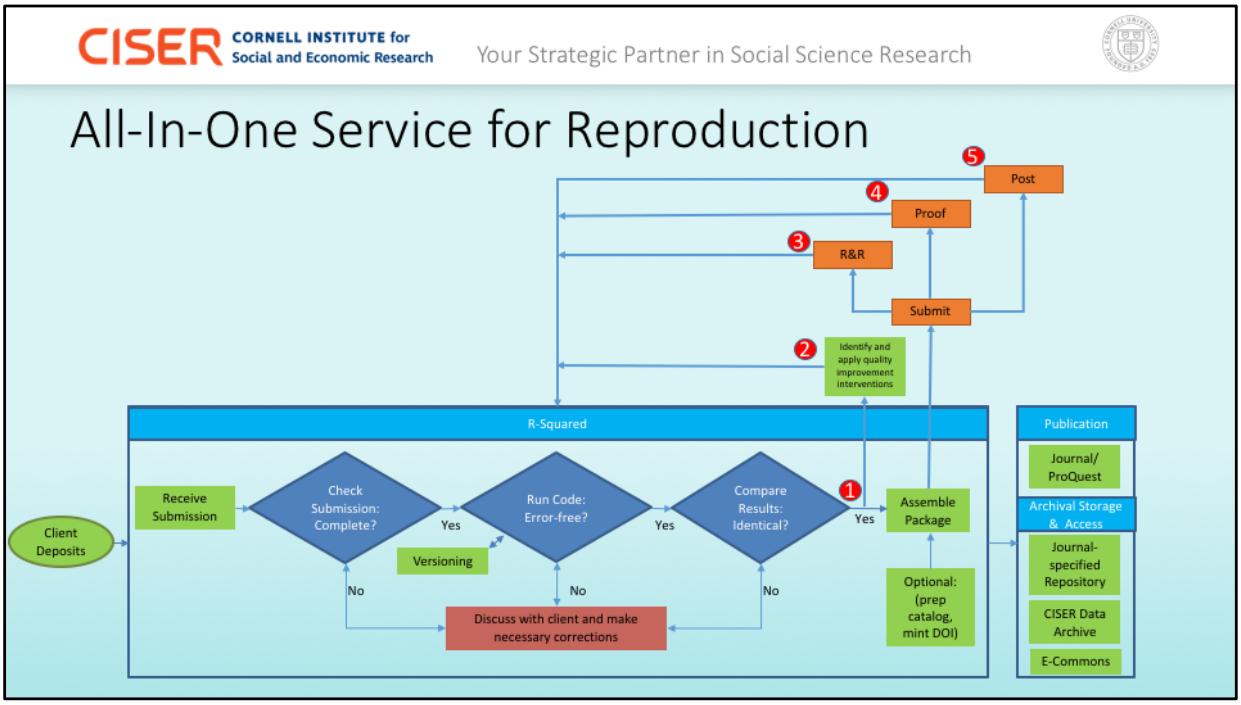
CISER CORNELL INSTITUTE for Social and Economic Research

Your Strategic Partner in Social Science Research

Working With Clients

Results Reproduction:
Computational reproduction of research results to ensure Reproducibility and Transparency.

Here is an outline of what I will speak about today. Starting out I will share the current status of our workflow and discuss some of the elements of it. This will include a brief discussion of the various elements of the workflow. Then I will discuss differences between projects and the inherent difficulty that brings with it. From there I will move to the most common errors that we see, first discussing the most common computer script errors and then the most common manuscript errors. After that, I will discuss good practices to avoid some of these common errors and solutions to them and finally I will discuss our future plans for the workflow.



- This is our current workflow and I will quickly walk you through it. First we receive the submission and check it for completeness. Then we run it and begin to version the code and submission. Once we manage to get it to run all the way through we compare the results with the manuscript. If we find any inconsistencies or differences, we will discuss with the client and make the necessary corrections. During this process we will also look at the code and suggest improvements for either curation or for future usability such as better commenting. Finally we assemble the final package and check that against the manuscript once again to ensure that any changes were made didn't affect the results. As you can see from the diagram, we make multiple checks if our clients want us to, either at Revise and Resubmit, after they have a Proof, and then even post publication to ensure everything lines up.

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

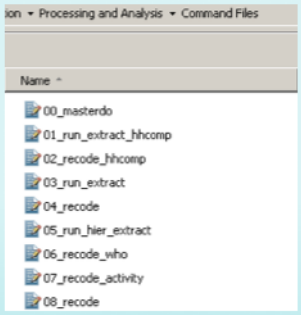
Step 1. Check the submission

Create a project folder following this convention:

R2-<year>-<lastname>-[#]

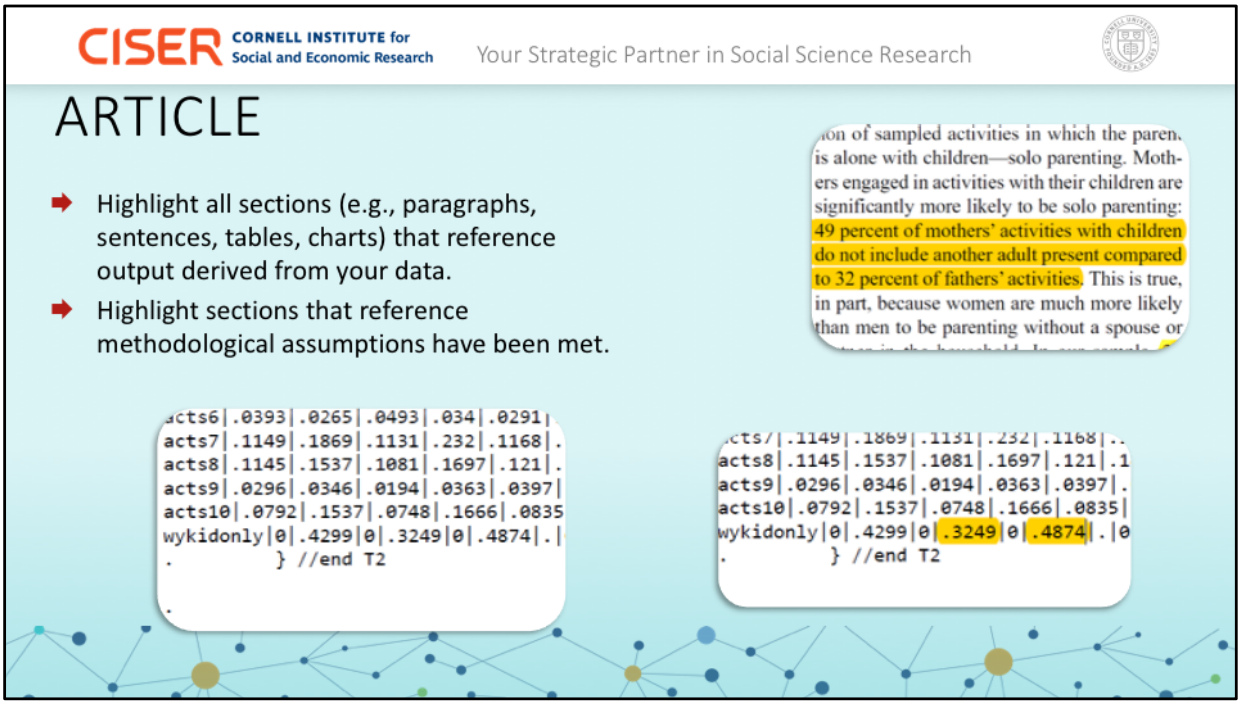
(eg. R2-2018-KIM-1)

1. Presence of a Readme (often non existant)
2. File names of code contain the order of execution, i.e., prefixed with a step ##.
3. Presence of master program file – often non-existent
4. Identify software used and use base version



First we check the files submitted, with regards to the code:
To do this we create a project folder with the naming convention of "R2-Year-Lastname-Number"

1. We check presence of a readme that would provide instructions on the requirements to reproduce the results. Often there is no readme file. This is most important once the project is finished or in final stages but getting one started early isn't bad.
2. We check the program file names that they indicate the order of execution and/or what it does, so we can easily identify which code to run first or its function.
3. We check for the presence of master program file that executes all the program files in sequence – to simplify the experience of reproducing results
4. We identify the software used by the researchers – once identified we use a freshly installed version of the software, just the base, so we can detect all the packages/modules/libraries/ados that installed by the researchers that were not part of the base package. The code will break and produce an error if the command or library being called by the code is not part of the base.



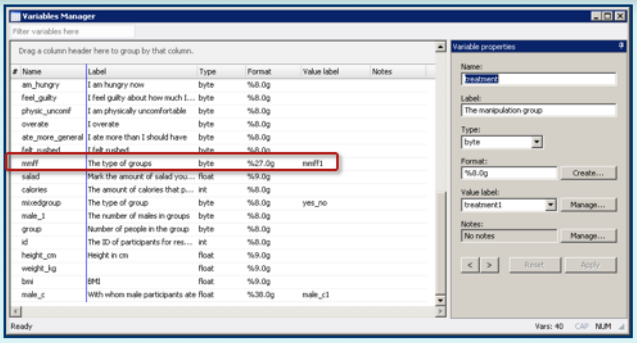
To do any further checks (or even to check the code itself) we require something to check it against which is where the article comes in. We ask the researcher to highlight the sections of their article that contain data to be checked, to insure we don't miss any in line numbers. To that end, we also ask that they print out an output file and highlight where the numbers came from. This serves as a sort of check in itself and helps to clarify which numbers we should be looking at, though authors are not great about doing this for us.

CISER CORNELL INSTITUTE for Social and Economic Research

Your Strategic Partner in Social Science Research

DATA

- All variables and values labeled
- Free of errors and inconsistencies



When we check the data itself to ensure that it is free of errors and inconsistencies if possible and that variable values are well labeled and named.

Step 2 – Inspect the contents of code

1. Change absolute paths to relative paths – relative to the location of the code
2. Run the code
3. Errors?
   - Correct the error in order to proceed (and not waste time waiting for authors)
   - If unclear how to correct and can't proceed, communicate with researchers

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

CODE

- Add comments that map sections of code to results in the manuscript. Make sure commands that generate results are preceded by comments that indicate which result the command generates. For example:

```
*The following command generates column 1 of Table 1
```

```
*The following command generates the mean age mentioned on page 3, paragraph 3
```

When we check the code we make sure that it is prefixed with a number to en

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

```
************************************************************
**********************CODE TITLE***************************
************************************************************
clear
log using "..\Documents\demoCodeLog.smcl", replace text
import delimited "..\Original Data\HelloWorld.txt"

//Labeling the variables and values
label variable treatment "The manipulation group"
label define treatment1 1 "Well Documented" 2 "Poorly
Documented"
label value treatment treatment1
label variable pieces "Good labels are descriptive but not
too long"
label variable gender "Gender"
label define gender1 1 "Male" 2 "Female"

******** Table 1 - Descriptive statistics of the sample
tab mmff
ttest age if mmff ==1 | mmff == 2, by(mmff) unequal
ttest age if mmff ==3 | mmff == 4, by(mmff) unequal
```

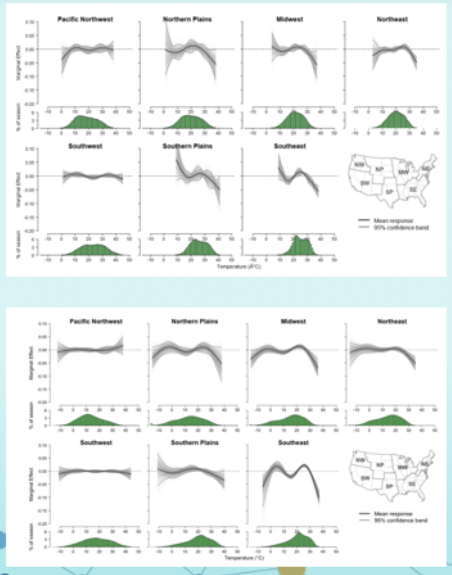
- New variables and values are labeled
- Codes are commented to describe processes and mapped to paper sections
- Code produces outputs in the same order as they appear in the manuscript
- Remove or comment out unnecessary code

CISER CORNELL INSTITUTE for Social and Economic Research

Your Strategic Partner in Social Science Research

Lack of Seed

Any random generation requires a seed or else it can have different results and won't be reproduced. Example on the top is without the seed and bottom is with it.

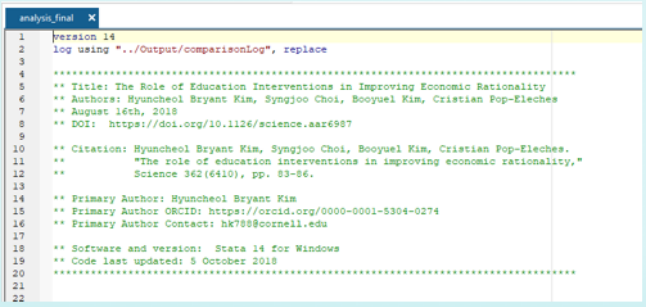


Lack of a seed is a serious problem in reproduction, and is a tricky thing to handle. Findings should, in theory, be robust enough to be able to withstand using a random seed and get the same results, without the specific seed it is impossible to reproduce the exact findings. The example shown displays how findings can change between two seeds even though there were thousands of generated numbers and the findings were robust.

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

Insert header/metadata

1. Title of study
2. Author(s)
3. Article DOI
4. Article Citation
5. Primary author contact info and ORCID
6. Date of last code update
7. Code citation
8. License



** Code Citation (i.e., how to cite this code):

<author(s) names> (<date>) <title of program/source code> (<code version>) [<type>]. Web address or publisher.

** License: CC-BY

Usually there is no header in the code.
We put a header.

It contains:
1. Software and version used
2. Command to save the output file. Reusers can compare this output file to the one we include in the package so they know they'v executed the code correctly.
3. The title of the study
4. The DOI which we mint
5. Authors, their contact info, and ORCID number so the authors can be contacted should re-users have questions.

The path is anchored on the location of the code, i.e. we make the location of the code as the active working directory, so that reproduction materials package is portable and installable in any folder location without having to recreate or mimic the directory path of the authors.

The code should save the output file as these results are the once the researchers will

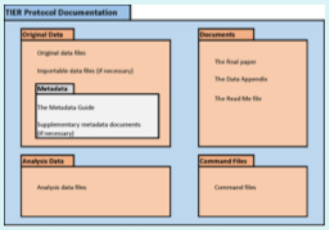
compare to the manuscript.

ORCID is important so that in case the author moves to a different institution, researchers will still be able to get in touch with them, assuming they update their ORCID profile.

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

Step 3. Organize and Package the Reproduction Materials

1. If files are unorganized within the project folder, we use **TIER Protocol** (projectier.org).
2. If files are organized within project folder, retain researcher's existing structure
3. Add a **Prerequisites folder** to store the version of the libraries, packages, ados that researchers used to generate their results



Example of our expanded TIER structure

We use a modified version of the TIER protocol to organize our materials, but really any sort of organization is acceptable, as long as there is a structure and it is organized.

Multiple checks are better, as we are all only human and make mistakes. Multiple checks by multiple people are better still!

Now onto the difficulties associated with projects. First are the differences between projects which include the level of work that needs to be done, level of complexity, the timeline requirements, the skill and experience of the client as well as their level of comfort with reproduction tools.

Different levels of work need to be done

- The more work needed the harder each part is to do
- Finding all the various scripts that were used and removing unneeded parts
- Creating Master Files
- Re-ordering code to match order of output

Levels of Complexity

- Number of different programing languages
- Length of execute time
  - Can lead to lots of dead time and add complexity
- Amount of outputs to be checked

CISER CORNELL INSTITUTE for Social and Economic Research Your Strategic Partner in Social Science Research

Timeline requirements

- Still planning out the paper/data not collected
- Pre-Prints
- Not submitted yet
- Revise and Resubmit
- Already Published

Differences in Programming Experience/Skill

- Some disciplines focus more on coding skills, while some don't place as much of an emphasis on it
- Something that may be obvious to someone in one discipline may not be inherently obvious to everyone, most prevalently in unneeded acronyms that can be confusing.

Levels of Comfort With Reproducibility Tools

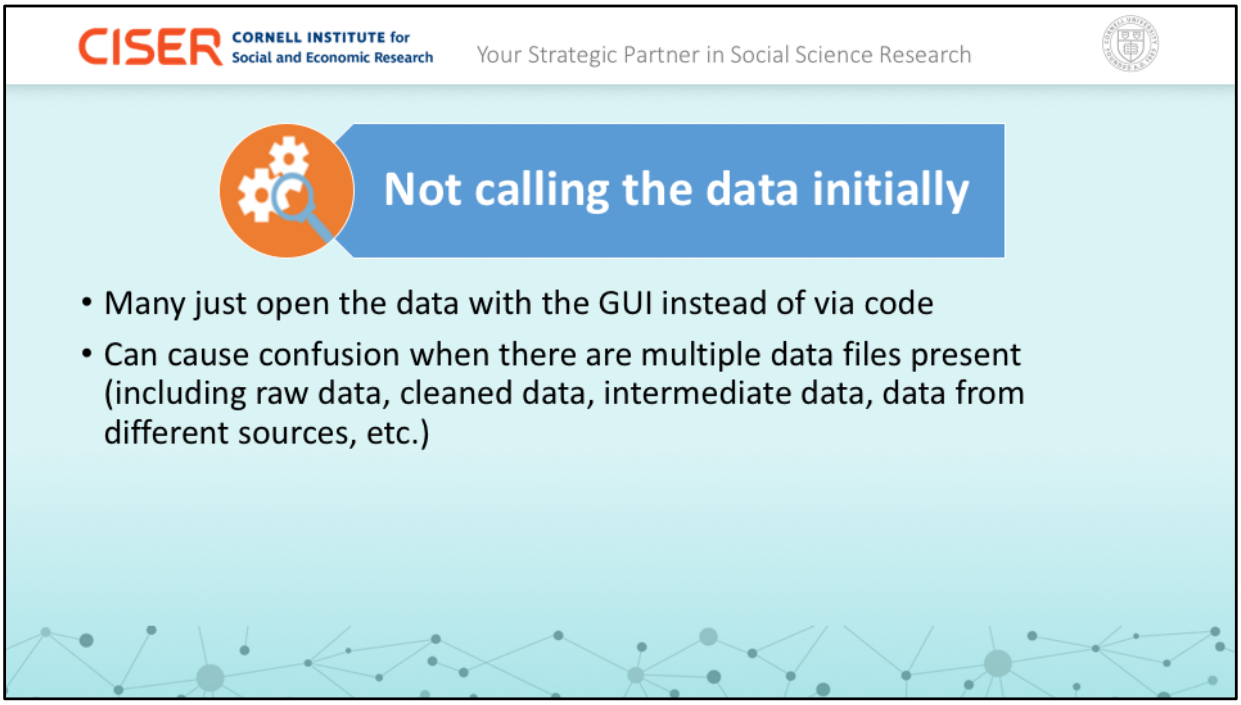
- Version control such as GitHub
- Dynamic Documentation
- Relative paths in coding

CISER CORNELL INSTITUTE for Social and Economic Research — Your Strategic Partner in Social Science Research

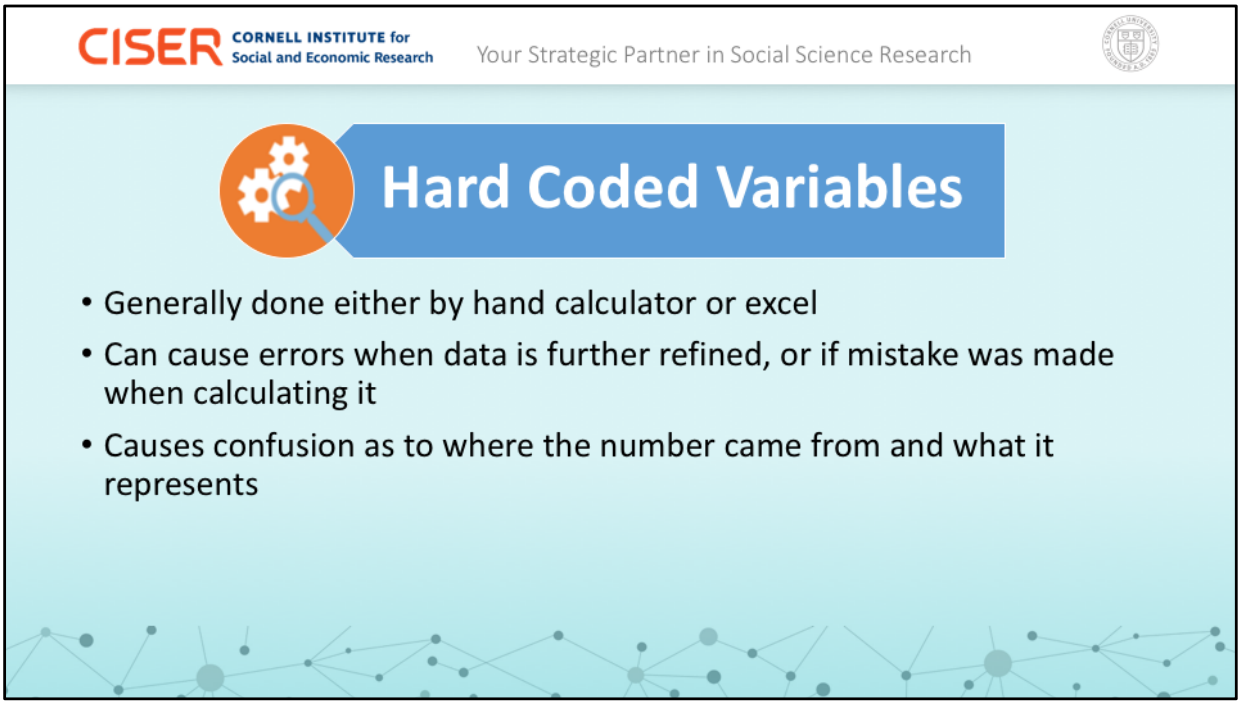
Most Common Error Types present in Scripts

- Not calling the data initially
- Hard Coded Variables
- Missing Files
- Poor or No Commenting
- Uncertain Order of Execution
- Doesn't follow the order of the Output
- Prerequisites
- Extraneous output/lines of code

For the next two sections, I will talk about the most common types of errors we see, and how often I've seen them in the past 10 papers that I have reviewed to give an idea of how common they are.



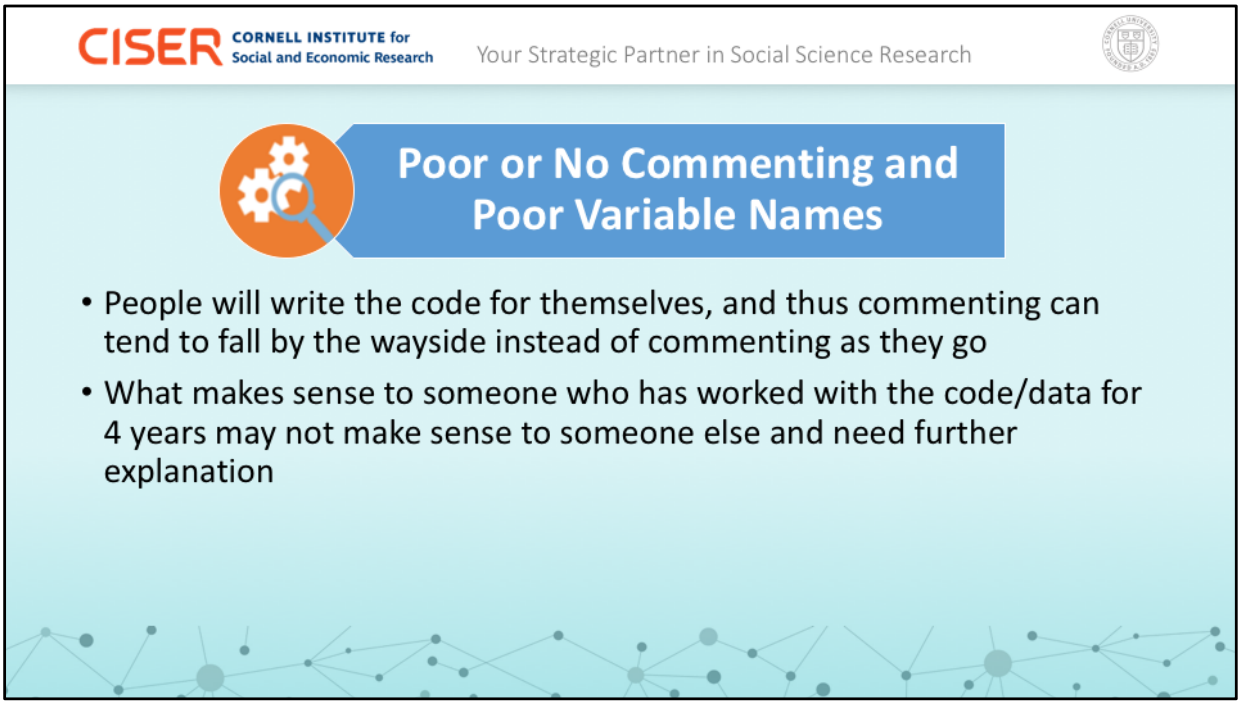
5 of last 10 had this problem


CISER
CORNELL INSTITUTE for Social and Economic Research
Your Strategic Partner in Social Science Research
Hard Coded Variables
• Generally done either by hand calculator or excel
• Can cause errors when data is further refined, or if mistake was made when calculating it
• Causes confusion as to where the number came from and what it represents

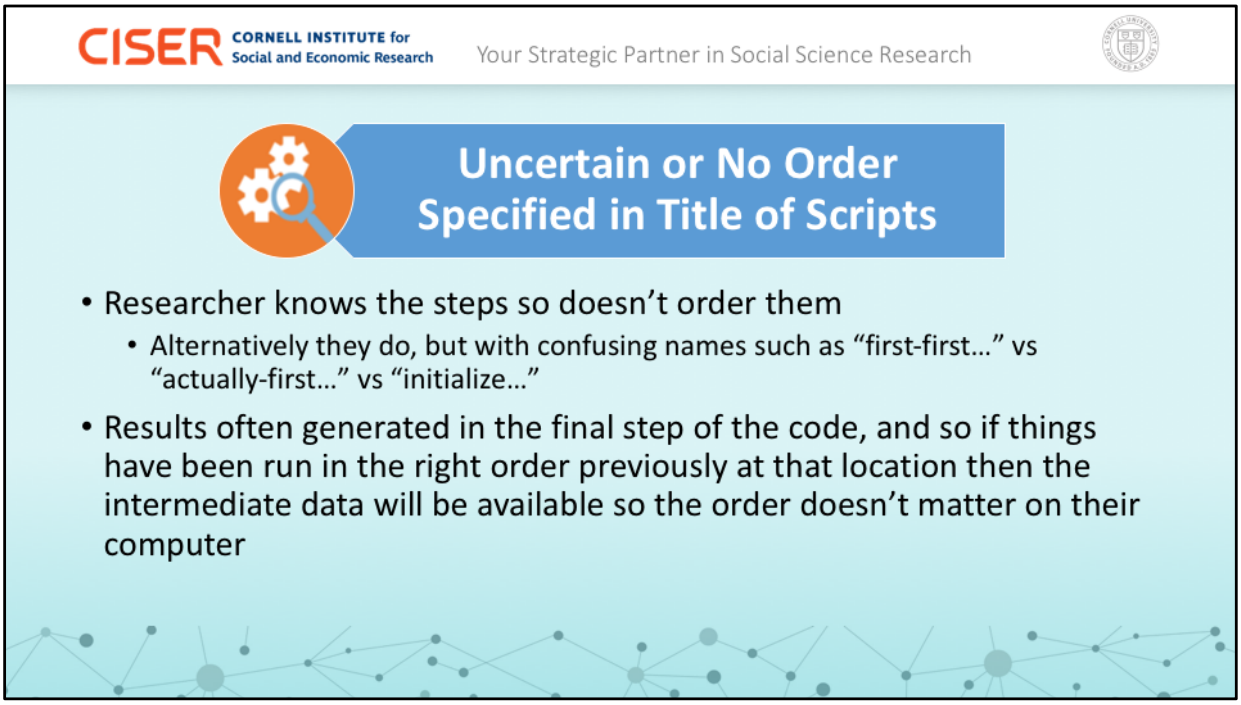
3 of last 10


CISER
CORNELL INSTITUTE for Social and Economic Research
Your Strategic Partner in Social Science Research
Missing Files
• Impossible to fully reproduce results with needed files missing
• Often done because researchers will work on multiple computers or utilize poor version control and delete things
• Can also be caused by hard coding a variable and then discarding the code

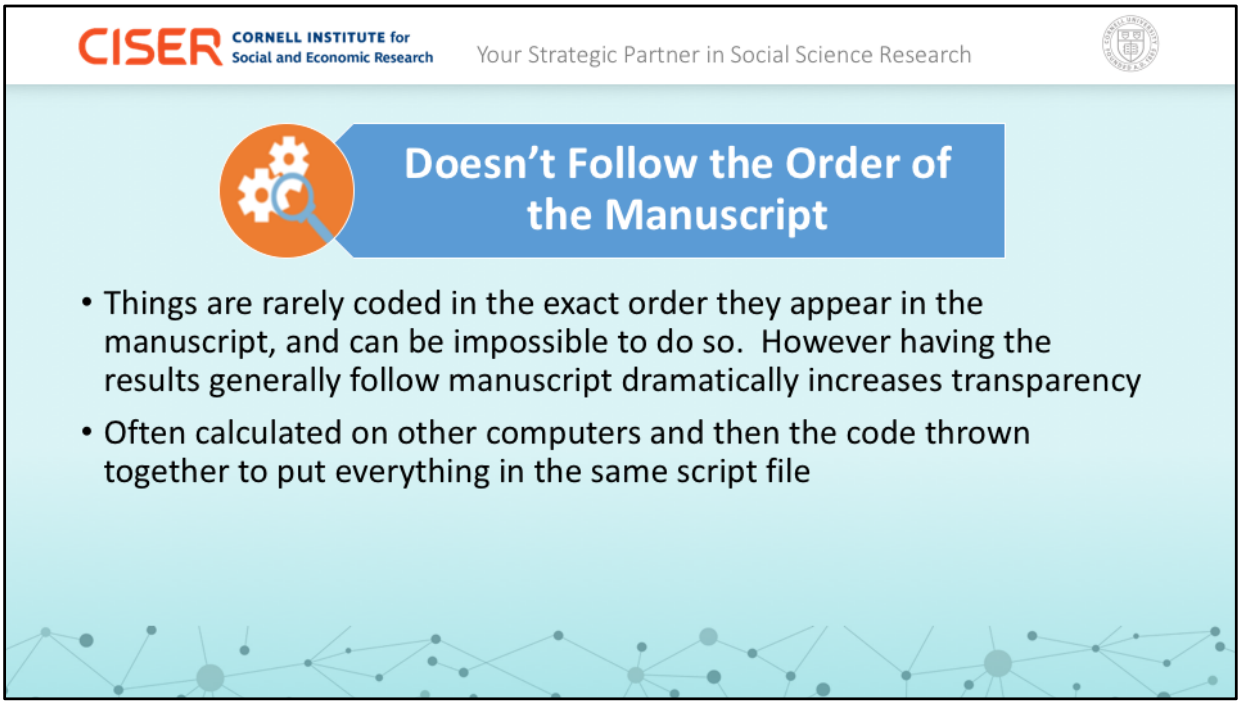
2 of last 10



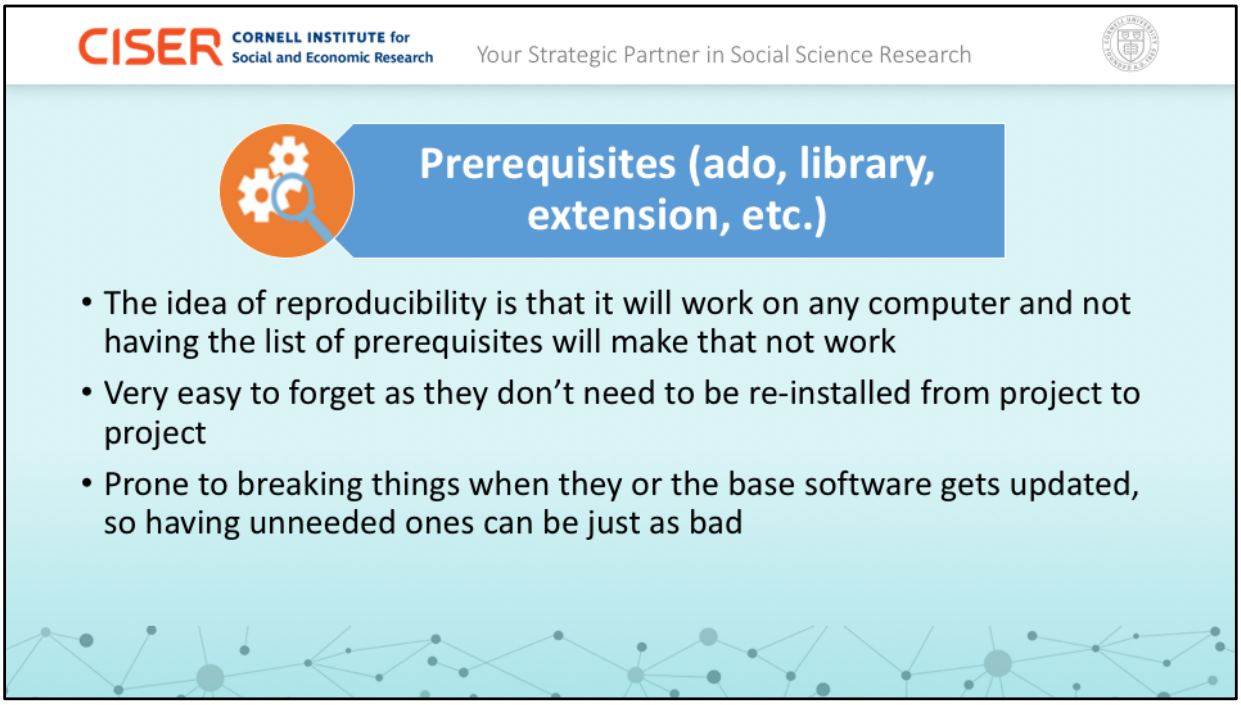
3 of last 10, but much more common than that



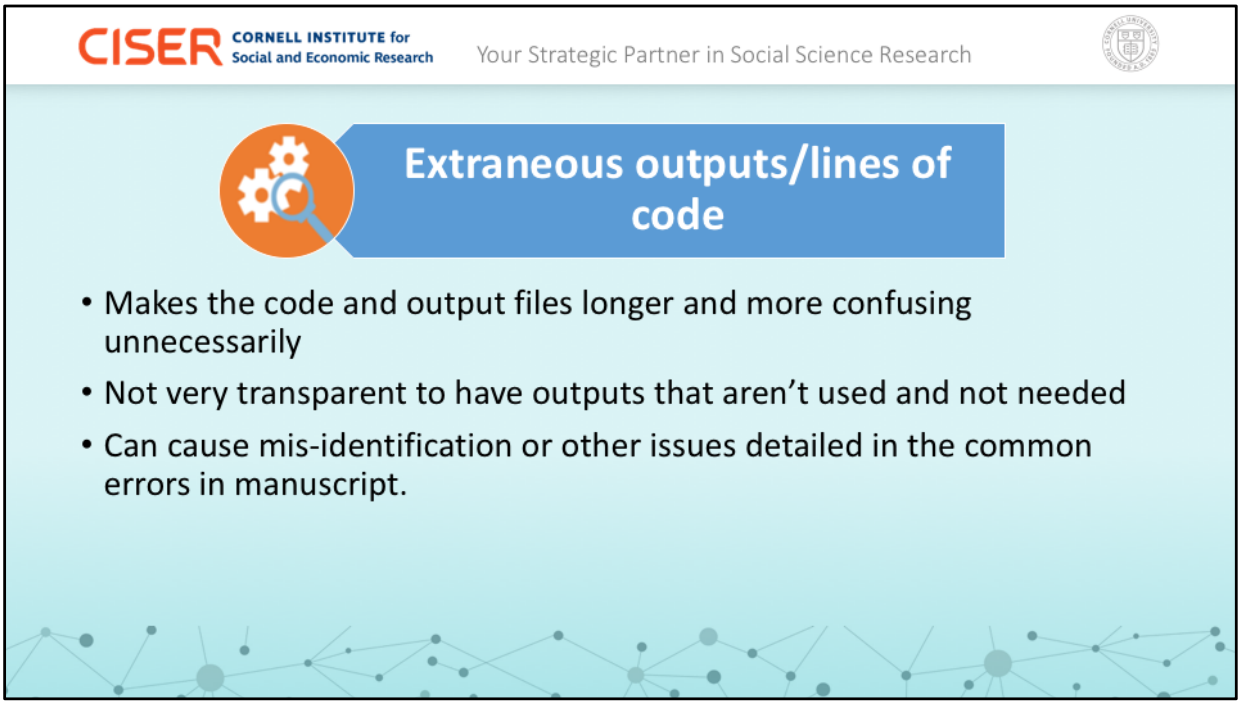
7 of last 10 missing Master File, but not always uncertain order (some of those no order)



Hard to say, as code and manuscript evolve over time.



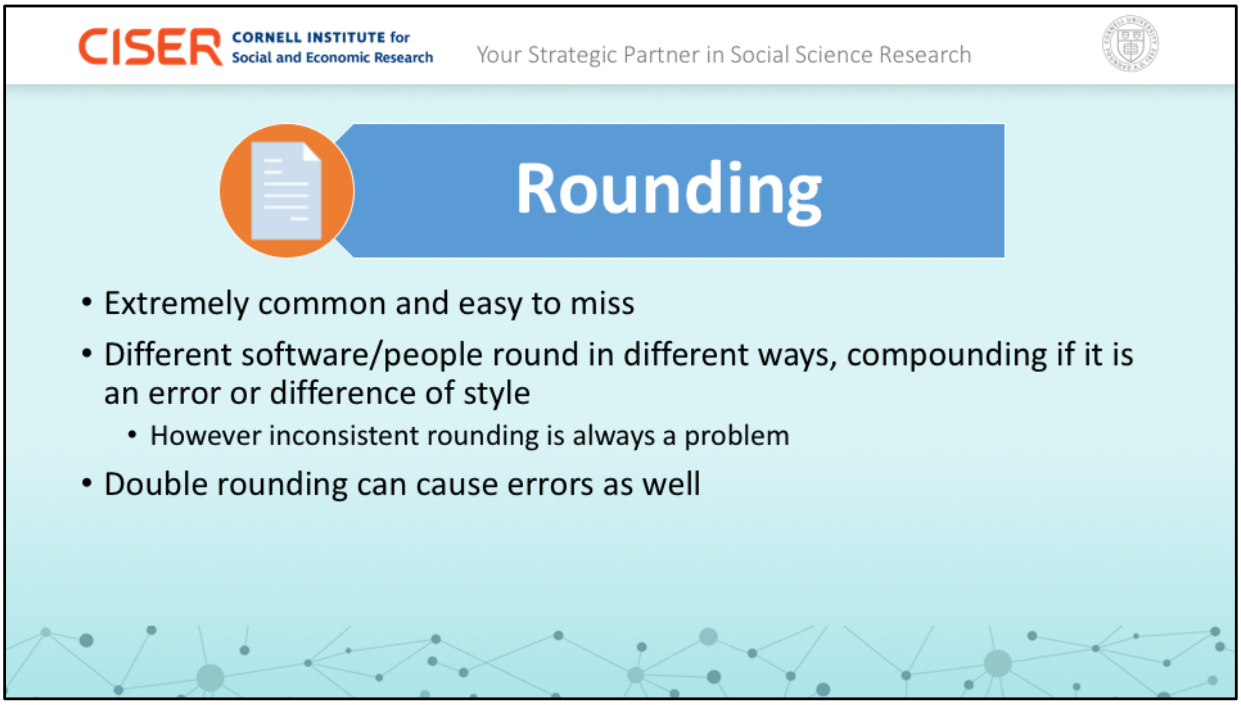
3 of last 10 were missing a prerequisite library when it was needed/suggestable



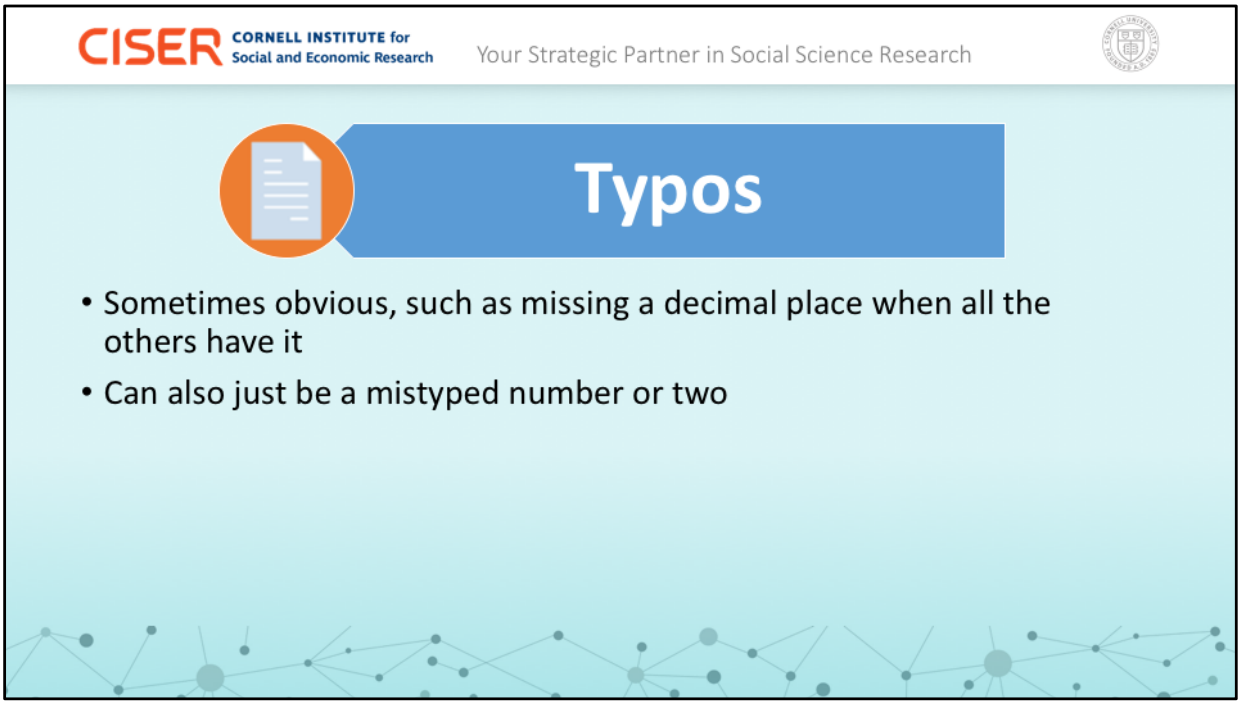
Similar to following the order of the manuscript, as calculations enter or leave the final manuscript, code will become unnecessary or new code needed.

Most Common Error Types present in Manuscripts

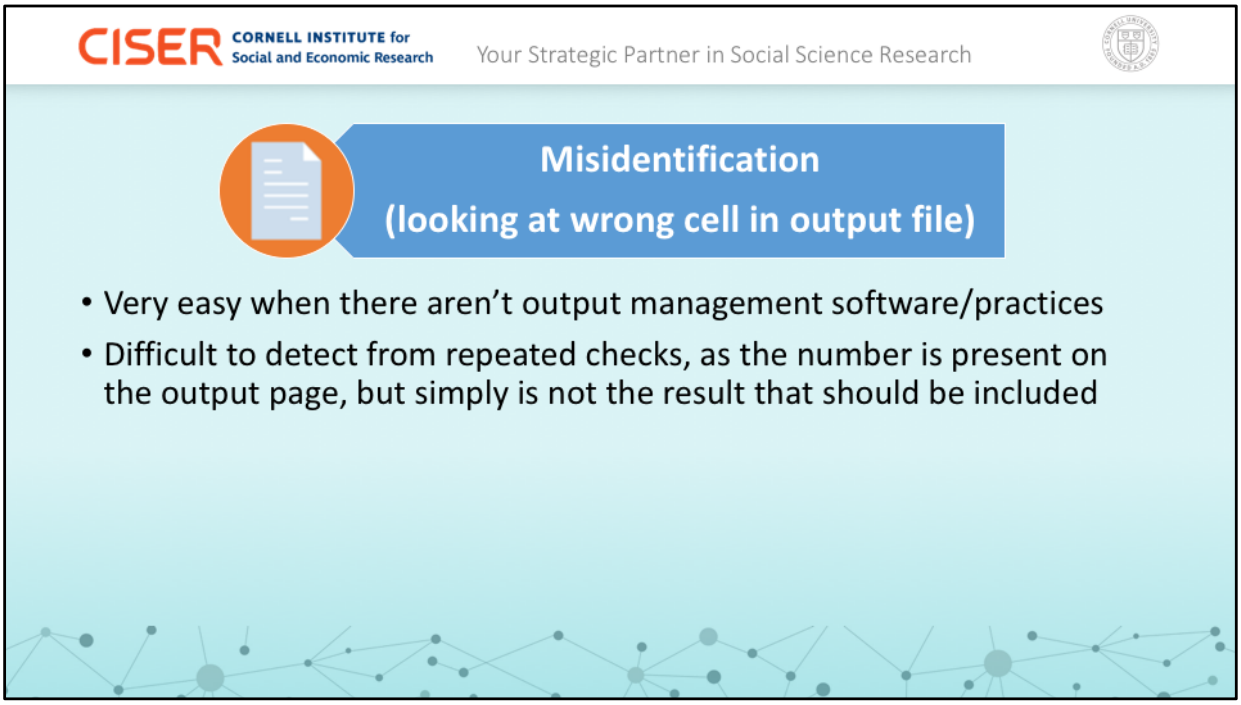
- Rounding
- Typos
- Misidentification
- Out of Date Tables
- Missing from Package
- Incorrectly Marking p-values



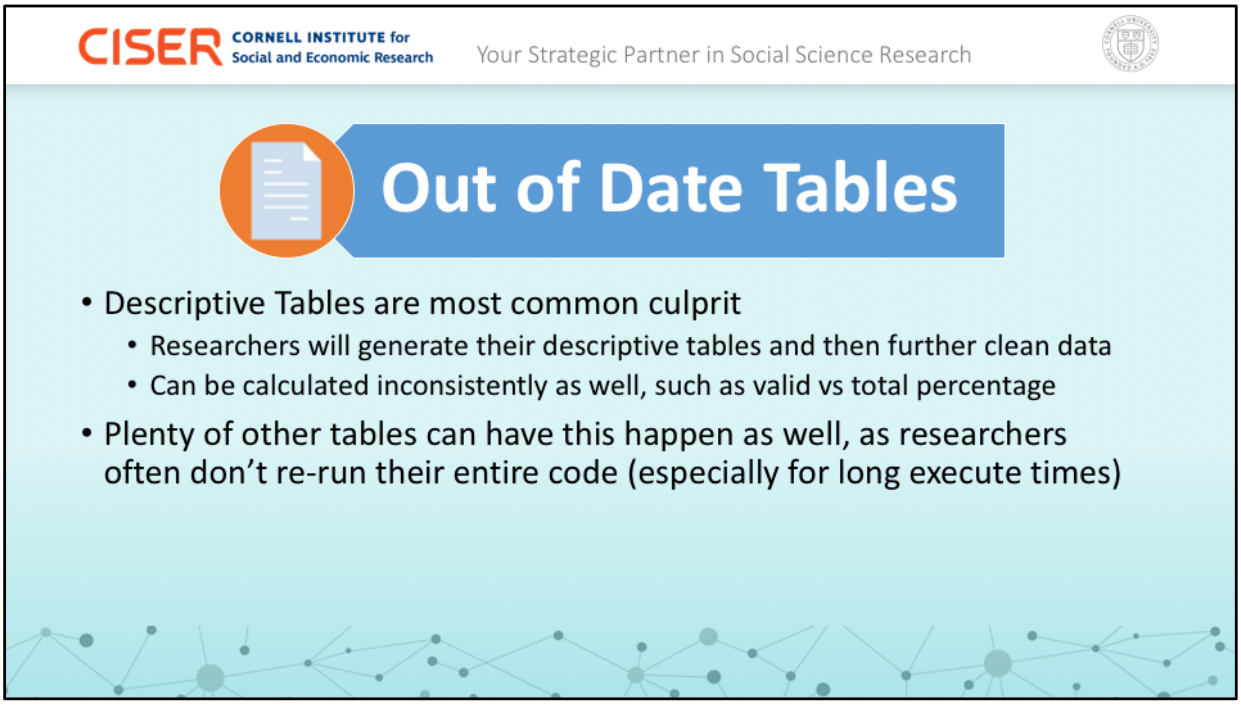
This is present in virtually all manuscripts, 9 of last 10 had an issue where rounding was most likely to blame.



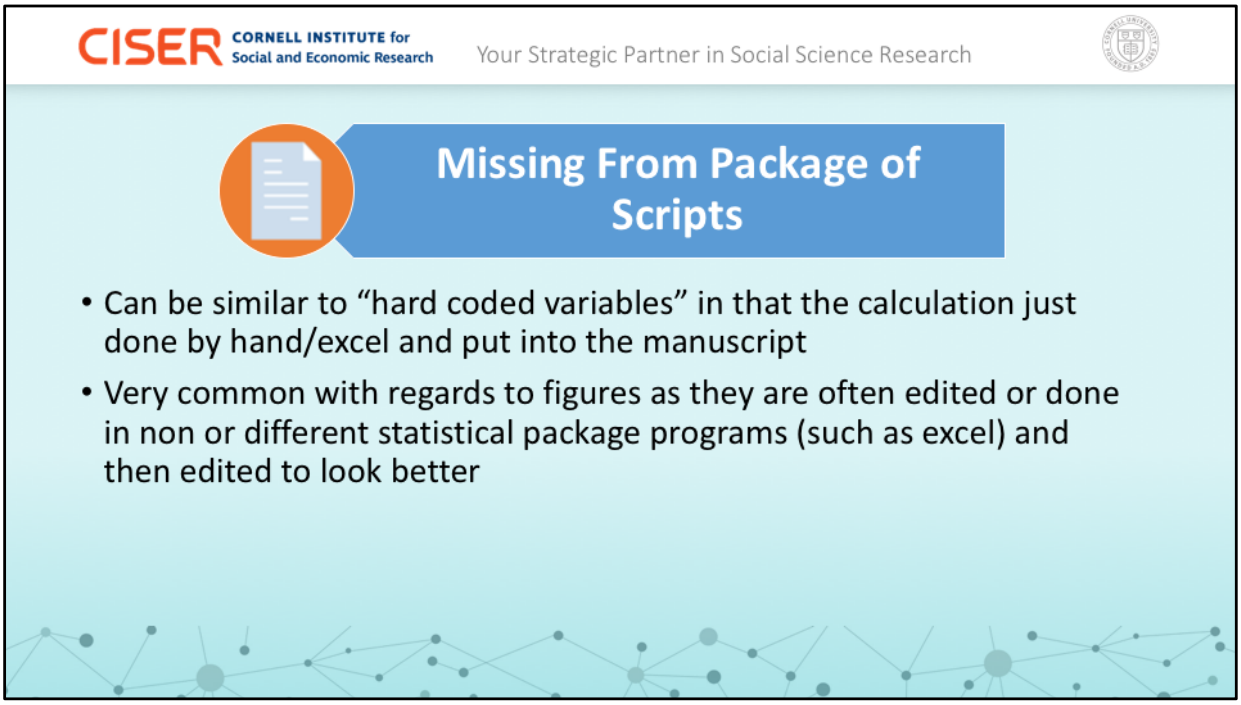
Can be hard differentiating this from rounding errors or the other type. It was certainly present in at least 2 of last 10.



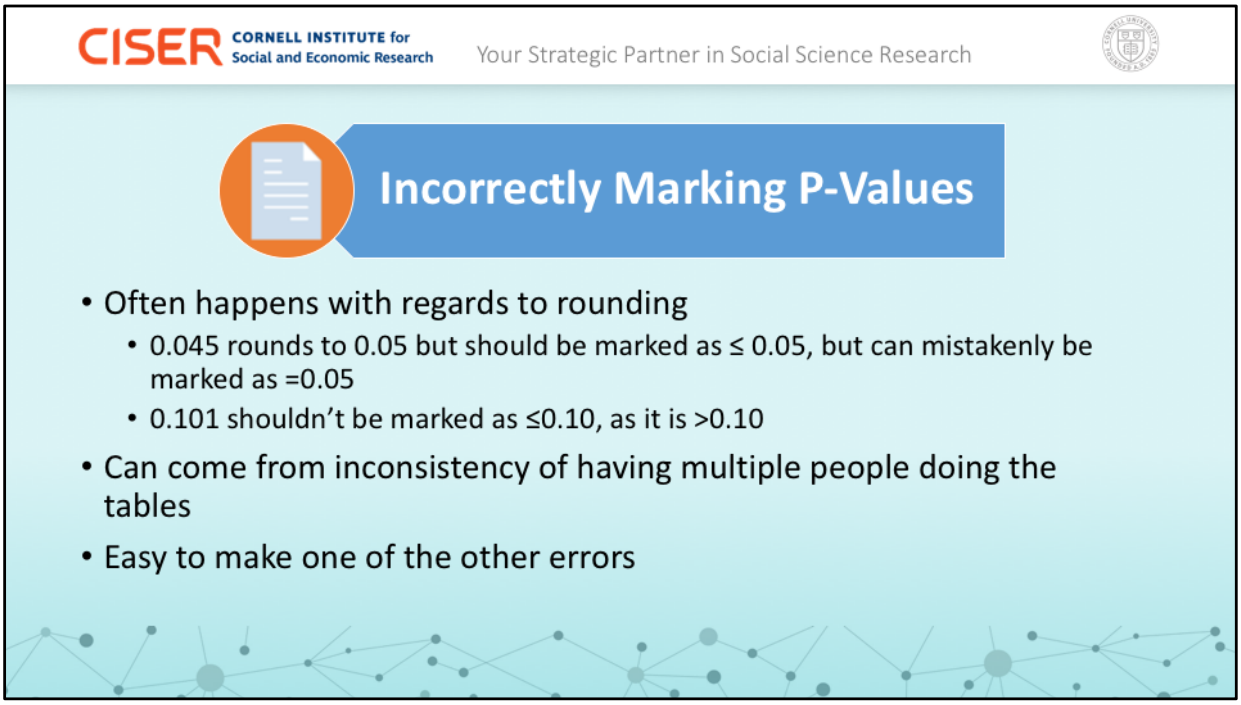
This wasn't present in any of the last 10, however we have had issues with this in the past, and it can be hard differentiating rounding/typos from this as well.



4 of the last 10 had issues that appear be related to out of date tables



3/10 of last 10. This occurs to differing degrees in many papers. Often this is done in excel or just by hand/hand calculator.



This happens fairly often when output management systems aren't used which we will talk about in the next section.

Good Practices

- Shared Workspace
- Versioning
- Output Management
- Relative Paths
- Folder Organization
- Scripts

Shared Workspace

- Shared both between the team working on reproduction and the client
- Helps to organize everything and ensures that changes are reflected between both
- Facilitates better back and forth between client and reproducer

Versioning

- Often will go hand in hand with the shared workspace
  - Such as github or even box
- Allows progress to be made on multiple ends, and for mistakes to happen but easily fixed all while retaining a clean work space

Output Management

- Actual output management code/prerequisites
  - Ado/library prerequisite such as Outreg2 in Stata or Huxreg/Huxtables in R
  - Isolating and saving just the values used in the paper into their own table
- Dynamic Documentation
  - Markdown type
    - R Markdown or Stata Markdown and pandoc
  - Notebook type
    - Jupyter Notebook or Mathmatica Notebook

Relative Paths

- Allows for the script to be run on other computers without editing them
- Helps to ensure that everything is present, as if file is missing and only relative paths are used the script will fail, where as if there is an absolute path it will still work on that computer but not any others

Folder Organization

- Facilitates relative paths and leads to a much cleaner workspace
- TIER protocol
- Personal preference of organization with named folders and not everything in same place

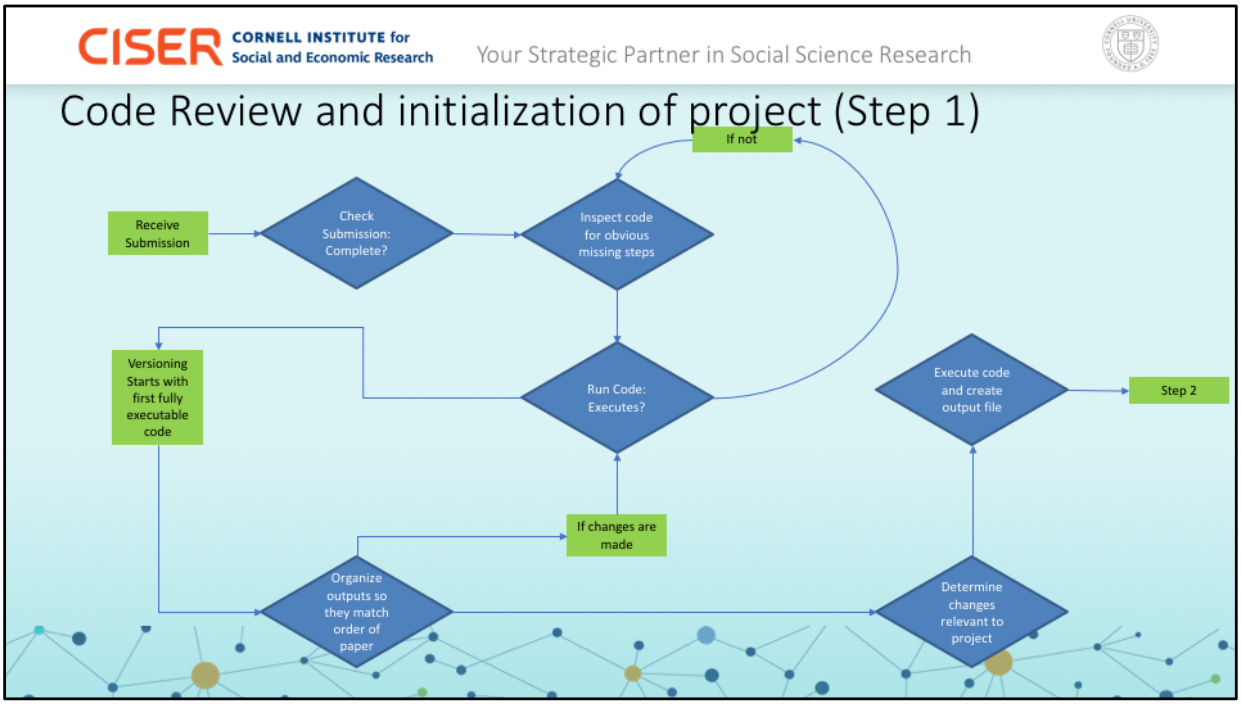
Scripts

- Well named and numbered to show sequence order
- Master script so only one script needs to be run
- Prerequisites listed out or automatically installed
  - Best to have them locally installed to specific folder to ensure none are missing
  - Prerequisites folder should be part of folder structure
- Code header present
- Execute beginning to end fresh installation of software

CISER CORNELL INSTITUTE for Social and Economic Research

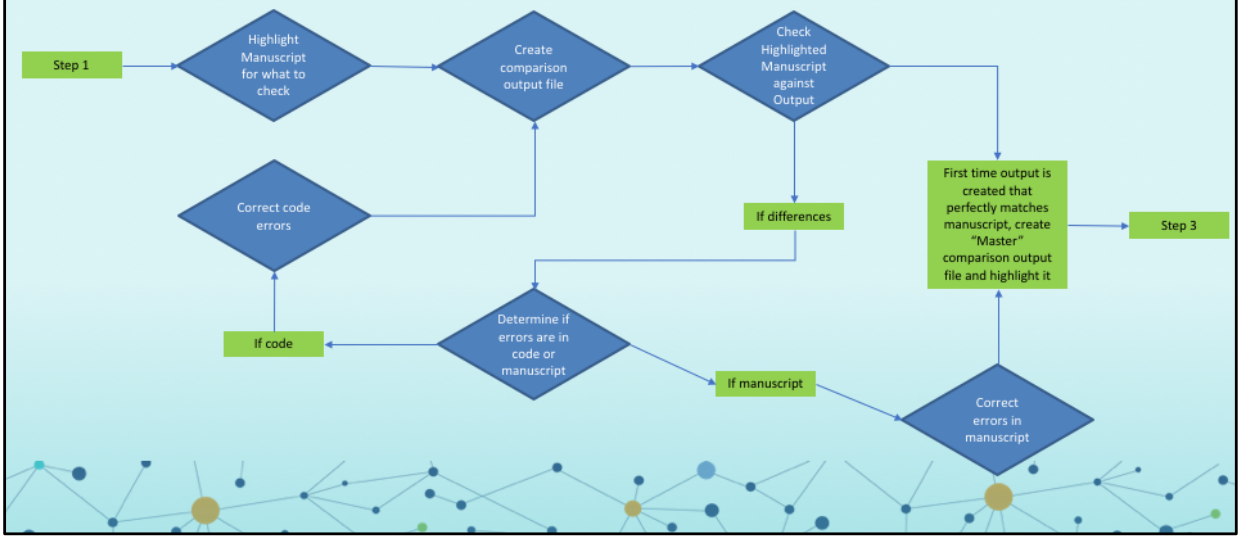
Your Strategic Partner in Social Science Research

Code Review and initialization of project (Step 1)



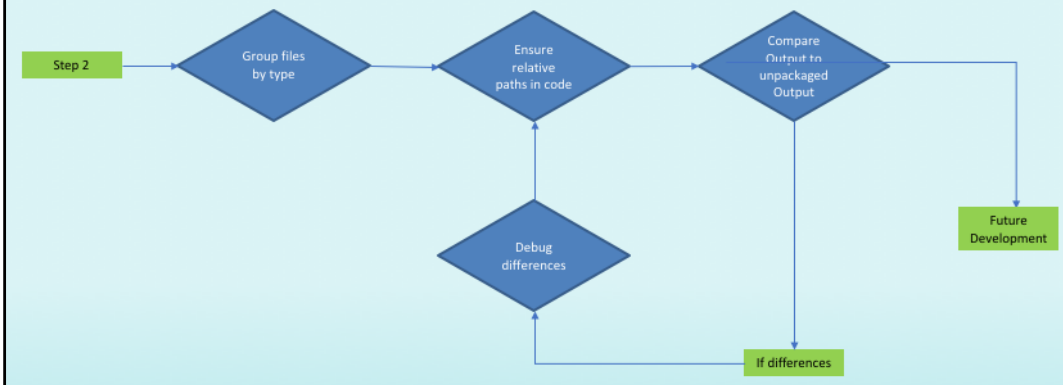
CISER CORNELL INSTITUTE for Social and Economic Research Your Strategic Partner in Social Science Research

Comparison Output (Step 2)



CISER CORNELL INSTITUTE for Social and Economic Research Your Strategic Partner in Social Science Research

Package Creation (Step 3)



Conclusion

Many of the common errors are related and can be solved in similar ways such as output management.

Each project is fairly unique as clients come with very different starting places and very different needs

CISER CORNELL INSTITUTE for Social and Economic Research

Your Strategic Partner in Social Science Research

THANK YOU

ciser@cornell.edu

Contact information