The Importance of Open and Reproducible Research

Eric Crandall

What is Reproducibility?

Gandrud 2014 gives this definition (especially for data analysis and computer science):

> *"The data and code used to make a finding are available and they are presented in such a way that it is (relatively) straightforward for an independent researcher to recreate the finding."*

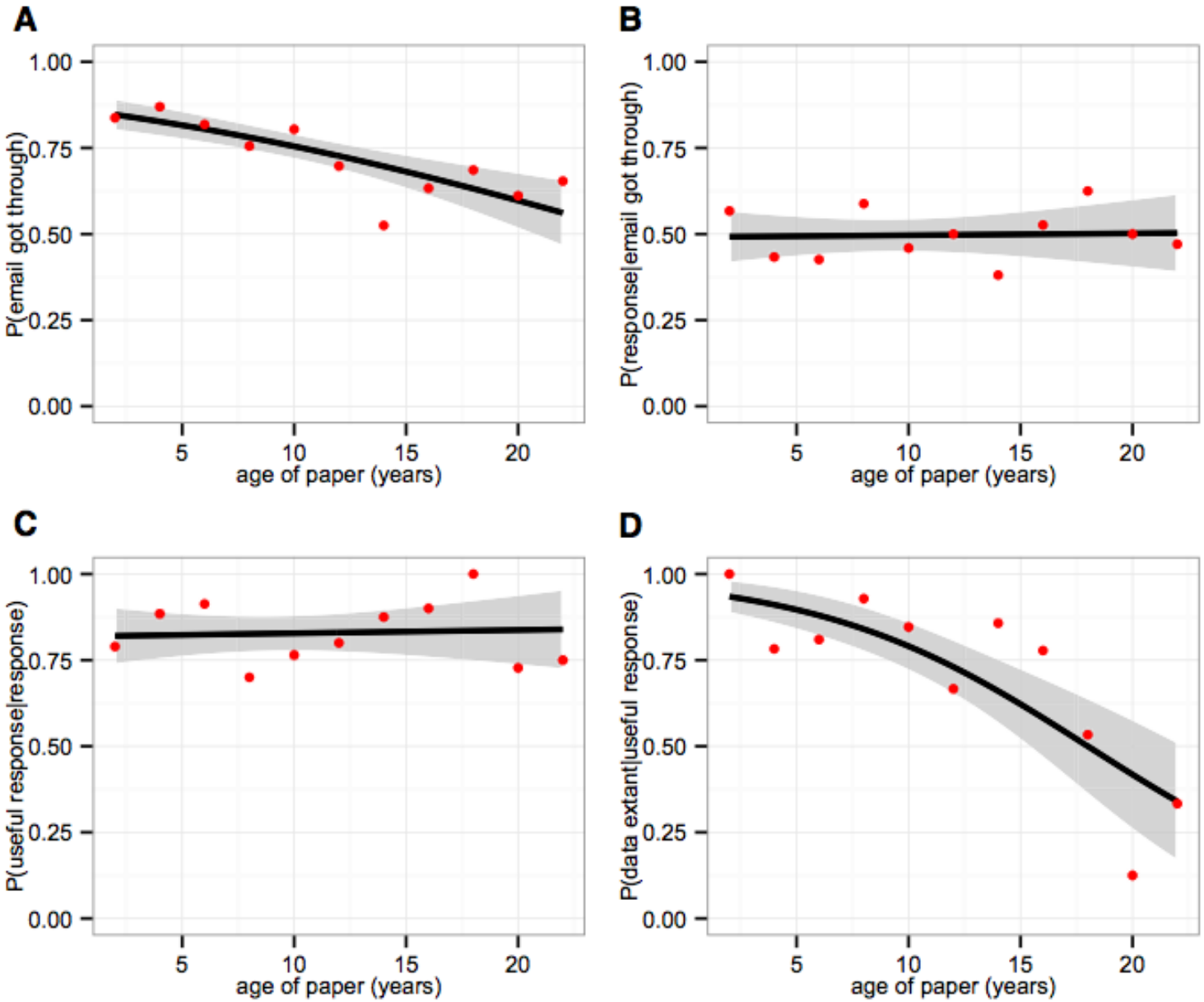
This actually seldom happens.

Consider two interesting articles by Tim Vines: * The Availability of Research Data Declines Rapidly with Article Age + Contacted Authors of 516 datasets with morphological data for discriminant analysis published between 1991 and 2011 + Received only 101 datasets!

- *"of 516 articles published between 2 and 22 years ago. . . the odds of a data set being extant fell by 17% per year."*

The Importance of Open and Reproducible Research

Eric Crandall



Data not only need to be easily available, but methods also need to be reproducible

- Gilbert et al. 2012 - Recommendations for utilizing and reporting population genetic analyses: the reproducibility of genetic clustering using the program structure

Data not only need to be easily available, but methods also need to be reproducible

- Gilbert et al. 2012 - Recommendations for utilizing and reporting population genetic analyses: the reproducibility of genetic clustering using the program structure
  - "*we reanalysed data sets gathered from papers using the software package 'structure'. . . 30% of analyses were unable to reproduce the same number of population clusters.*"

Data not only need to be easily available, but methods also need to be reproducible

- Gilbert et al. 2012 - Recommendations for utilizing and reporting population genetic analyses: the reproducibility of genetic clustering using the program structure
  - "*we reanalysed data sets gathered from papers using the software package 'structure'... 30% of analyses were unable to reproduce the same number of population clusters.*"
- Scientific articles have fairly detailed methods sections, but those are typically insufficient to actually reproduce an analysis.

Data not only need to be easily available, but methods also need to be reproducible

- Gilbert et al. 2012 - Recommendations for utilizing and reporting population genetic analyses: the reproducibility of genetic clustering using the program structure
  - "*we reanalysed data sets gathered from papers using the software package 'structure'. . . 30% of analyses were unable to reproduce the same number of population clusters.*"
- Scientific articles have fairly detailed methods sections, but those are typically insufficient to actually reproduce an analysis.
- Scientists owe it to themselves and their community to have an explicit record of all the steps in an analysis done at a computer.

Why are Open Data and Reproducibility Important For Science?

- Standard to judge scientific claims

Why are Open Data and Reproducibility Important For Science?

- Standard to judge scientific claims
- Data and methods need to be openly available in order to be reproducible

Why are Open Data and Reproducibility Important For Science?

- Standard to judge scientific claims
- Data and methods need to be openly available in order to be reproducible
- Avoiding effort duplication

Why are Open Data and Reproducibility Important For Science?

- Standard to judge scientific claims
- Data and methods need to be openly available in order to be reproducible
- Avoiding effort duplication
- Encouraging cumulative knowledge development

Why are Open Data and Reproducibility Important for *You*?

- Better work habits

Gandrud 2014

Why are Open Data and Reproducibility Important for *You*?

- Better work habits
  - better, clearer documentation

Gandrud 2014

Why are Open Data and Reproducibility Important for *You*?

- Better work habits
  - better, clearer documentation
- Better teamwork

Gandrud 2014

Why are Open Data and Reproducibility Important for *You*?

- Better work habits
  - better, clearer documentation
- Better teamwork
- Re-analysis is easier

Gandrud 2014

Why are Open Data and Reproducibility Important for *You*?

- Better work habits
  - better, clearer documentation
- Better teamwork
- Re-analysis is easier
- Higher research impact

Gandrud 2014

Tools for Research Reproducibility

- Open Source Everything

Tools for Research Reproducibility

- Open Source Everything
- R language

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR
- Markdown and LaTeX

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR
- Markdown and LaTeX
- Unix operating system

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR
- Markdown and LaTeX
- Unix operating system
- GitHub and git - version control (not covered in this class)

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR
- Markdown and LaTeX
- Unix operating system
- GitHub and git - version control (not covered in this class)
- Creative Commons Licensing

Tools for Research Reproducibility

- Open Source Everything
- R language
- Rstudio and knitR
- Markdown and LaTeX
- Unix operating system
- GitHub and git - version control (not covered in this class)
- Creative Commons Licensing
- Online Repositories (Dryad, Genbank,GBIF, GEOME, new ones all the time)

Open Science

Open Science Links

- Open Data

Figure 3: Big Data

Open Science

Open Science Links

- Open Data
- Reproducible Methods

Figure 3: Big Data

Open Science

Open Science Links

- Open Data
- Reproducible Methods
- Open Access Publications

Figure 3: Big Data

Reproducible Research Habits

Good habits to get into as a student!

10 Things Every Graduate Student Should Do By Carly Strasser

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure
  - Stops you from learning better ways to handle data

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure
  - Stops you from learning better ways to handle data
- Proprietary software

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure
  - Stops you from learning better ways to handle data
- Proprietary software
- Easy to mess up your data, no provenance

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure
  - Stops you from learning better ways to handle data
- Proprietary software
- Easy to mess up your data, no provenance
- Dates!

Stop Using Excel!

- OK, maybe not entirely - its good for quick visualizing, data entry, etc.
- It tends to be a crutch.
  - Stops you from thinking carefully about your data structure
  - Stops you from learning better ways to handle data
- Proprietary software
- Easy to mess up your data, no provenance
- Dates!
- At least keep your raw data in text format

Learn to Code

Any language.

R is a great starting place.

Here is code to paste cells from excel into an R data frame!

```
data <- read.table(pipe("pbpaste"),header=T)
```

Make a plan for managing data in each project

- Keep all data (and ideally analyses) in a text file

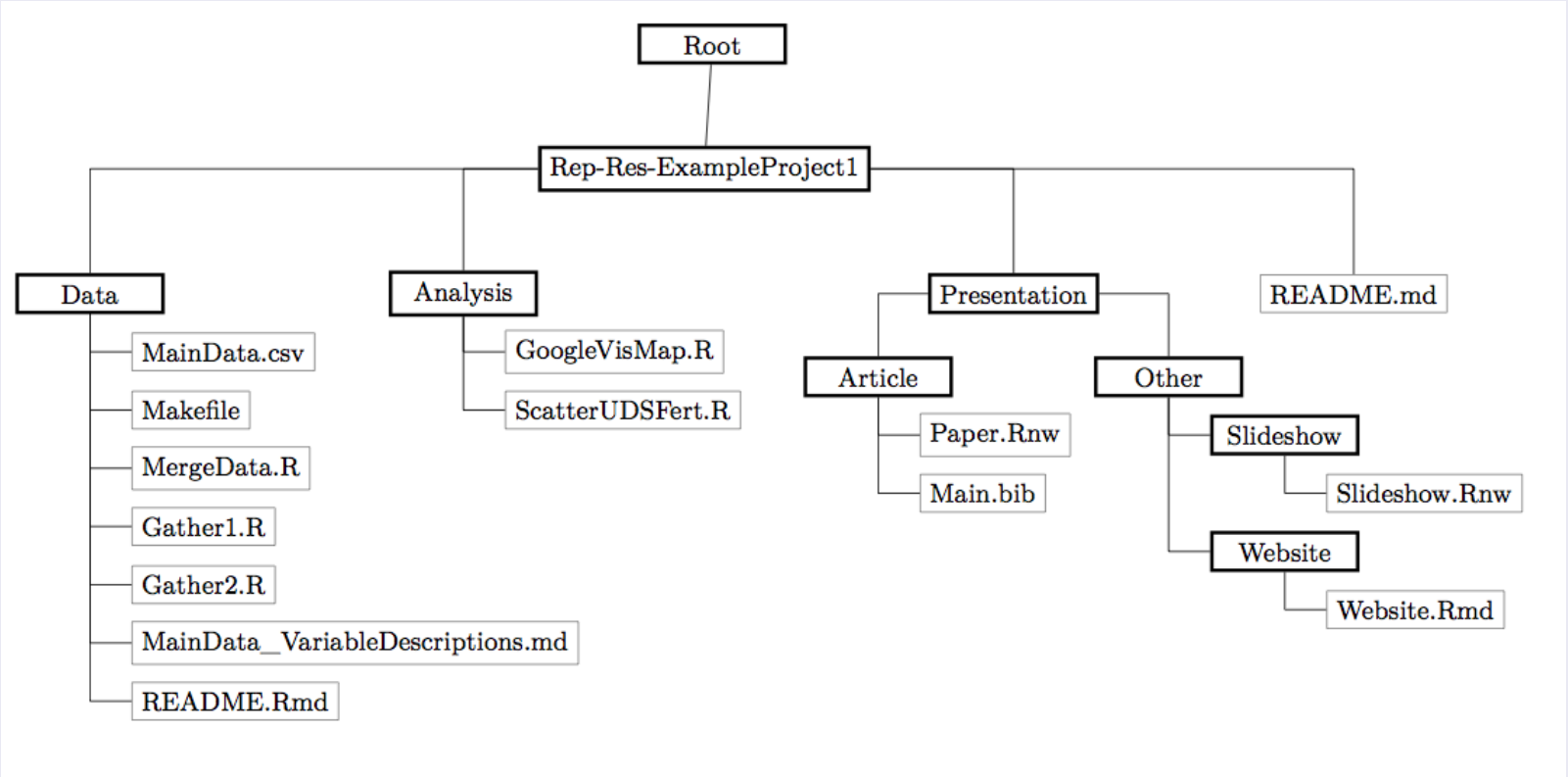


Figure 4: Gandrud Figure 1

Make a plan for managing data in each project

- Keep all data (and ideally analyses) in a text file
- Think about your file structure

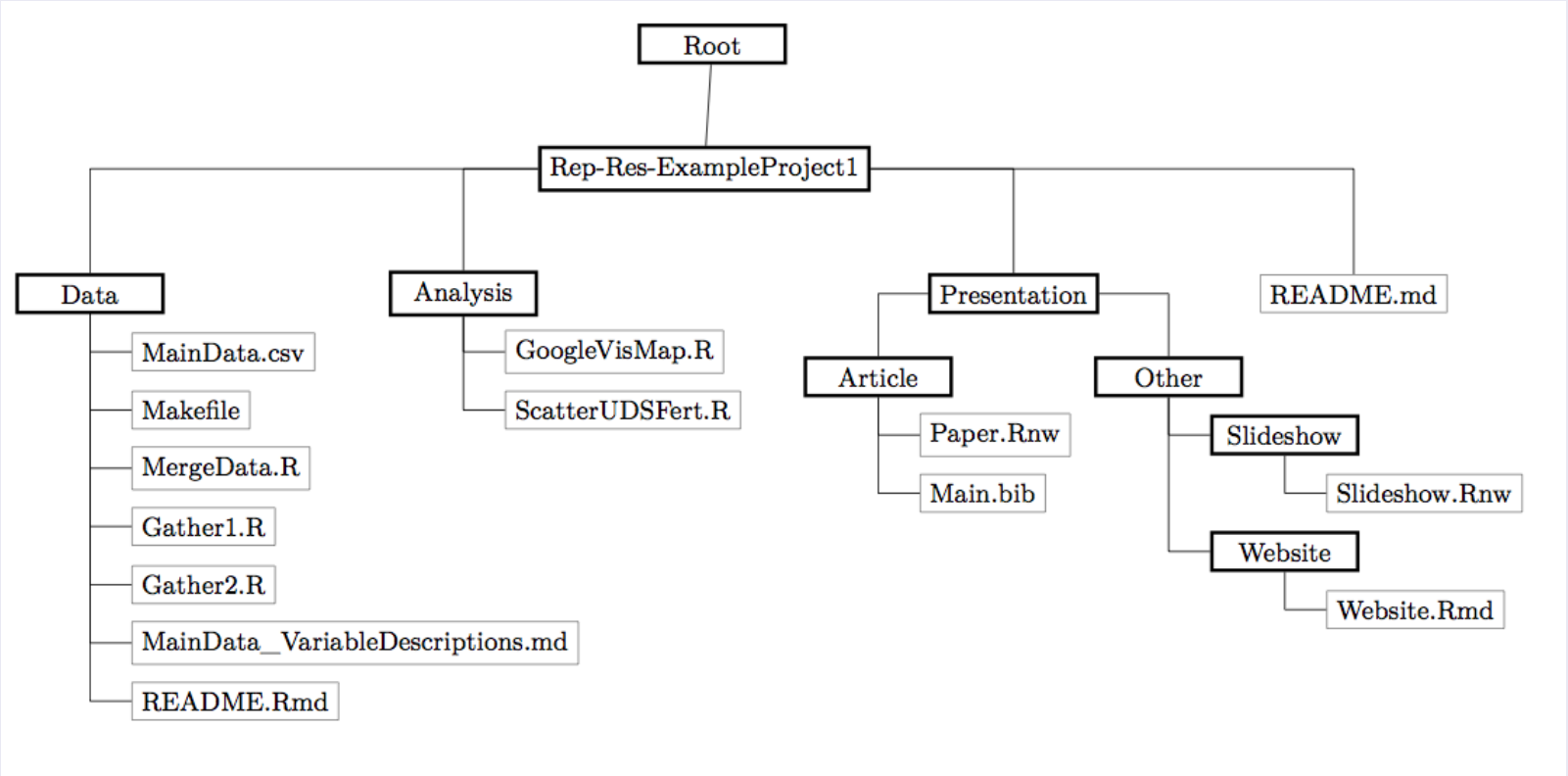


Figure 4: Gandrud Figure 1

Make a plan for managing data in each project - 2

- Document everything

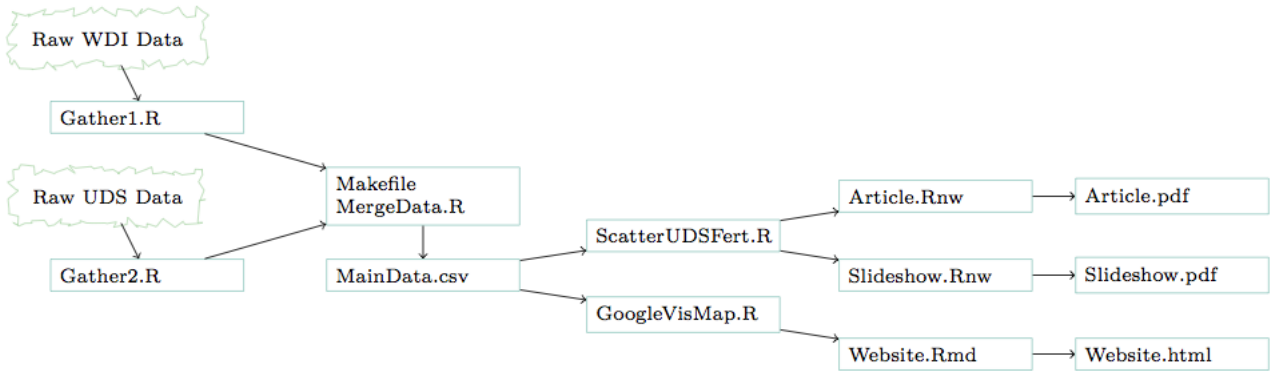
Make a plan for managing data in each project - 2

- Document everything
- Explicitly tie your files together

Make a plan for managing data in each project - 2

- Document everything
- Explicitly tie your files together
- Data management plans are now explicitly required by the NSF and other funding agencies!!!

Short Example Main File Ties



:

`download.file` `read.csv` *knitr*
`Make` `source`
`merge`
`WDI`

Keep an electronic (online) notebook

- Document everything!!

Keep an electronic (online) notebook

- Document everything!!
  - repeat your own analysis!!

Keep an electronic (online) notebook

- Document everything!!
  - repeat your own analysis!!
  - show others what you did

Keep an electronic (online) notebook

- Document everything!!
  - repeat your own analysis!!
  - show others what you did
- Dokuwiki is great (keeps things in plain text)

Keep an electronic (online) notebook

- Document everything!!
  - repeat your own analysis!!
  - show others what you did
- Dokuwiki is great (keeps things in plain text)
- I now use GitHub and Rmarkdown notebooks

Keep an electronic (online) notebook

- Document everything!!
  - repeat your own analysis!!
  - show others what you did
- Dokuwiki is great (keeps things in plain text)
- I now use GitHub and Rmarkdown notebooks
- Go open if you're brave

Communicate Your Science

Start doing outreach now!

NSF vs. House Committee on Science

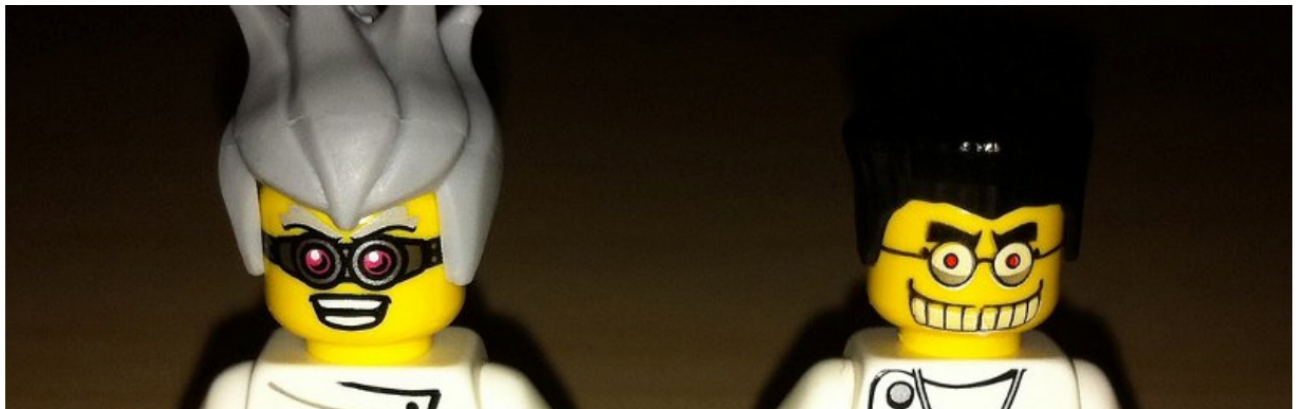
KQED Science npr PBS

Radio | TV | News | Arts | Food | Science | Education

Why Scientists are Seen as Competent but Untrustworthy (and Why it Matters)

Dr. Barry Starr, Tech Museum & Stanford University | October 6, 2014 | 1 Comment

Share: Print



Publish Open-Access Articles

- Biology Failed the Internet

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).
  - Therefore you may legally post a pre-print

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).
  - Therefore you may legally post a pre-print
  - BiorXiv

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).
  - Therefore you may legally post a pre-print
  - BiorXiv
  - PeerJ

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).
  - Therefore you may legally post a pre-print
  - BiorXiv
  - PeerJ
  - Most universities now have pre-print platforms

Publish Open-Access Articles

- Biology Failed the Internet
  - Physics moved to pre-print servers a long time ago!
  - Open access journals (e.g. PLoS) were supposed to be a stop-gap measure!
- Many schools now have open-access funds
- The dark side of open-access
- Most journals only own the "typesetting" (because that is all they did!).
  - Therefore you may legally post a pre-print
  - BiorXiv
  - PeerJ
  - Most universities now have pre-print platforms
- ResearchGate, Academia.edu etc. are social networks whereby you are sharing articles with your "friends"

Rstudio

R studio is an Integrated Developer Environment for R * Is an IDE (integrated development environment) that *sits on top of* R and makes it easier to interact with R. * Organizes your work in R in neatly-contained packages of work (typically data and code) called “projects” * Nothing mysterious about these—just collections of files stored together in a single directory on your computer.

Git

- A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:

Git

- A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:
  - Which changes were made?

Git

- A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:
  - Which changes were made?
  - Who made the changes?

Git

- A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:
  - Which changes were made?
  - Who made the changes?
  - When were the changes made?

Git

- A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:
  - Which changes were made?
  - Who made the changes?
  - When were the changes made?
  - Why were changes needed?

Github is a distributed version control system (DVCS)

- DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.

Github is a distributed version control system (DVCS)

- DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.
- Unlike once popular centralized version control systems, DVCSs like Git don't need a constant connection to a central repository. Developers can work anywhere and collaborate asynchronously from any time zone.

Github is a distributed version control system (DVCS)

- DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.
- Unlike once popular centralized version control systems, DVCSs like Git don't need a constant connection to a central repository. Developers can work anywhere and collaborate asynchronously from any time zone.

Github is a distributed version control system (DVCS)

- DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.
- Unlike once popular centralized version control systems, DVCSs like Git don't need a constant connection to a central repository. Developers can work anywhere and collaborate asynchronously from any time zone.

What's a repository?

- A *repository*, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history.

What's a repository?

- A *repository*, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history.
- The file history appears as snapshots in time called *commits*, and can be organized into multiple lines of development called branches.

What's a repository?

- A *repository*, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history.
- The file history appears as snapshots in time called *commits*, and can be organized into multiple lines of development called branches.
- Because Git is a DVCS, repositories are self-contained units and anyone who owns a copy of the repository can access the entire codebase and its history.

What's a repository?

- A *repository*, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history.
- The file history appears as snapshots in time called *commits*, and can be organized into multiple lines of development called branches.
- Because Git is a DVCS, repositories are self-contained units and anyone who owns a copy of the repository can access the entire codebase and its history.
- Using the command line or other ease-of-use interfaces, a git repository also allows for: interaction with the history, cloning, creating branches, committing, merging, comparing changes across versions of code, and more.

What's a repository?

- Working in repositories keeps development projects organized and protected. Developers are encouraged to fix bugs, or create fresh features, without fear of derailing mainline development efforts.

What's a repository?

- Working in repositories keeps development projects organized and protected. Developers are encouraged to fix bugs, or create fresh features, without fear of derailing mainline development efforts.
- Through platforms like GitHub, Git also provides more opportunities for project transparency and collaboration. Public repositories help teams work together to build the best possible final product.

The Github Flow

Video

The Github Flow

Video

The Github Flow

Video

The Github Flow

Video

The Github Flow

Video

The Github Flow

Video

Alternatives to Github exist!

- Bitbucket

Let's make a new project from a Github repository

Do This in RStudio:

Alternatives to Github exist!

- Bitbucket
- Gitlab

Let's make a new project from a Github repository

Do This in RStudio:

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project
2. Create project from: Choose "Version Control"

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project
2. Create project from: Choose "Version Control"
    1. Choose "Git"

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project
2. Create project from: Choose "Version Control"
    1. Choose "Git"
    2. Input the url of this repository
       1.https://github.com/ericcrandall/reproducible_research

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project
2. Create project from: Choose "Version Control"
    1. Choose "Git"
    2. Input the url of this repository 1.https://github.com/ericcrandall/reproducible_research
    3. and put it somewhere in your home directory (browse to where you want to put it in the "create project as subdirectory of:")

Alternatives to Github exist!

- Bitbucket
- Gitlab
- Gitbucket

Let's make a new project from a Github repository

Do This in RStudio:

1. Select File -> New Project
2. Create project from: Choose "Version Control"
    1. Choose "Git"
    2. Input the url of this repository 1.https://github.com/ericcrandall/reproducible_research
    3. and put it somewhere in your home directory (browse to where you want to put it in the "create project as subdirectory of:")
    4. Hit `Create Project`

Intro to Rmarkdown

- Designed as a text *markup* language that would be

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive
  - Intuitive

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive
  - Intuitive
  - Capable of conveying intent even without being compiled into HTML or PDF

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive
  - Intuitive
  - Capable of conveying intent even without being compiled into HTML or PDF
- There are many Markdown interpreters. The Rstudio folks have been using pandoc to crunch Markdown into other formats. It provides many useful extensions.

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive
  - Intuitive
  - Capable of conveying intent even without being compiled into HTML or PDF
- There are many Markdown interpreters. The Rstudio folks have been using pandoc to crunch Markdown into other formats. It provides many useful extensions.
- Customizations of style are mostly separate from the **content**.

Intro to Rmarkdown

- Designed as a text *markup* language that would be
  - Simple
  - Expressive
  - Intuitive
  - Capable of conveying intent even without being compiled into HTML or PDF
- There are many Markdown interpreters. The Rstudio folks have been using pandoc to crunch Markdown into other formats. It provides many useful extensions.
- Customizations of style are mostly separate from the **content**.
- This presentation was made in Rmarkdown!

The Importance of Open and Reproducible Research

Eric Crandall