How to Teach Reproducibility in Classwork

Western Economic Association International

Fernando Hoces,

Berkeley Initiative for Transparency in the Social Sciences

27 June 2020 | slides

(note: use the left and right arrow keys to navigate through slides)

Today's Presentation

Part I: Accelerating Computational Reproducibility in Economics (X)

`[10 min break]`

Part II: Hands-on Practices for Computational Reproducibility (90' - X)

Target Audience:

- Instructors of Empirical/Applied Courses in Economics (and related) PhDs
- Advisers of undergraduate students
- Researchers interested in conducting reproductions

Part I: Accelerating Computational Reproducibility in Economics (ACRE)

Table of Contents for Part I

1. **BITSS**
2. Reproducibility
3. ACRE Guidelines
4. ACRE Platform

Table of Contents for Part I

1. BITSS
2. **Reproducibility**
3. ACRE Guidelines
4. ACRE Platform

Motivation 1: "Reproducibility Crisis"

Replication in Social Sciences (same method, different sample)	Reproduction in Economics (same data and methods)
OSC (2015): 30%-60%	Chang & Li (2015): 43%
Camerer et. al. (2016): ~60%	Gertler et. al. (2017): 14%
Nosek & Camerer et. al. (2018): ~60%	Kingi et. al. (2018): 43%
Klein et. al. (2018): 50%	Wood et. al. (2018): 25%

M2: More Inclusive Concept Scholarly Output

> Clarebout Principle:
>
> “An article about computational science in a scientific publication is not the scholarship itself, it’s merely scholarship advertisement. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”
>
> *Buckheit and D.L. Donoho (1995, 2009)*

M2: More Inclusive Concept Scholarly Output

> Clarebout Principle:
>
> “An **article** about computational science in a scientific publication is not the scholarship itself, **it’s merely scholarship advertisement**. The actual **scholarship is the complete software development environment and the complete set of instructions which generated the figures**.”
>
> *Buckheit and D.L. Donoho (1995, 2009)*

M2: More Inclusive Concept Scholarly Output

Potential benefits of following the Clarebout Principle

Well discussed potential positive effects on:

- Pedagogy
- Incremental generation of knowledge

Under discussed:

- Possible positive effect on diversity, equity and inclusion: no connections or language skills ("appropriate politeness") required to obtain materials

M3: Prevent Loss of Knowledge

Every semester, graduate students around **the world** take an Empirical/Applied [...] Economics course. A typical assignment consists of reproducing the results of a paper and, possibly, testing the robustness of its results.

Stage	**New Knowledge**
Scope (select and verify)	Data and code exist?
Assess	Degree of reproducibility for specific part of the paper
Improve	E.g. fixed paths, libraries, added missing files, etc.
Test robustness	Results are robust to additional specifications

Table of Contents for Part I

1. BITSS
2. Reproducibility
3. **ACRE Guidelines**
4. ACRE Platform

Context for ACRE

- American Economics Association (AEA) creates first data policy in 2006.
    - Must publish some data (waivers available)
- AEA updates policy in 2019.
    - Must post all data and code. Publication is conditional on verifying reproducibility (if confidential: must document extensively)
    - A new requirement is to post all cleaning code, even for data that is not public
    - See Lars Vilhuber's presentation after this one (same zoom channel) for more information x
- We should expect high levels of computational reproducibility after 2019 (AEA).
- We should not demand 100% reproducibility before, but we could identify the gaps and try to improve some.

Beyond Binary Judgments

Reproductions can easily gravitate towards adversarial exchanges.

- Early career researcher (ECR) have incentives to emphasize unsuccessful reproductions
- Original authors have a more senior position and can use it to deter in-depth reproductions from ECRs.
- The media also focuses on eye-catching headlines

Our approach:

We do not want to say

> "Paper X is (ir)reproducible"

We do want to say

> "Paper X's result Y has a high/low **level** of reproducibility according to **several** reproduction attempts. Moreover, **improvements** have been made to the original reproduction package, **increasing** its reproducibility to a higher level"

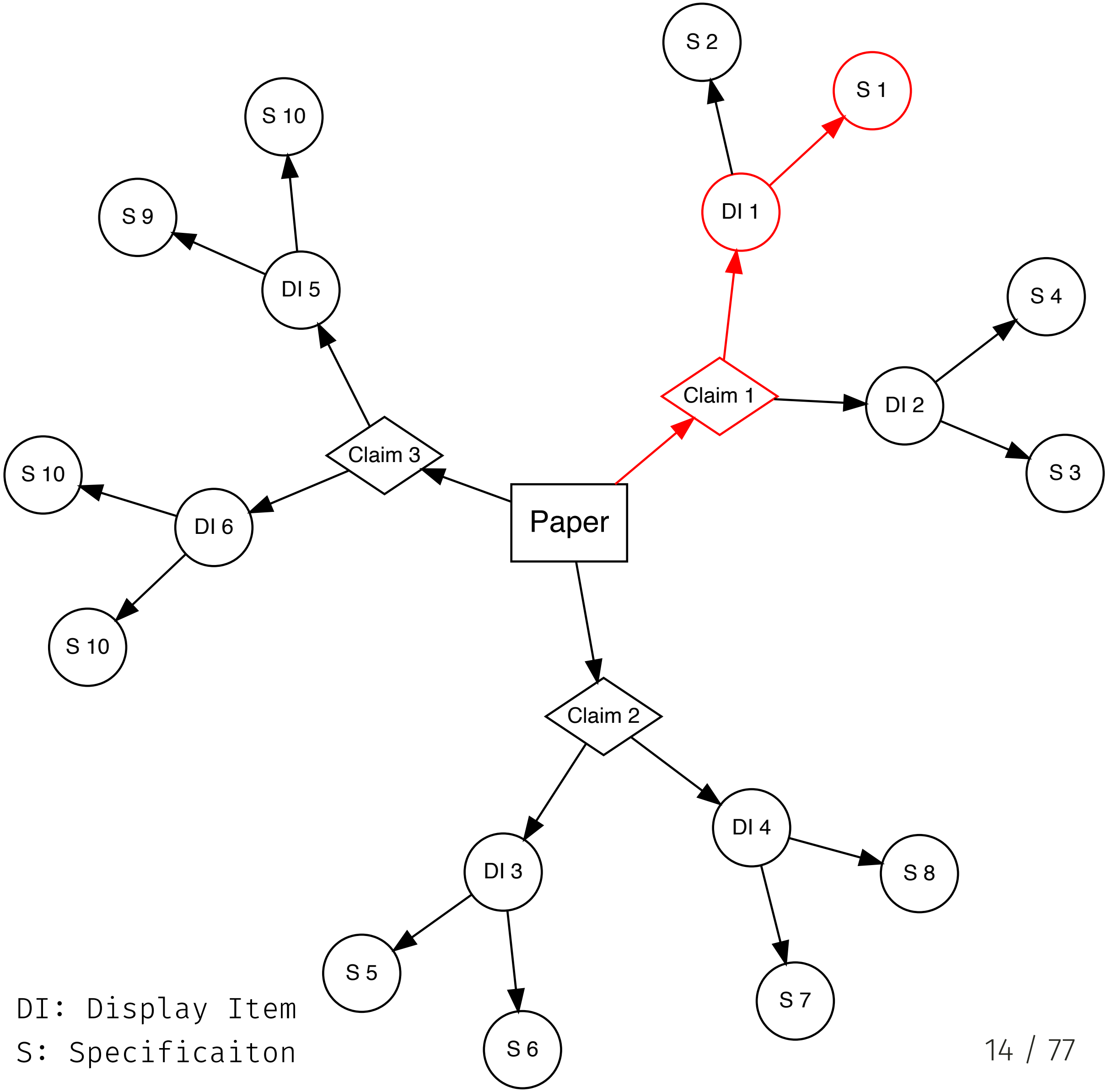
Our Framework

Each **reproduction attempt** is centered around scientific **claims**

One paper can contain several claims.

Each claim may be supported by various **display items** (tables, figures & inline)

A reproduction attempt is at the claim level, and reproducers must record their **specifications** of interest.

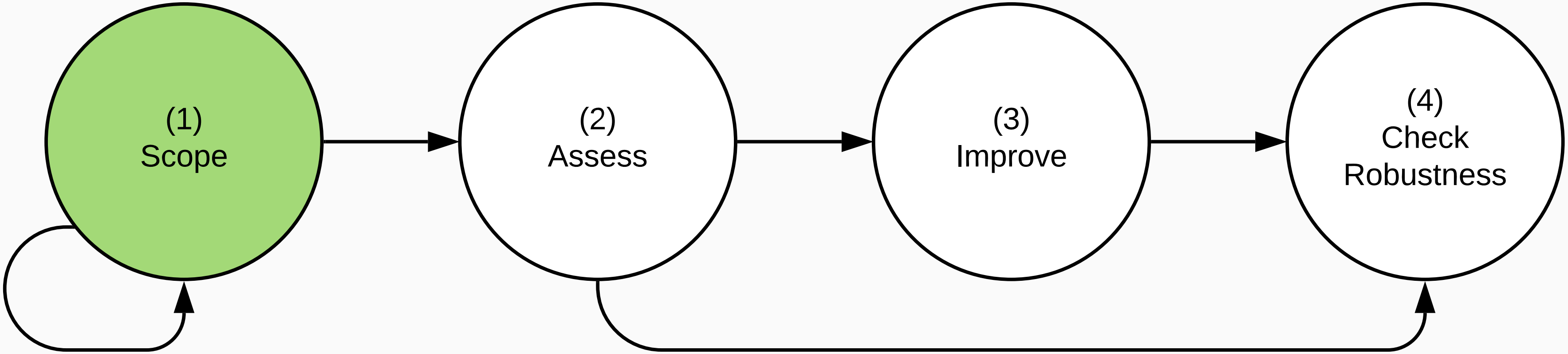


DI: Display Item
S: Specificaiton

Large part of this exercise is about standardization

- **Computational Reproduction (or Reproduction)**
- **Replication (will not mention this term again!)**
- Reproduction attempt
- **Reproduction package**
- Claim
- Display item
- Specification
- Preferred specification
- **Raw data**
- **Analysis data**
- **Candidate paper**
- **Declared paper**
- Reproduction tree
- Complete Workflow
- Computationally Reproducible from Analytic data (CRA)
- Computationally Reproducible from Raw data (CRR)
- Reasonable test
- Feasible test
- Minimal effort

Stages



Scoping

1. Select or be assigned a candidate paper
2. Check ACRE Platform for previous entries and verify availability of reproduction package (RP)
3. If not RP, leave a short record, and repeat with a different candidate paper
4. Once RP is found then candidate becomes declared paper
5. Only then: read the paper and select claim(s), display items and specification to reproduce

Box 1: Summary Report Card for ACRE Paper Entry

Title: Sample Title

Authors: Jane Doe & John Doe

Original Reproduction Package Available: URL/No

[If "No"]

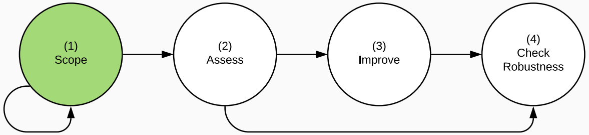
Contacted Authors?: Yes/No

[If "Yes(contacted)"]

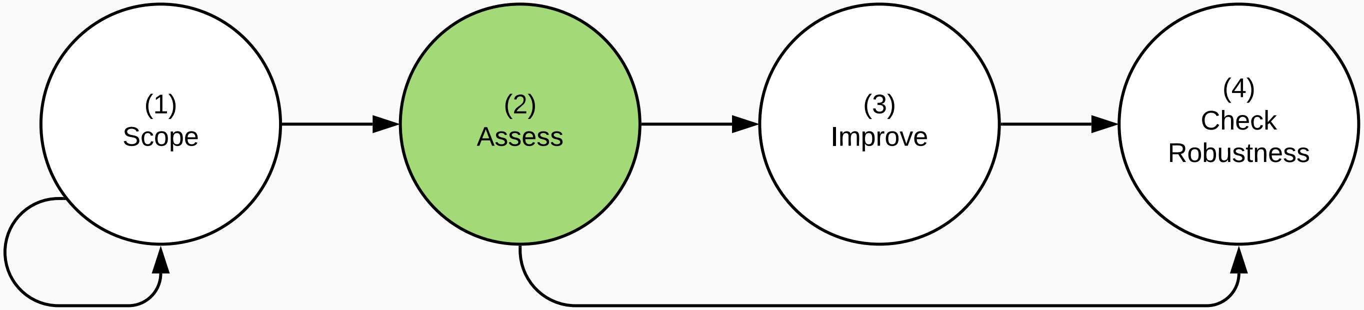
Type of Response: Categories (6).

Additional Reproduction Packages: Number (eg., 2)

Authors Available for Further Questions for ACRE Reproductions: Yes/No/Unknown



Assessment



Two main parts for assessment:

1. Find all the elements behind a display item
2. Score the reproducibility of that display item

Identify All the Elements Behind a Display Item

Reproducers will be asked to draw a clear connection to the raw data sources mentioned in the paper and the display item under reproduction.

Data sources

Connect the data sources in the paper's text with specific raw data files.

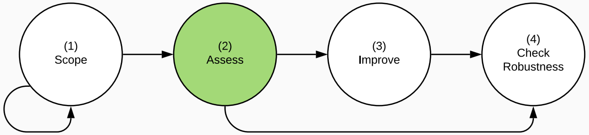
Analytic data sets

Describe each analytic data file.

Code files

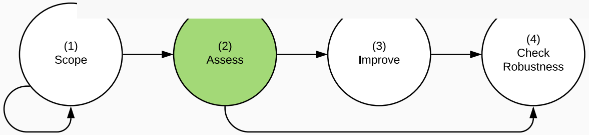
Inspect all code files and record all their inputs and outputs.

With all the information recorded above, reproducers can use the *ACRE Diagram Builder* to generate a **reproduction tree**.



Reproduction Tree

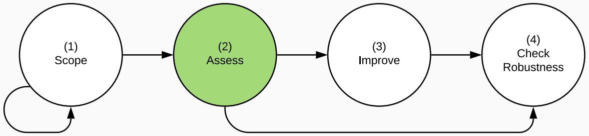
```
table1.tex
    |___[code] analysis.R
        |___analysis_data.dta
            |___[code] final_merge.do
                |___cleaned_1_2.dta
                |   |___[code] clean_merged_1_2.do
                |       |___merged_1_2.dta
                |           |___[code] merge_1_2.do
                |               |___cleaned_1.dta
                |               |   |___[code] clean_raw_1.py
                |               |       |___raw_1.dta
                |               |___cleaned_2.dta
                |                   |___[code] clean_raw_2.py
                |                       |___raw_2.dta
                |___cleaned_3_4.dta
                    |___[code] clean_merged_3_4.do
                        |___merged_3_4.dta
                            |___[code] merge_3_4.do
                                |___cleaned_3.dta
                                |   |___[code] clean_raw_3.py
                                |       |___raw_3.dta
                                |___cleaned_4.dta
                                    |___[code] clean_raw_4.py
                                        |___raw_4.dta
```



Levels

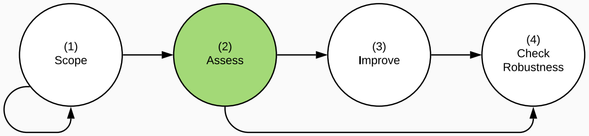
```
                      Levels of Computational Reproducibility
                     (P denotes "partial", C denotes "complete")

                                 | Availability of materials, and reproducibility |
                                 |------------------------------------------------|
                                 |Analysis| Analysis|     | Cleaning| Raw   |     |
                                 |Code    | Data    | CRA | Code    | Data  | CRR |
                                 | P | C  | P  | C  |     | P  |  C | P | C |     |
                                 ---------|---------|-----|---------|-------|-----|
L1: No materials.................| -   -  | -    -  |  -  |  -    - | -   - |  -  |
---------------------------------|--------|---------|-----|---------|-------|-----|
L2: Only code ...................| ✔   ✔  | -    -  |  -  |  -    - | -   - |  -  |
L3: Partial analysis data & code.| ✔   ✔  | ✔    -  |  -  |  -    - | -   - |  -  |
L4: All analysis data & code.....| ✔   ✔  | ✔    ✔  |  -  |  -    - | -   - |  -  |
L5: Reproducible from analysis...| ✔   ✔  | ✔    ✔  |  ✔  |  -    - | -   - |  -  |
---------------------------------|--------|---------|-----|---------|-------|-----|
L6: Some cleaning code...........| ✔   ✔  | ✔    ✔  |  ✔  |  ✔    - | -   - |  -  |
L7: All cleaning code............| ✔   ✔  | ✔    ✔  |  ✔  |  ✔    ✔ | -   - |  -  |
L8: Some raw data................| ✔   ✔  | ✔    ✔  |  ✔  |  ✔    ✔ | ✔   - |  -  |
L9: All raw data.................| ✔   ✔  | ✔    ✔  |  ✔  |  ✔    ✔ | ✔   ✔ |  -  |
L10:Reproducible from raw data...| ✔   ✔  | ✔    ✔  |  ✔  |  ✔    ✔ | ✔   ✔ |  ✔  |
```

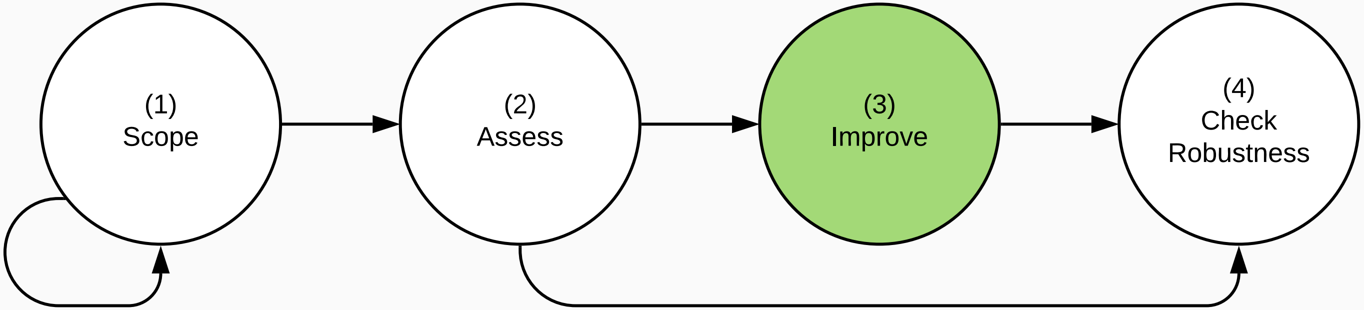


Levels: Proprietary/Confidential Data

```
                        Levels of Computational Reproducibility
                           with Proprietary/Confidential Data
                       (P denotes "partial", C denotes "complete")
                                  | Availability of materials, and reproducibility |
                                  |-----------------------------------------------|
                                  |        | Instr.   |     |         | Instr.|     |
                                  |Analysis| Analysis|     | Cleaning| Raw    |     |
                                  |Code    | Data     | CRA | Code     | Data   | CRR |
                                  | P | C  | P  | C   |     | P  |   C | P | C |     |
                                  ---------|---------|-----|---------|-------|-----|
     L1: No materials.................| -   -  | -    -  |  -  | -    - | -   - |  -  |
     ---------------------------------|--------|---------|-----|---------|-------|-----|
     L2: Only code ...................| ✔   ✔  | -    -  |  -  | -    - | -   - |  -  |
     L3*: Partial analysis data & code| ✔   ✔  | ✔    -  |  -  | -    - | -   - |  -  |
     L4*: All analysis data & code....| ✔   ✔  | ✔    ✔  |  -  | -    - | -   - |  -  |
     L5*: Proof of third party CRA....| ✔   ✔  | ✔    ✔  |  ✔  | -    - | -   - |  -  |
     ---------------------------------|--------|---------|-----|---------|-------|-----|
     L6: Some cleaning code...........| ✔   ✔  | ✔    ✔  |  ✔  | ✔    - | -   - |  -  |
     L7: All cleaning code............| ✔   ✔  | ✔    ✔  |  ✔  | ✔    ✔ | -   - |  -  |
     L8*: Some instr. for raw data....| ✔   ✔  | ✔    ✔  |  ✔  | ✔    ✔ | ✔   - |  -  |
     L9*: All instr. for raw data.....| ✔   ✔  | ✔    ✔  |  ✔  | ✔    ✔ | ✔   ✔ |  -  |
     L10*:Proof of third party CRR....| ✔   ✔  | ✔    ✔  |  ✔  | ✔    ✔ | ✔   ✔ |  ✔  |
```



Improvements

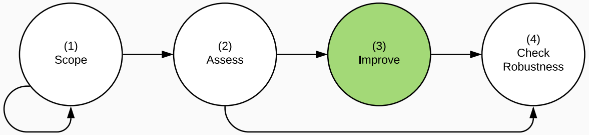


Three types of improvements:

1. Improvements at the paper level
2. Improvements at the display-item level
3. Specific future improvements

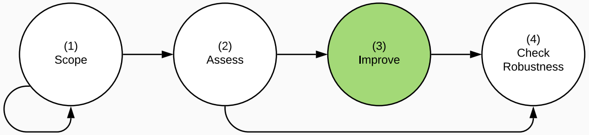
Improvements: Paper-level

- Use version control software (Git/Github).
- Improve documentation: comments, indentations, object names, etc.
- Re-organize the reproduction package into a set of folders and sub-folders that follow standardized best practices, and add a master script that executes all the code in order, with no further modifications. See AEA's reproduction template.
- Literate programming environment (e.g., Jupyter notebooks, RMarkdown)
- Re-write code using a differenet statistical software (ideally open source, like R, Python, or Julia).
- Set up a computing capsule (e.g., Binder and Code Ocean).



Improvements: Display item-level

- Adding missing raw data: files or meta-data
  - Example: "Add raw temperature and relative humidity data"
- Adding missing analytic data files
  - Example: "Copy the row files from Data folder into new `Analysis\trade cost\Input`"
- Adding missing analysis or cleaning code
  - Example: "Replaced broken Wald bootstrap code with updated code/command"
- Debugging code
  - Example: "was counting each group 4 times in round 1, so fixed that"



Improvements: future possible

We ask reproducer to leave concise and actionable tasks for other reproducers in the future.

Example 1:

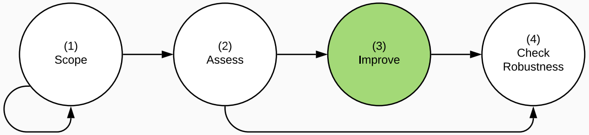
> "Revise the .aml and .bat code scripts to reflect reorganized structure"

Example 2:

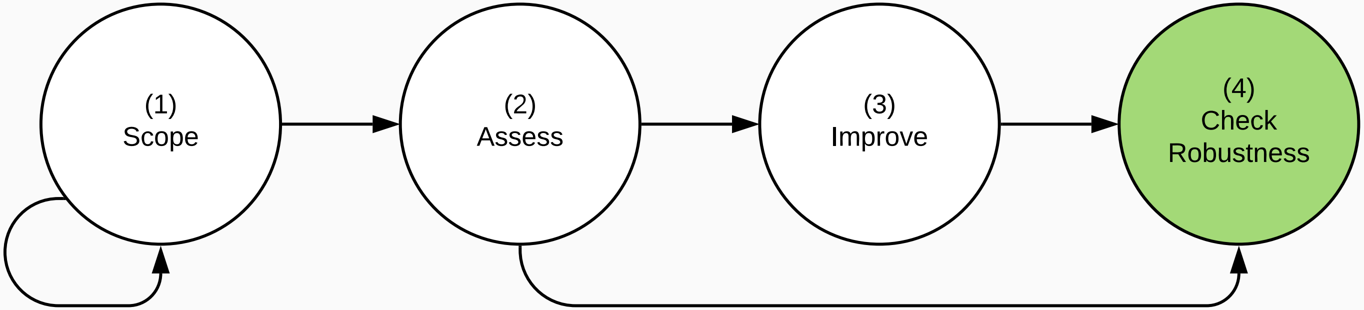
> "Provide data and codes generating the other two figures in the paper, which are not given in the replication file."

Example 3:

> "Table 3 can be reproduced identically from the [...] analytic data files. I was not able to reproduce the analytic data files due to lack of access to ArcGIS software, but the code scripts and raw data files [...] are included in the reproduction package."



Robustness Checks



Two main parts for robustness:

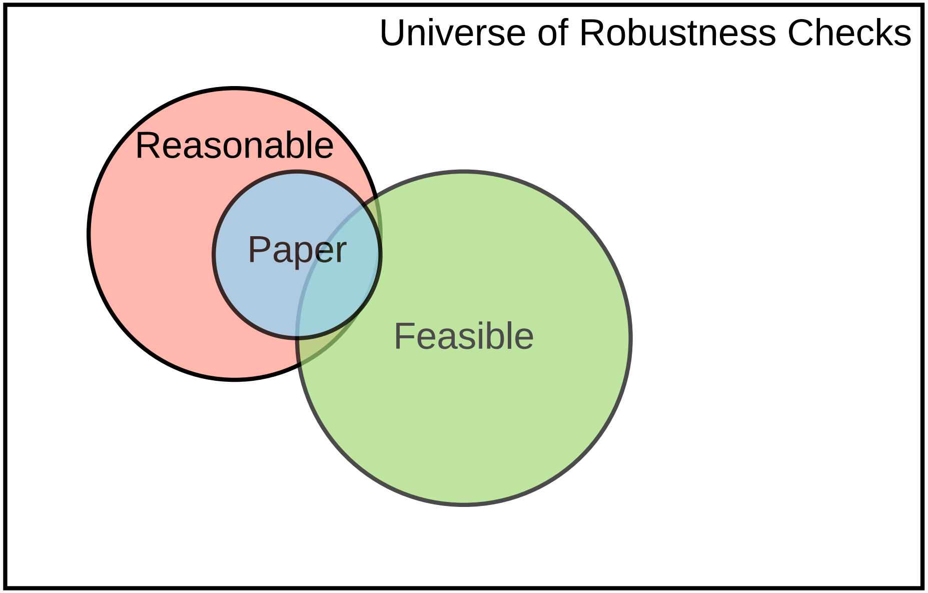
1. Increase the number of robustness checks
2. Justify the appropriateness of a specific test

Robustness

Robustness checks: any possible change in a computational choice, both in data analysis and data cleaning

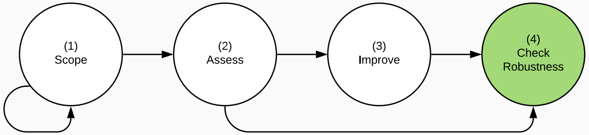
Reasonable specifications (Simonsohn et. al., 2018):

1. Sensible tests of the research question
2. Expected to be statistically valid, and
3. Not redundant with other specifications in the set.



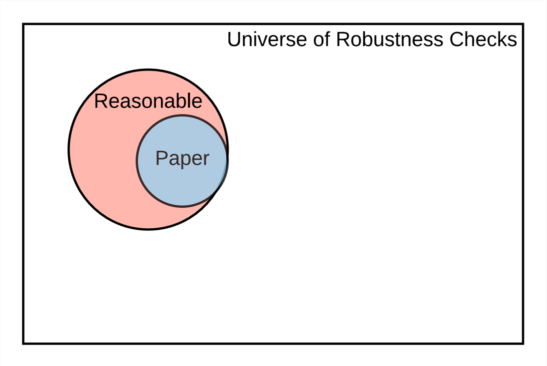
Reproducers will be able to record two types of contributions:

- Mapping the universe of robustness checks
- Proposing a specific robustness check

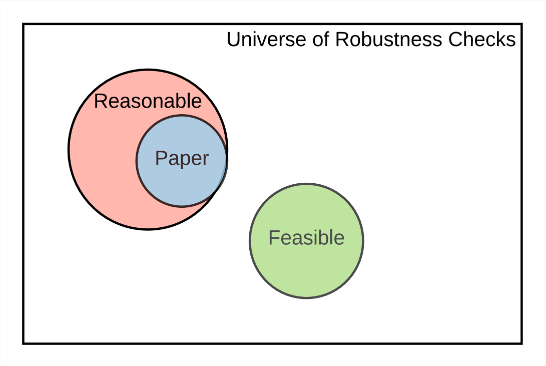


Robustness & Reproducibility

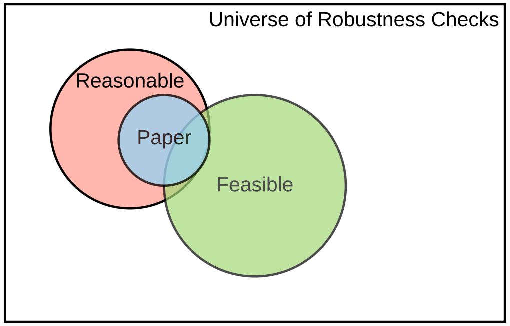
Robustness with level 1



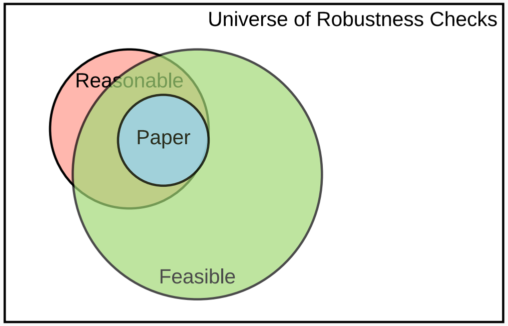
Robustness with levels 2-4

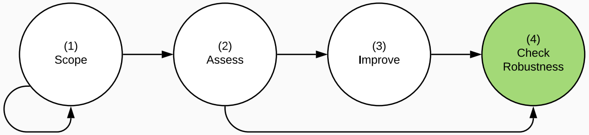


Robustness with levels 5-9



Robustness with level 10





Promoting a Constructive Exchange

1 - Contacting the original author(s) when there is no reproduction package

2 - Contacting the original author(s) to request specific missing items of a reproduction package

3 - Asking for additional guidance when some materials have been shared

4 - Response when the original author has refused to share due to *undisclosed reasons*

5 - Response when the original author has refused to share due to legal or ethical restrictions of the data

6 - Contacting the original author to share the results of your reproduction exercise

7 - Responding to hostile responses from original authors

Under development: sample responses form authors to reproducers

Example 1: There is no reproduction package

> **Subject:** Reproduction package for `[“Title of the paper”]`
>
> Dear Dr. `[Lastname of Corresponding Author]`,
>
> I am contacting you to request a reproduction package for your paper titled `[Title]` which was published in `[Reference]`. A reproduction package may contain (raw and/or analytic) data, code, and other documentation that makes it possible to reproduce paper. Would you be able to share any of these items?
>
> I am a `[position]` at `[Institution]`, and I would like to reproduce the results, tables, and other figures using the reproduction materials mentioned above. I have chosen this paper because `[add context ... ]`. **Unfortunately, I was not able to locate any of these materials on the journal website, Dataverse `[or other data and code repositories]`, or in your website**.
>
> **I will record the result of my reproduction attempt on ACRE [...]. With your permission, I will also record the materials you share with me, which would allow access for other reproducers and avoid repeated requests directed to you. Please let me know if there are any legal or ethical restrictions that apply to all or parts of the reproduction materials so that I can take that into consideration during this exercise.**
>
> In addition to your response above, would you be available to respond to future (non-repetitive) inquiries from me or other reproducers conducting an ACRE excercise? **Though your cooperation with my and/or any future request would be extremely helpful, please note that you are *not required to comply*.**
>
> Since I am required to complete this project by `[date]`, I would appreciate your response by `[deadline]`.
>
> Let me know if you have any questions. Please also feel free to contact my supervisor/instructor `[Name (email)]` for further details on this exercise. Thank you in advance for your help!
>
> Best regards,
> `[Reproducer]`

Example 1: Following up on additional materials

Template email:

> **Subject:** Clarification for reproduction materials for `[“Title of the paper”]`
>
> Dear Dr. `[Lastname of Corresponding Author]`,
>
> Thank you for sharing the materials. They have been immensely helpful for my work.
>
> Unfortunately, I ran into a few issues as I delved into the reproduction exercise, and I think your guidance would be helpful in resolving them. **`[Describe the issues and how you have tried to resolve them. Describe whatever files or parts of the data or code are missing. Refer to examples 1 and 2 below for more details]`**.
>
> Thank you in advance for your help.
>
> Best regards,
> `[Reproducer]`

An example of well described issues:

> Specifically, I am attempting to reproduce OUTPUT X (e.g., table 1, figure 3). I found that the following components are required to reproduce to reproduce OUTPUT X:

```
OUTPUT X
   └───[code] formatting_table1.R
       ├───output1_part1.txt
       |   └───[code] output_table1.do
       |       └───[data] analysis_data01.csv
       |          └───[code] data_cleaning01.R*
       |             └───[data] UNKNOWN
       └───output1_part2.txt
           └───[code] output_table2.do
               └───[data] analysis_data02.csv
                  └───[code] data_cleaning02.R
                     └───[data] admin_01raw.csv*
```

> I have marked with an asterisk (*) the items that I could not find in the reproduction materials: **data_cleaning01.R** and **admin_01raw.csv**. After accessing these files, I will also be able to identify the name of the raw data set required to obtain output1_part1.txt. This is to let you know that I may need to contact you again if I cannot find this file (labeled as **UNKNOWN** above) in the reproduction materials.
>
> I understand that this request will require some work for you or somebody in your research group, but I want to assure you that I will add these missing files to the reproduction package for your paper on the ACRE platform. **Doing this will ensure that you will not be asked twice for the same missing file.**

Ok, I get it. But what is in for me?

- Standardized homework/project: everything is set up in terms of structure and deliverables.
- Easy to grade (homework format).
- Easy to guide and oversee (undergraduate dissertation format).
- Easy to setup as an independent study.
- Reduces duplication of requests to authors.
- Facilitates a constructive exchange of ideas.
    - When emailing authors.
    - When discussion reproduction attempts.
- Personal satisfaction that you're contributing a public good to the profession!

Easy to grade: report 1

This browser does not support PDFs. Please download the PDF to view it: Download PDF.

Easy to grade: report 1

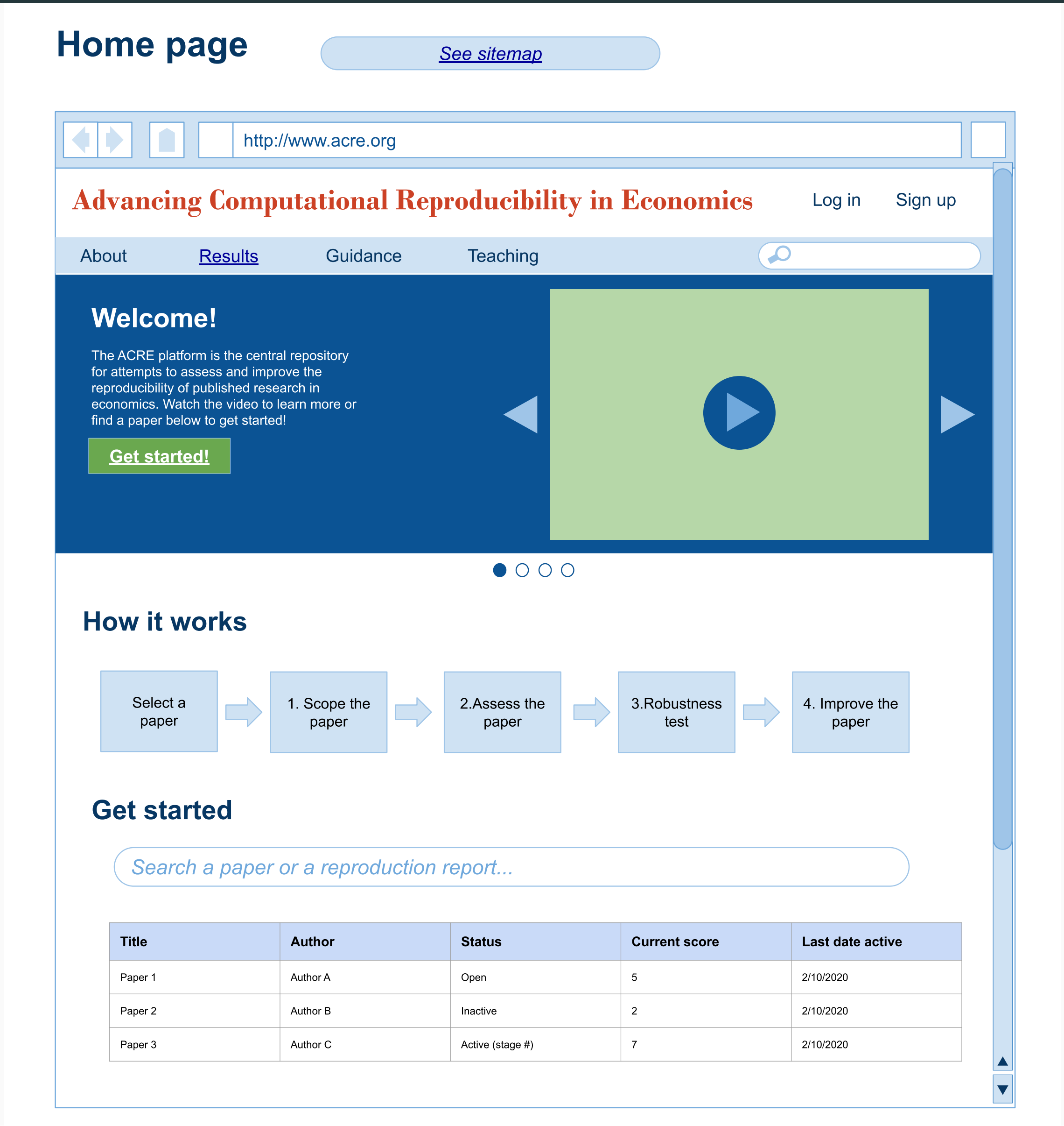
This browser does not support PDFs. Please download the PDF to view it: Download PDF.

This browser does not support PDFs. Please download the PDF to view it: Download PDF.

Table of Contents for Part I

1. BITSS
2. Reproducibility
3. ACRE Guidelines
4. **ACRE Platform**

Platform: Home page



Home page

See sitemap

Advancing Computational Reproducibility in Economics

Welcome!

The ACRE platform is the central repository for attempts to assess and improve the reproducibility of published research in economics. Watch the video to learn more or find a paper below to get started!

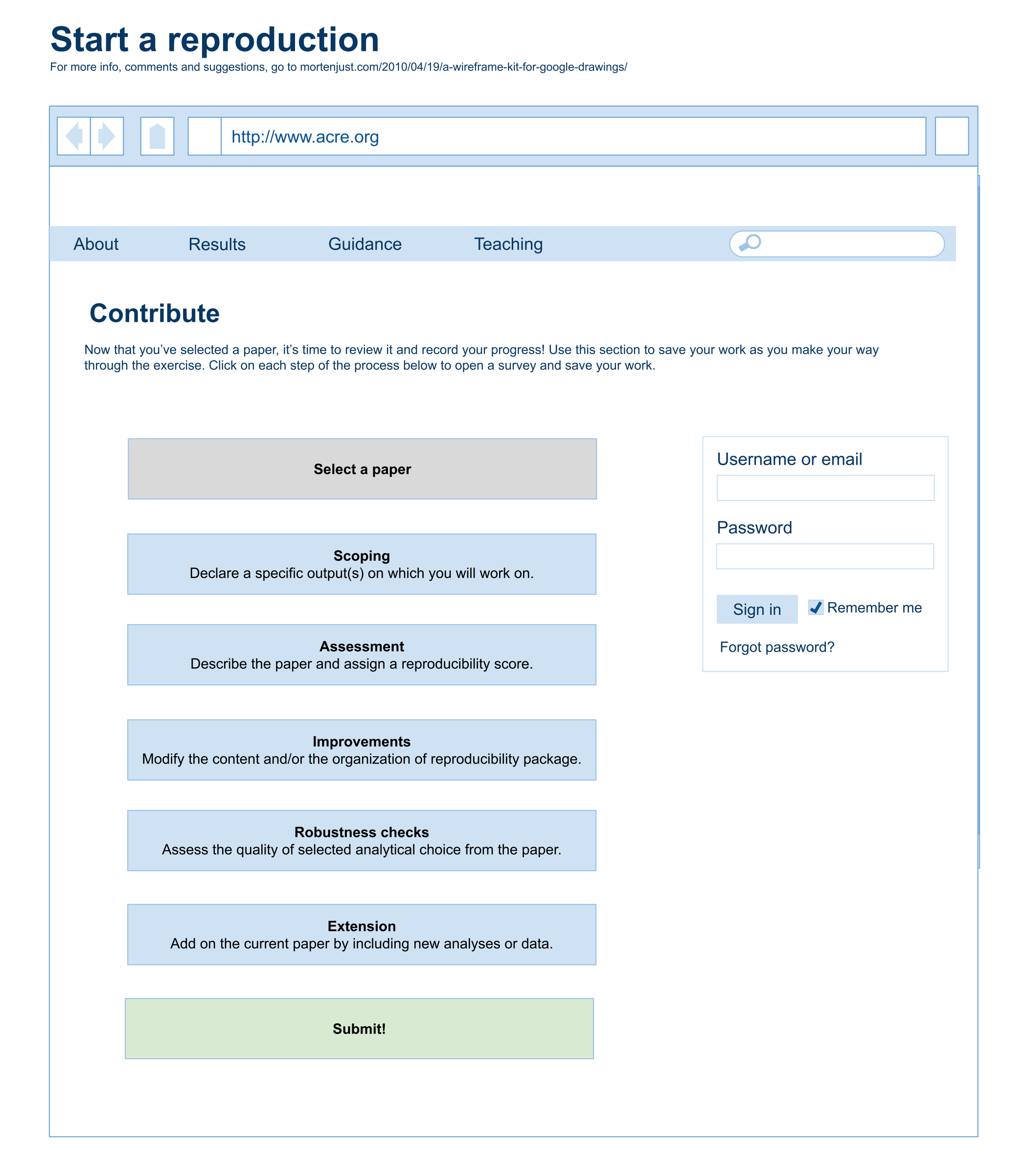
How it works

Select a paper → 1. Scope the paper → 2.Assess the paper → 3.Robustness test → 4. Improve the paper

Get started

Title	Author	Status	Current score	Last date active
Paper 1	Author A	Open	5	2/10/2020
Paper 2	Author B	Inactive	2	2/10/2020
Paper 3	Author C	Active (stage #)	7	2/10/2020

Platform: Home page



Start a reproduction

For more info, comments and suggestions, go to mortenjust.com/2010/04/19/a-wireframe-kit-for-google-drawings/

http://www.acre.org

About Results Guidance Teaching

Contribute

Now that you've selected a paper, it's time to review it and record your progress! Use this section to save your work as you make your way through the exercise. Click on each step of the process below to open a survey and save your work.

Select a paper

Scoping
Declare a specific output(s) on which you will work on.

Assessment
Describe the paper and assign a reproducibility score.

Improvements
Modify the content and/or the organization of reproducibility package.

Robustness checks
Assess the quality of selected analytical choice from the paper.

Extension
Add on the current paper by including new analyses or data.

Submit!

Username or email

Password

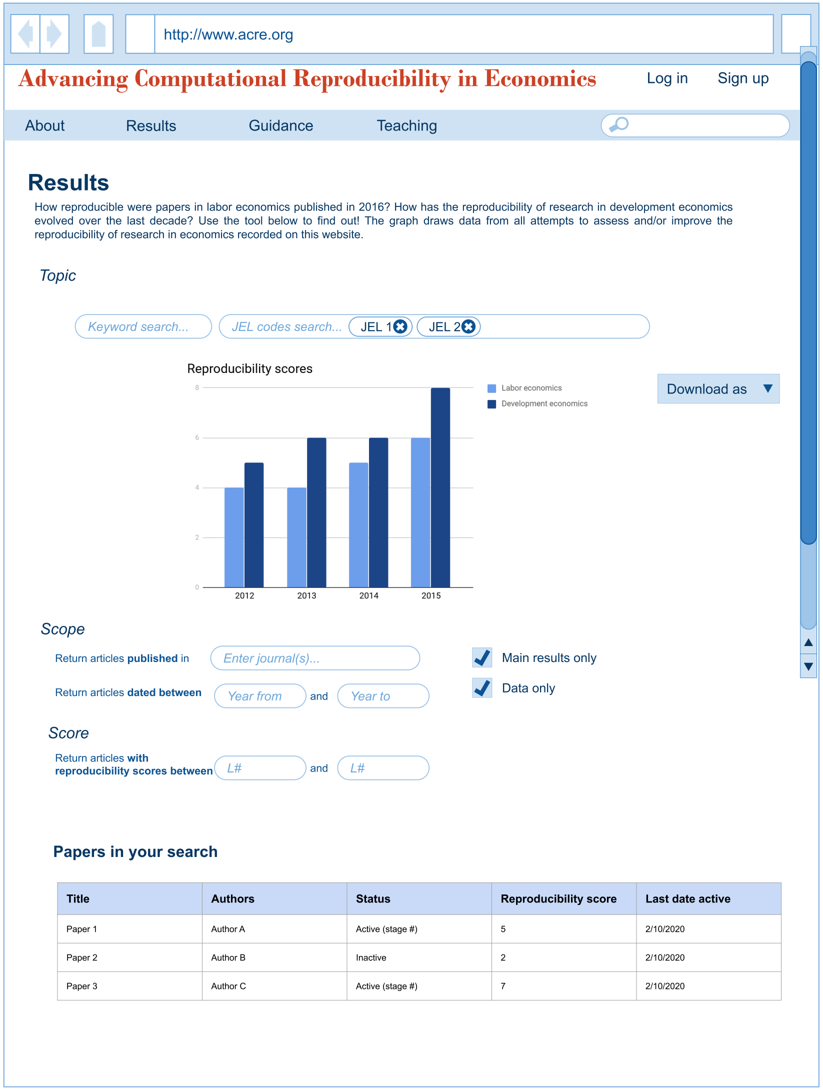
Sign in ☑ Remember me

Forgot password?

Platform: Home page

Home page

For more info, comments and suggestions, go to mortenjust.com/2010/04/19/a-wireframe-kit-for-google-drawings/



http://www.acre.org

Advancing Computational Reproducibility in Economics

Log in Sign up

About Results Guidance Teaching

Results

How reproducible were papers in labor economics published in 2016? How has the reproducibility of research in development economics evolved over the last decade? Use the tool below to find out! The graph draws data from all attempts to assess and/or improve the reproducibility of research in economics recorded on this website.

Topic

Keyword search... *JEL codes search...* JEL 1⊗ JEL 2⊗

Reproducibility scores

Download as ▼

Scope

Return articles **published** in *Enter journal(s)...*

☑ Main results only

Return articles **dated between** *Year from* and *Year to*

☑ Data only

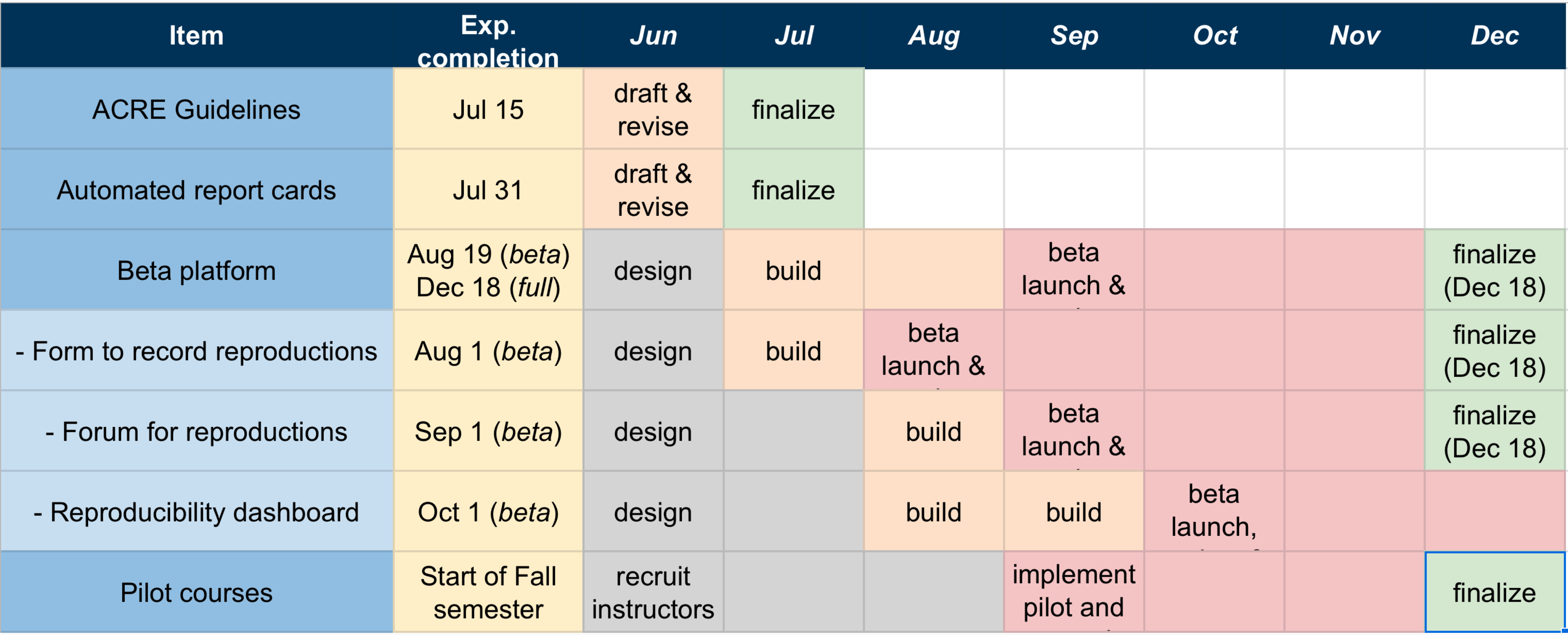
Score

Return articles **with reproducibility scores between** *L#* and *L#*

Papers in your search

Title	Authors	Status	Reproducibility score	Last date active
Paper 1	Author A	Active (stage #)	5	2/10/2020
Paper 2	Author B	Inactive	2	2/10/2020
Paper 3	Author C	Active (stage #)	7	2/10/2020

Timeline



Item	Exp. completion	*Jun*	*Jul*	*Aug*	*Sep*	*Oct*	*Nov*	*Dec*
ACRE Guidelines	Jul 15	draft & revise	finalize					
Automated report cards	Jul 31	draft & revise	finalize					
Beta platform	Aug 19 (*beta*) Dec 18 (*full*)	design	build		beta launch &			finalize (Dec 18)
- Form to record reproductions	Aug 1 (*beta*)	design	build	beta launch &				finalize (Dec 18)
- Forum for reproductions	Sep 1 (*beta*)	design		build	beta launch &			finalize (Dec 18)
- Reproducibility dashboard	Oct 1 (*beta*)	design		build	build	beta launch,		
Pilot courses	Start of Fall semester	recruit instructors			implement pilot and			finalize

Ok, I Am Interested. What's Next?

You can check the ACRE Guidelines, and also contribute if you want: https://bitss.github.io/ACRE/

Guidelines for Verification of Computational Reproducibility in Economics

A beta version of the platform will be online by early September.

Sign up here

Acknowledgements

Arnold Ventures

Everybody who has participated in the pilots so far:

Ted Miguel's Graduate Development Economic Course (2019, 2020) - UC Berkeley

Dina Pomeranz undergraduate thesis for Marc Richter - University of Zurich

Slides template: Grant McDermott.

10' Break

Target Audience for Part II:

- Interested about version control software but have never used git or github.
- Tried to used it a couple of times but was heavily discouraged by its obscurtity.

Please make sure the following software is installed in your computer:

- Atom
- Github Desktop App
- Optional: RStudio and R (no, we will not be using R)

Also, please create and account in github.com, and confirm that account in your email. Write down the password you creat for github.com.

Part II: Hands-on Tutorial

Table of Contents of Part II

1. A Short Tutorial on Git/Github
2. Other Resources for Reproducibility

Git/Github for Version Control

- Git and Github are tools to track the complete history of your files.
- They are very popular among programmers, but not so much among non-programmers.
- Why? I believe it has to do with GUIs.

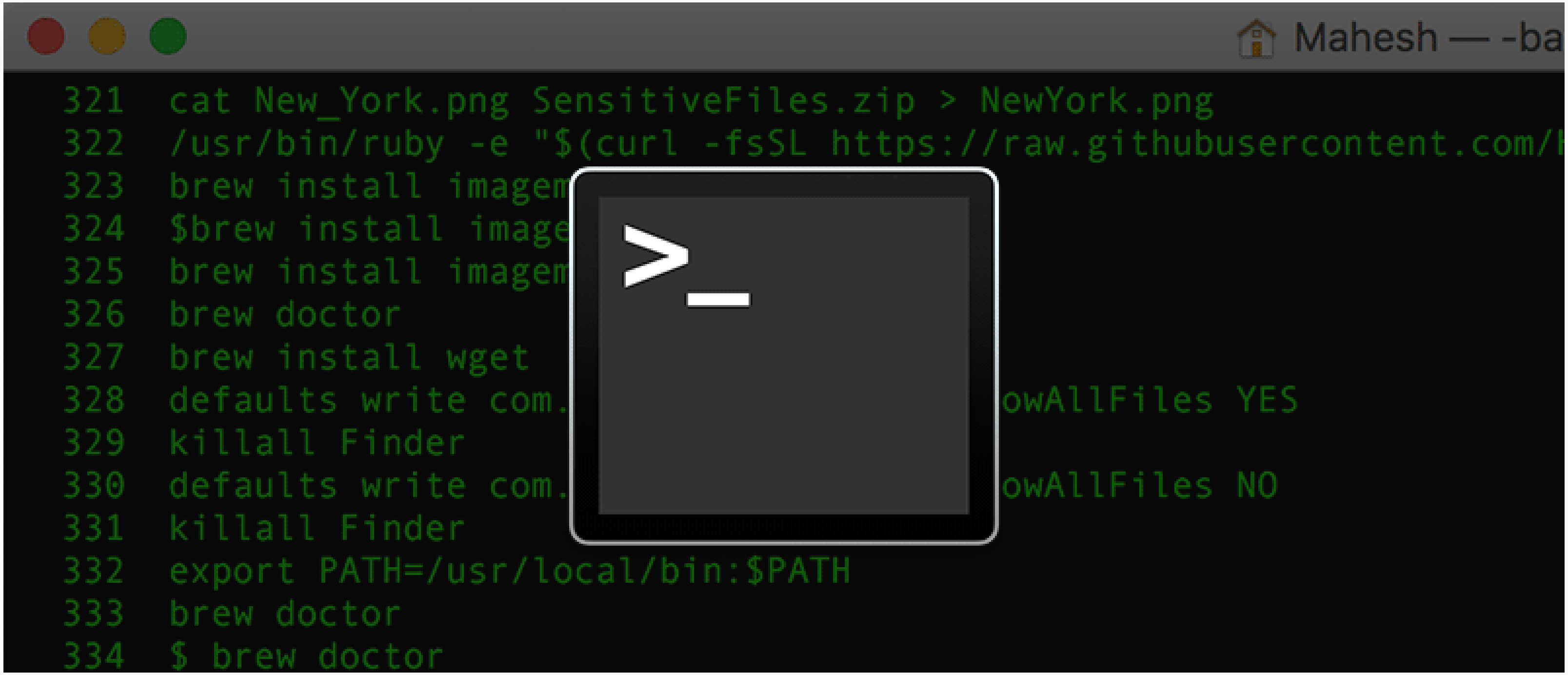
What is a GUI and why the bad

Graphical **U**ser **I**nterface

- For most of us (non-programmers): *GUI = Software*.
- GUIs are behind the popularization of personal computers.
- Unfortunately GUIs are pretty bad at keeping a record of actions taken (bad for reproducibility).

What is not a GUI?

- Any software that is run in the command line (aka terminal, shell, bash, etc).



```
321  cat New_York.png SensitiveFiles.zip > NewYork.png
322  /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/
323  brew install imagem
324  $brew install image
325  brew install imagem
326  brew doctor
327  brew install wget
328  defaults write com.                        owAllFiles YES
329  killall Finder
330  defaults write com.                        owAllFiles NO
331  killall Finder
332  export PATH=/usr/local/bin:$PATH
333  brew doctor
334  $ brew doctor
```

- Git was designed to run in the command line.
- Today we will learn Git **without** the command line.

What is Git 1/2

- Git is a software designed to track the **entire** history of the code of a project.
- Designed originally for software development, it has gained important traction in the research community.
- Main appeal: facilitates full reproducibility and collaboration.
- Git is mainly meant to work as a non-GUI (in the command line) software.
  However: most of the key features can be used through a GUI.

What is Git 2/2

- By code git understands any type of plain text file (`myfile.R`, `myfile.do`, `.tex/.md/.txt/.csv/.etc`).
- This type of file can be understood as "human readable" as machine and human see the same fie.
- Files that are "non-human readable" are called binary files (`myfile.docx`, `myfile.xlxs`, `.pdf/.exe/.dta/.etc`).
- Git can also detect changes in binary files, but it cannot show those changes.

What is Github

- Github is a company that provides two services (that we care of):
  - A web hosting service for all our files track with git (public free/private $ or free if academic).
  - A GUI software (Desktop App) that provides user friendly access to git.
- Others hosting ss include: Bitbucket, GitLab, Gitkraken, etc.
- Other GUIs include: SourceTree, Gitkraken, Atom, RStudio.

The Primary Goal of Version Control (for

The Goal: keep track of any potentially meaningful modification to your code.

Secondary Goal: learn how to collaborate with others using Github.

Bonus track: get you excited about using open source statistical software (R, Python, Julia, etc)

Strategy 1:

1 - Agree on a naming convention with your co-authors (eg: YYYYMMDDfilename_INITALS).

2 - Begin working from the last saved version (eg: `20180325demo_FH.do`).

3 - At the end of the day, save on a new version (eg: `20180327demo_FH.do`).

Pros: Easy adoption.

Cons: Error prone, hard to document, lots of files for each document.

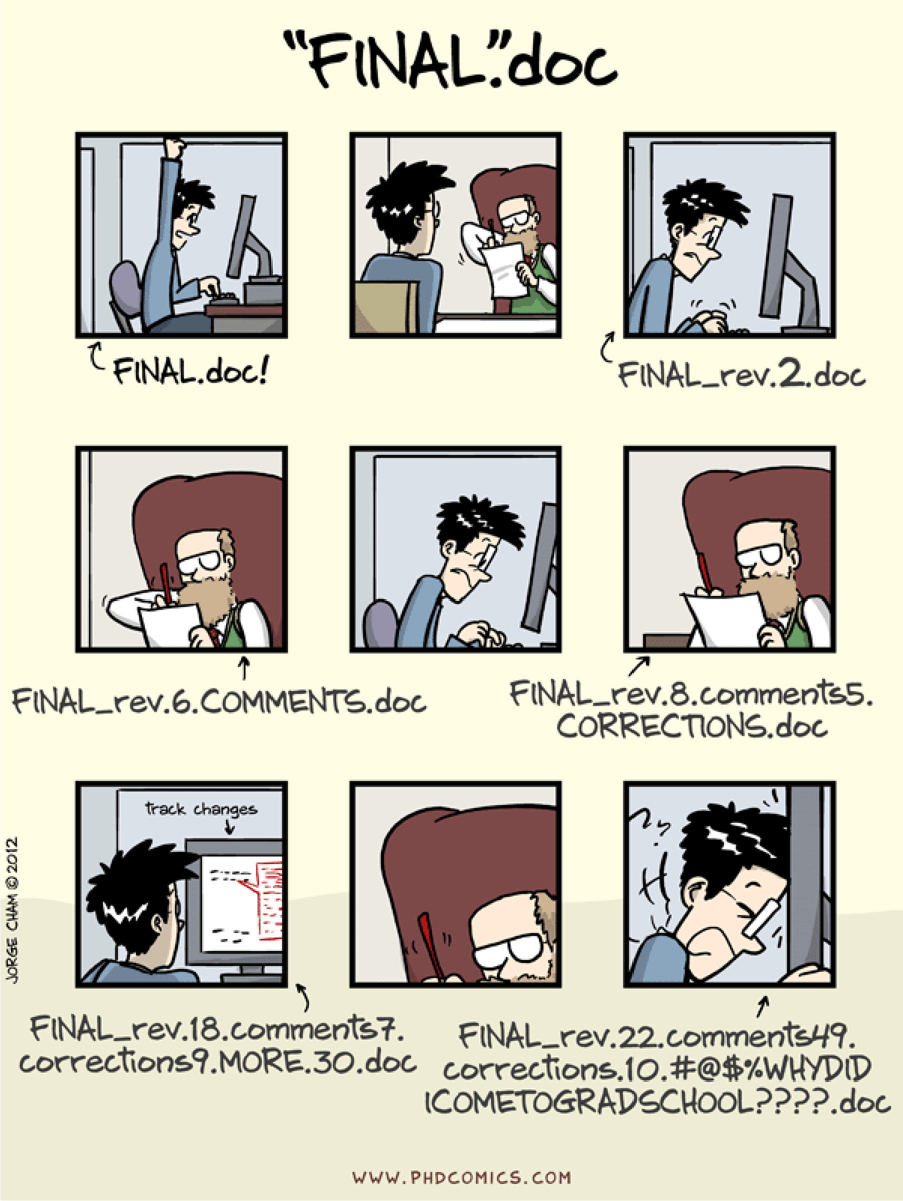
Strategy 2:

1 - Name your file `filename` (ideally `01_filename`)
2 - Take a snapshot of your work every time you complete relevant change (day, hour or minutes).
3 - Update your entire working folder to the cloud.

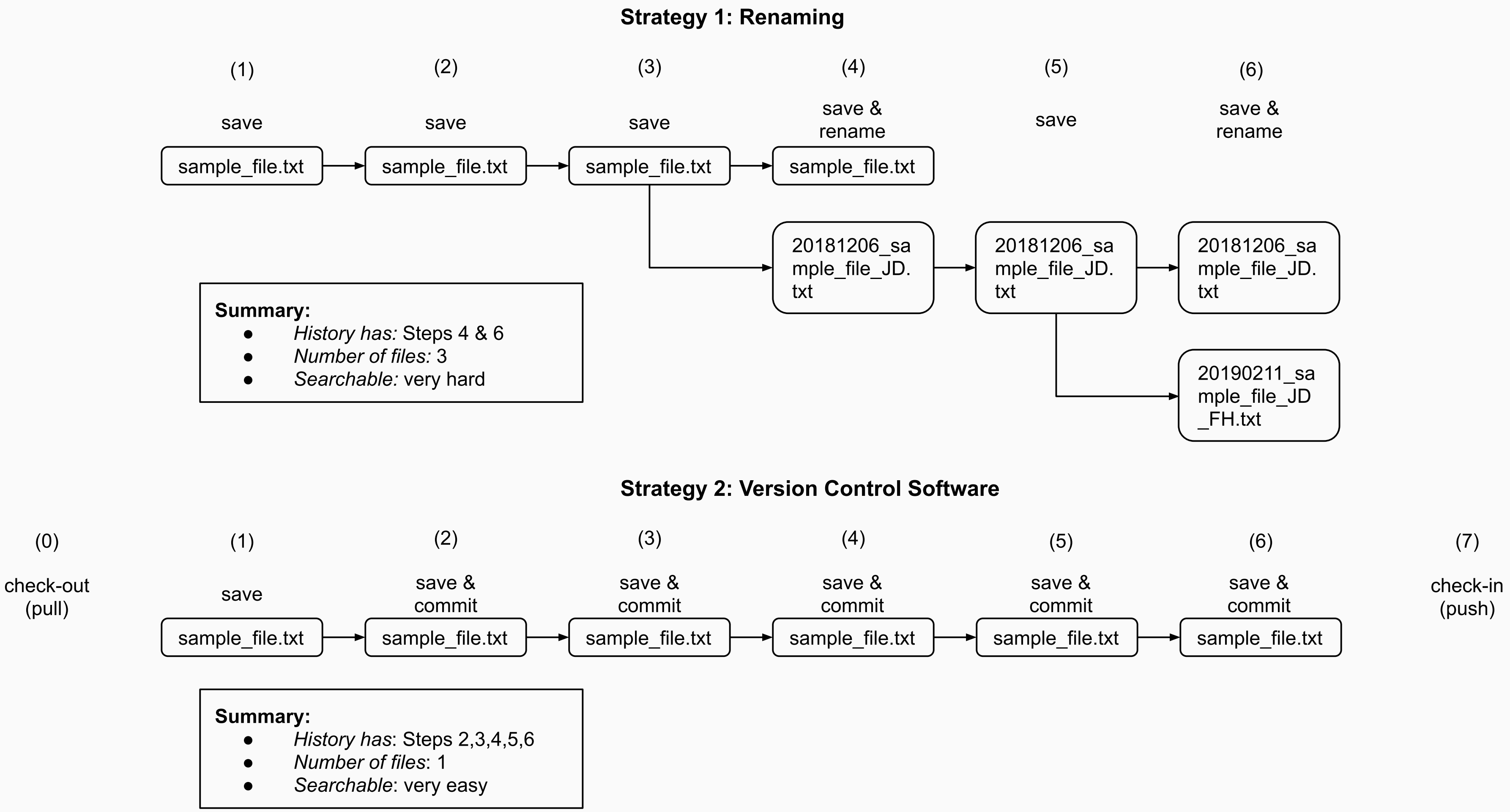
Pros: Error proof, seamless documentation, one file per document, track differences across all versions, meant to work with the cloud.

Cons: Harder adoption.

We want to avoid this situation:



Comparison of Workflows



Strategy 1: Renaming

Summary:
- *History has:* Steps 4 & 6
- *Number of files:* 3
- *Searchable:* very hard

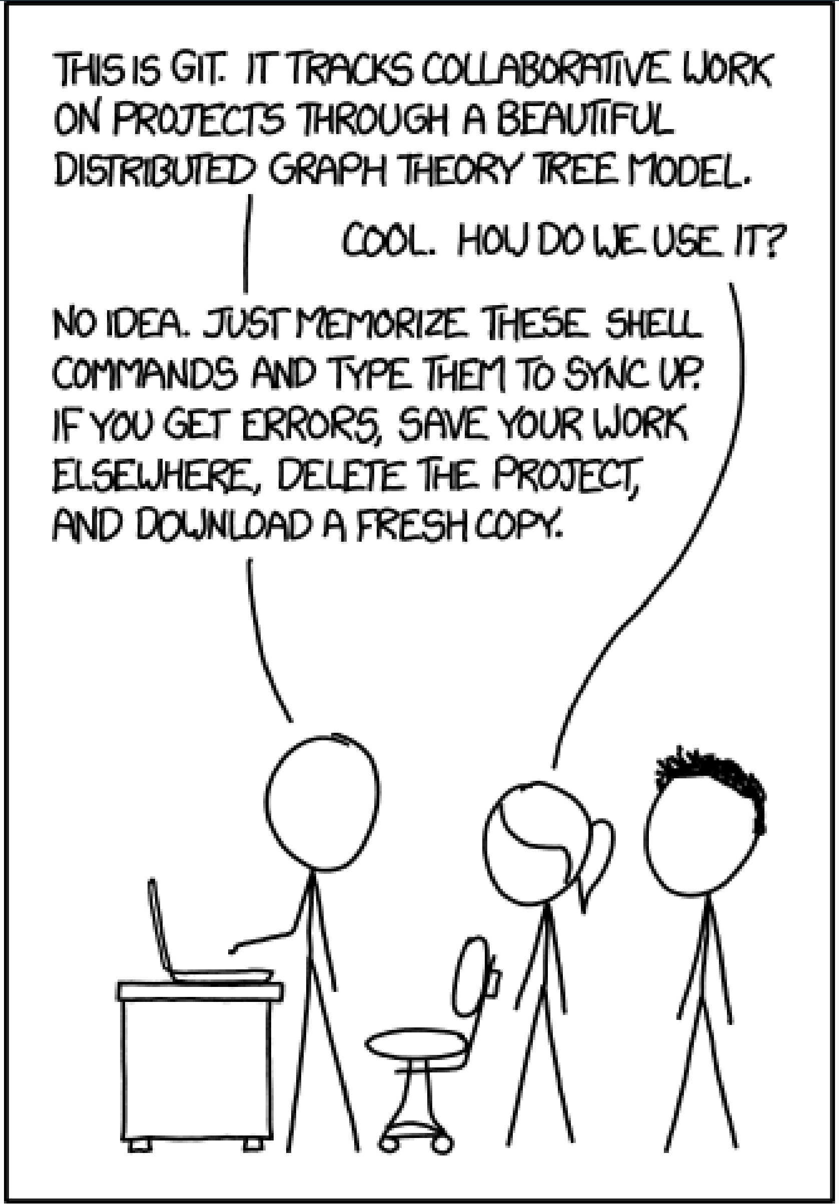
Strategy 2: Version Control Software

Summary:
- *History has*: Steps 2,3,4,5,6
- *Number of files*: 1
- *Searchable*: very easy

Other reasons to use git

- To access a whole new world of knowledge!
- Great tool for collaboration.
- Easier to test all sorts of ideas/models.

Managing expectations


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Demos

Five Demos:

1 - **Simple but instructive.**

2 - Repeat with branches.

3 - Repeat with collaboration: pull requests.

4 - Repeat with collaboration: shared ownership.

5 - Explore a real life repo.

Demo #1: We Start in the Cloud

1 - Create github.com account and sign in.

2 - Let's look at some **repos**.

3 - First way to access content: download.

4 - What if you want to have your own copy of the repo? **Fork** it!

5 - Now create your own repo. Initiate readme and make some edits.

Demo #1: We move to our local

6 - Clone the repo. Explore the files and location.

7 - Create new files, edit. And commit. Edit again, and commit again.

8 - Push. Edit on github.com, and pull.

9 - For this tutorial, best way to access previous version: explore in github.com and download.

Five Demos 2/5:

1 - Simple but instructive.
Review: def repo, github.com, download, clone, destination folder, fork, create repo, commit, push, pull, delete, search repo, download old version.

2 - **Repeat with branches.**

3 - Repeat with collaboration: pull requests.

4 - Repeat with collaboration: shared ownership.

5 - Explore a real life repo.

Demo #2: Branches and collaboration

1 - Create a branch from previous repo.

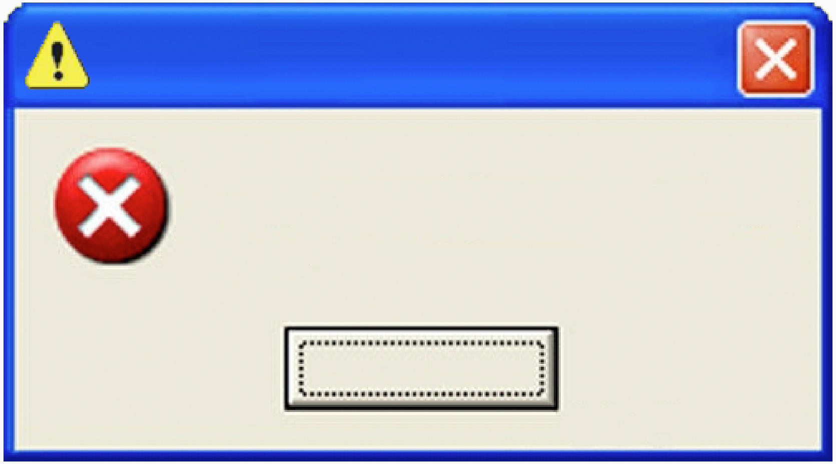
2 - Add new content (do not replace), commit a few times, and go back and forth to the main branch.

3 - Go back to main branch (master), observe file, merge.

4 - Look at the history of the main branch.

5 - Repeat 1-3 but now replace instead of adding content.

Fatal Error!



Burn it and start with a fresh copy!

Five Demos: 3/5

1 - Simple but instructive.

Review: def repo, github.com, download, clone, destination folder, fork, create repo, commit, push, pull, delete, search repo, download old version.

2 - Repeat with branches.

Review: All of the above, plus: branch, merge, resolve conflicts.

3 - **Repeat with collaboration: pull requests.**

4 - Repeat with collaboration: shared ownership.

5 - Explore a real life repo.

Demo #3: Pull requests

1 - Fork repo github.com/BITSS/test_birthday, and clone it into your machine.

2 - Edit fields of name, and birth date.

3 - Save, commit and push.

4 - Create your first **pull request**.

5 - Let's see if I can manage all those pull requests very quickly (maybe illustrate issues).

6 - Now find your neighbors repo of Demos 1 & 2, fork it, clone it, make a change, save, commit, and...

Two formats of collaboration

- One owner, many pull requests.
  - Easier to control, requires constant updating.
- Many owners, all can push.
  - **Very** important to pull at the beginning and at before each push.

Five Demos: 4/5

1 - Simple but instructive.

Review: def repo, github.com, download, clone, destination folder, fork, create repo, commit, push, pull, delete, search repo, download old version.

2 - Repeat with branches.

Review: All of the above, plus: branch, merge, resolve conflicts.

3 - Repeat with collaboration: pull requests.

Review: collaborate via fork + PR

4 - **Repeat with collaboration: shared ownership.**

5 - Explore a real life repo.

Demo #4: Many owners

1. Half of you (#1): go back to the repo of demo 1 & 2 and invite a collaborator. (Suggestion: the "forker" creates the repo, the "forkee" is invited , edit, commit, push/pull)
2. The other half (#2): clones, commits and pushes.
3. #1 commits and pushes in **different lines**.
4. Switch and repeat 2 & 3: #2 commits first and pushes, then #1.
5. Repeat 2 - 4 but now both of you in the same lines.
6. Repeat now but with branches (optional).

Five Demos: 4/5

1 - Simple but instructive.

Review: def repo, github.com, download, clone, destination folder, fork, create repo, commit, push, pull, delete, search repo, download old version.

2 - Repeat with branches.

Review: All of the above, plus: branch, merge, resolve conflicts.

3 - Repeat with collaboration: pull requests.

Review: collaborate via fork + PR

4 - Repeat with collaboration: shared ownership.

Review: collaborate via share ownership.

5 - **Explore a real life repo.**

Demo #5: Look inside a real-life project

1- Find the following repo: `github.com/BITSS/opa-wealthtax`.

2- Fork it and clone it.

3- Open it in your computer: `opa-wealthtax.Rproj` (needs RStudio), look around and execute `code/dynamic_doc/wealth_tax_dd.Rmd`.

4- Find elasticities, fill in csv, document, submit. 5 - Find `code/interactive_visualization/server.R` and in line `1561` change `red` to `blue`

Five Demos: 5/5

1 - Simple but instructive.

Review: def repo, github.com, download, clone, destination folder, fork, create repo, commit, push, pull, delete, search repo, download old version.

2 - Repeat with branches.

Review: All of the above, plus: branch, merge, resolve conflicts.

3 - Repeat with collaboration: pull requests.

Review: collaborate via fork + PR

4 - Repeat with collaboration: shared ownership.

Review: collaborate via share ownership.

5 - Explore a real life repo.

Review: All of the above, plus: how does a real-life example looks like.

Now go and explore!

Some good habits:

- Commit often (<1hr)
- Always pull before you start a new session of work. Also good to pull before pushing.
- Think of your remote as the most important set of files. Get used to deleting things in your local machine.

Want to Learn More: Version Control

Tutorials

- Great 20 min intro to Git by Alice Bartlett
- Great 2hr tutorial to Github by Jenny Bryan (git ninja)
- Software Carpentry's step-by-step tutorial (command line).

Documentation

- Jenny Bryan's Happy Git
- Documentation from Matthew Gentzkow and Jesse Shapiro
- Karthik Ram's paper on Git for Research

Economists Doing Highly Reproducible Work[1]

People

- Nick Huntingon
- Shoshana Vasserman
- Lars Vilhuber
- Grant McDermott
- Tyler Ransom
- Ed Rubin
- Luiza Andrade
- Max Kasy
- Matt Jensen
- Richard Evans
- John Horton
- Cora Kingdon
- Alvaro Carril
- Andrew Heiss
- Lisa Rennels
- Michael Stepner
- Lachlan Deer
- Rebekah Din

Organizations

- LOST
- Opportunity Lab
- Congressional Budget Office
- Policy Simulation Library
- Gentzkow & Shapiro Lab
- Urban Institute

[1]: Non-exhaustive list of people and organizations doing amazing reproducible work on github (other than us!)