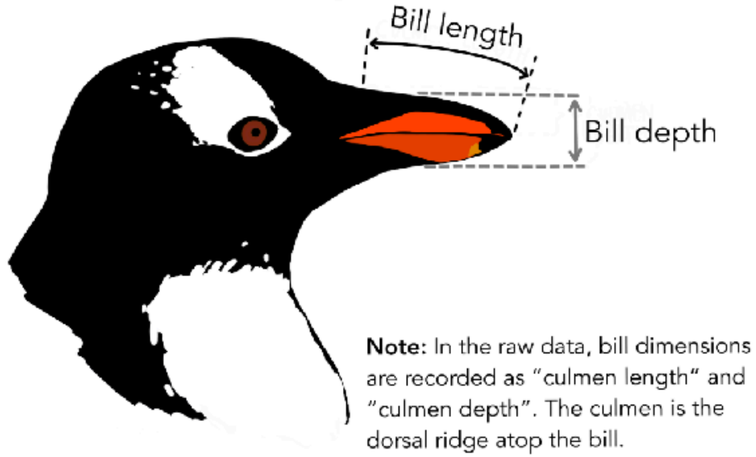


Table 1 matches Figure 2!

Mine Çetinkaya-Rundel

2020-11-10

In this report we evaluate the relationship between relationship between bill depth and flipper length of penguins. The data come from Gorman, Williams, and Fraser (2014).



Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

The original dataset has 3 species (Adelie, Gentoo, and Chinstrap), but we will only work with Adelie and Chinstrap species.

1 Exploratory data analysis

Table 1.1 shows some summary statistics.

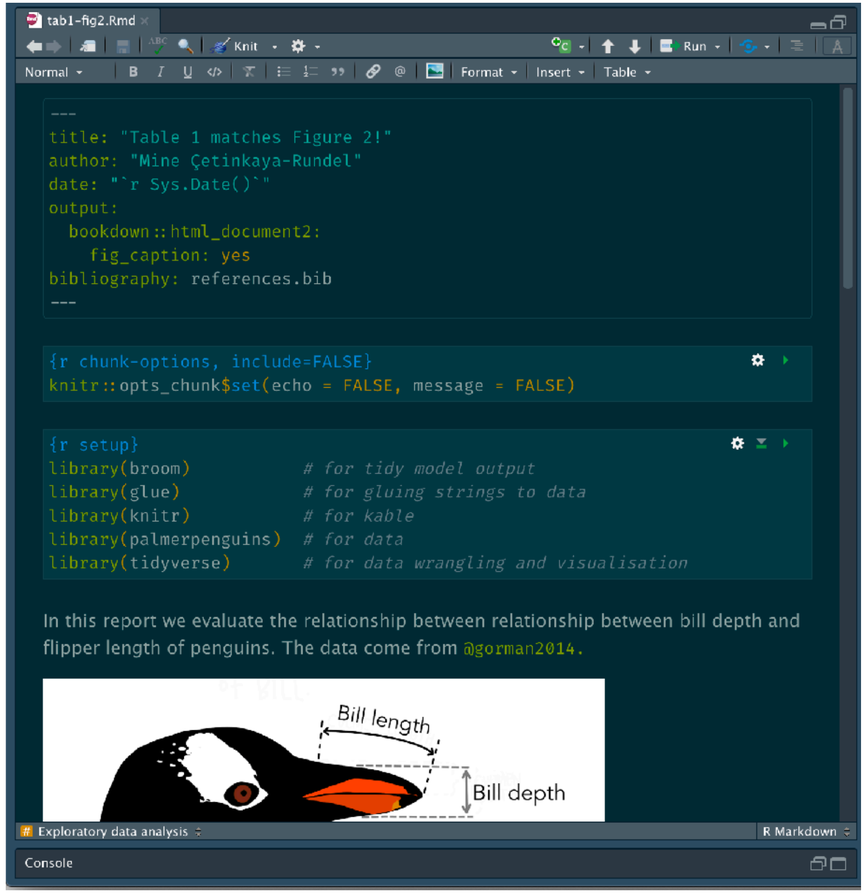
Table 1.1: Summary statistics


demo
rmarkdown

more resources...

- Learn more about **R Markdown**:
  - Documentation: rmarkdown.rstudio.com
  - Book: bookdown.org/yihui/rmarkdown
  - Book: bookdown.org/yihui/rmarkdown-cookbook
- Learn more about the **visual editor**:
  - Documentation: rstudio.github.io/visual-markdown-editing

use version control



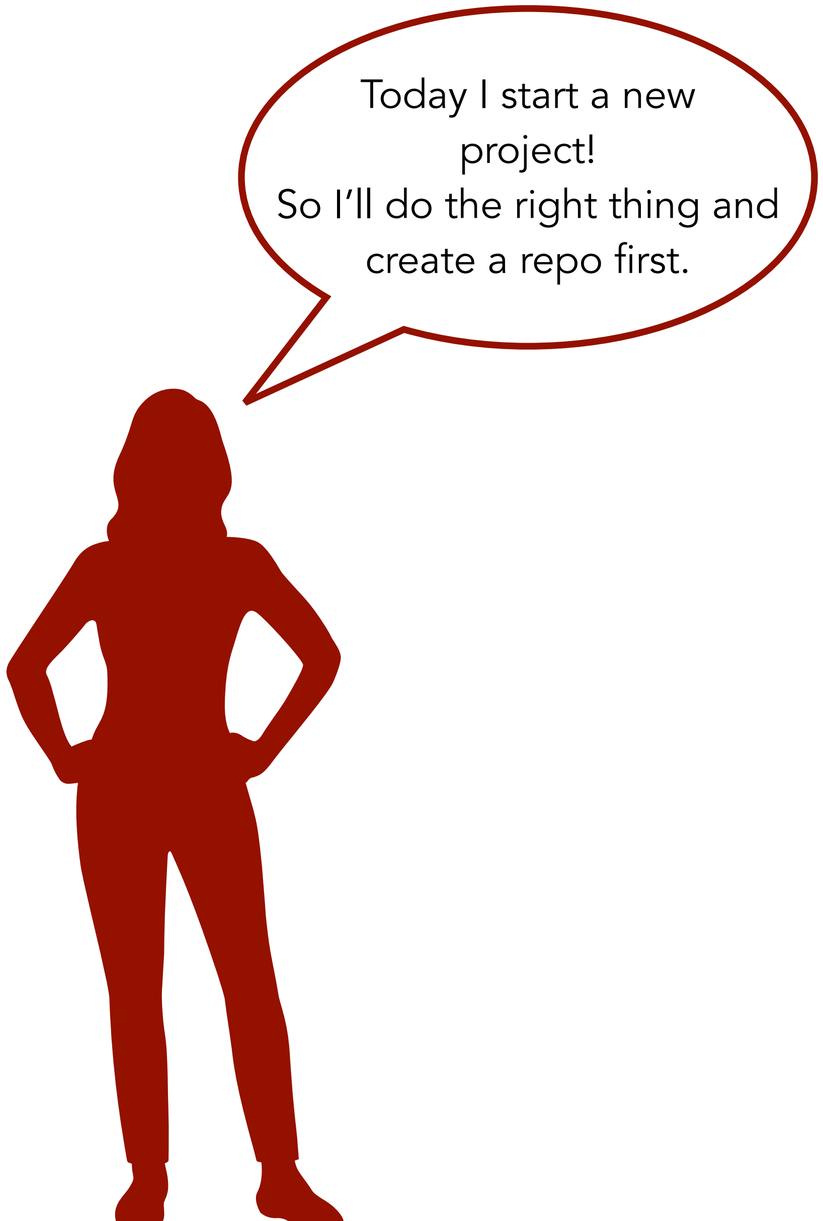
changes tracked by

hosted on

2 Git workflows

- GitHub first
- Local first

GitHub first



- Step 1: Create a new repo on GitHub
- Step 2: Copy the repo URL
- Step 3: Clone it using RStudio
- Step 4: Make changes locally
- Step 6: Commit and push to GitHub
- Step 7: Confirm your changes have propagated to GitHub

Local first

- Step 1: Create an RStudio Project from existing directory (if an `.Rproj` file doesn't already exist)
- Step 2: `usethis::use_git()` and follow instructions
- Step 3: `usethis::use_github()` and follow instructions


demo
git & github

more resources...

- Learn more about using **Git and GitHub with R**:
  - Book: happygitwithr.com
- Learn more about **Git setup**:
  - Documentation: usethis.r-lib.org/articles/articles/usethis-setup.html

7

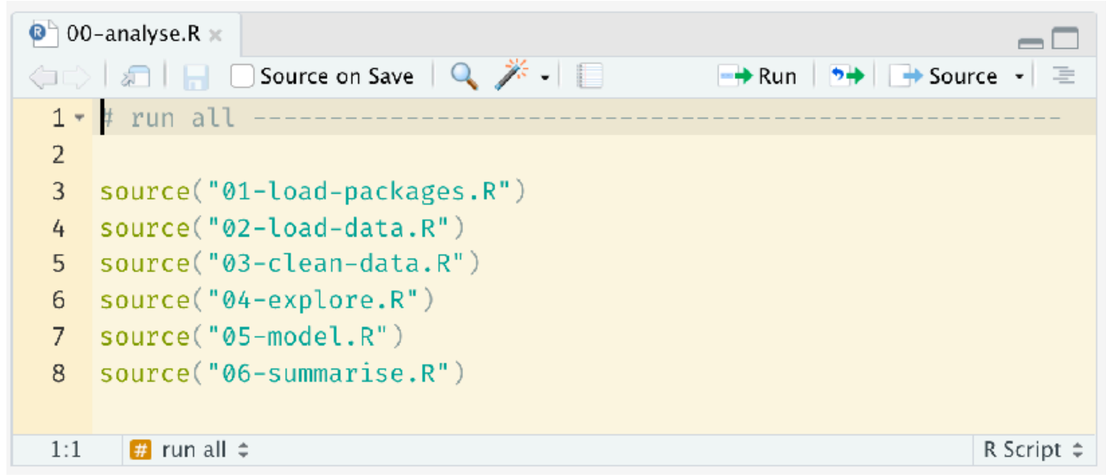
automate your process

raw-data

processed-data

scripts

- 00-analyse.R
- 01-load-packages.R
- 02-load-data.R
- 03-clean-data.R
- 04-explore.R
- 05-model.R
- 06-summarise.R



figures

manuscript

recommended reading

minimal make A minimal tutorial on make

I would argue that the most important tool for reproducible research is not Sweave or knitr but *GNU make*.

Consider, for example, all of the files associated with a manuscript. In the simplest case, I would have an R script for each figure plus a LaTeX file for the main text. And then a BibTeX file for the references.

Compiling the final PDF is a bit of work:

- Run each R script through R to produce the relevant figure.
- Run latex and then bibtex and then latex a couple of more times.

And the R scripts need to be run before latex is, and only if they've changed.

A simple example

GNU make makes this easy. In your directory for the manuscript, you create a text file called `Makefile` that looks something like the following (here using pdflatex).

```
mypaper.pdf: mypaper.bib mypaper.tex Figs/fig1.pdf Figs/fig2.pdf
    pdflatex mypaper
    bibtex mypaper
    pdflatex mypaper
    pdflatex mypaper

Figs/fig1.pdf: R/fig1.R
    cd R;R CMD BATCH fig1.R

Figs/fig2.pdf: R/fig2.R
    cd R;R CMD BATCH fig2.R
```

Each batch of lines indicates a file to be created (the *target*), the files it depends on (the *prerequisites*), and then a set of commands needed to construct the target from the dependent files. Note that the lines with the commands *must* start with a **tab** character (**not spaces**).

Karl Broman. "Minimal Make", kbroman.org/minimal_make.

recommended reading

The targets R Package User Manual

Will Landau

Chapter 1 Introduction

The `targets` package is a Make-like pipeline toolkit for Statistics and data science in R. With `targets`, you can maintain a reproducible workflow without repeating yourself. `targets` learns how your pipeline fits together, skips costly runtime for tasks that are already up to date, runs only the necessary computation, supports implicit parallel computing, abstracts files as R objects, and shows tangible evidence that the results match the underlying code and data.

1.1 Motivation

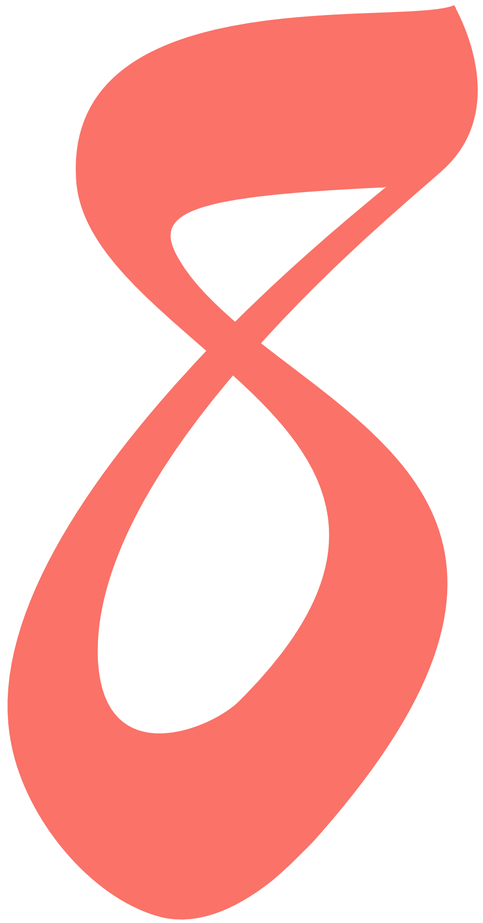
Data analysis can be slow. A round of scientific computation can take several minutes, hours, or even days to complete. After it finishes, if you update your code or data, your hard-earned results may no longer be valid. Unchecked, this invalidation creates chronic Sisyphean loop:

1. Launch the code.
2. Wait while it runs.
3. Discover an issue.
4. Restart from scratch.

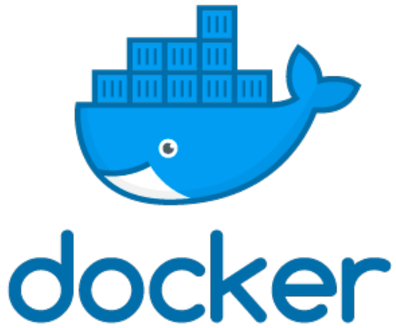
1.2 Pipeline toolkits

Pipeline toolkits like GNU Make break the cycle. They watch the dependency graph of the whole workflow and skip steps, or "targets", whose code, data, and upstream dependencies have not changed since the last run of the pipeline. When all targets are up to date, this is evidence that the results match the underlying code and data, which helps us trust the results and confirm the computation is reproducible.

Will Landau. "The targets R Package User Manual", books.ropensci.org/targets.



share computing environment


docker


binder


R Studio

1 **organize** your project

2 write **READMEs** liberally

3 keep data **tidy & machine readable**

4 **comment** your code

5 use **literate programming**

6 use **version control**

7 **automate** your process

8 share computing **environment**

Greg Wilson, Jennifer Bryan, Karen Cranston,
Justin Kitzes, Lex Nederbragt, Tracy K. Teal
“Good enough practices in scientific computing."
PLoS computational biology 13.6 (2017): e1005510.

PLOS | COMPUTATIONAL BIOLOGY

PERSPECTIVE

Good enough practices in scientific computing

Greg Wilson[1☯*], Jennifer Bryan[2☯], Karen Cranston[3☯], Justin Kitzes[4☯], Lex Nederbragt[5☯], Tracy K. Teal[6☯]

1 Software Carpentry Foundation, Austin, Texas, United States of America, 2 RStudio and Department of Statistics, University of British Columbia, Vancouver, British Columbia, Canada, 3 Department of Biology, Duke University, Durham, North Carolina, United States of America, 4 Energy and Resources Group, University of California, Berkeley, Berkeley, California, United States of America, 5 Centre for Ecological and Evolutionary Synthesis, University of Oslo, Oslo, Norway, 6 Data Carpentry, Davis, California, United States of America

☯ These authors contributed equally to this work.
* gvwilson@software-carpentry.org

Author summary

Computers are now essential in all branches of science, but most researchers are never taught the equivalent of basic lab skills for research computing. As a result, data can get lost, analyses can take much longer than necessary, and researchers are limited in how effectively they can work with software and data. Computing workflows need to follow the same practices as lab projects and notebooks, with organized data, documented steps, and the project structured for reproducibility, but researchers new to computing often don't know where to start. This paper presents a set of good computing practices that every researcher can adopt, regardless of their current level of computational skill. These practices, which encompass data management, programming, collaborating with colleagues, organizing projects, tracking work, and writing manuscripts, are drawn from a wide variety of published sources from our daily lives and from our work with volunteer organizations that have delivered workshops to over 11,000 people since 2010.

OPEN ACCESS

Citation: Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK (2017) Good enough practices in scientific computing. PLoS Comput Biol 13(6): e1005510. https://doi.org/10.1371/journal.pcbi.1005510

Editor: Francis Ouellette, Ontario Institute for Cancer Research, CANADA

Published: June 22, 2017

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Overview

We present a set of computing tools and techniques that every researcher can and should consider adopting. These recommendations synthesize inspiration from our own work, from the experiences of the thousands of people who have taken part in Software Carpentry and Data Carpentry workshops over the past 6 years, and from a variety of other guides. Our recommendations are aimed specifically at people who are new to research computing.

Introduction

Three years ago, a group of researchers involved in Software Carpentry and Data Carpentry wrote a paper called "Best Practices for Scientific Computing" [1]. That paper provided recommendations for people who were already doing significant amounts of computation in their research. However, as computing has become an essential part of science for all researchers, there is a larger group of people new to scientific computing, and the question then becomes, "where to start?"

Improve your workflow for reproducible science

@minebocek
mine-cetinkaya-rundel
cetinkaya.mine@gmail.com