# Guidelines for Computational Reproducibility in Economics

## Definitions to Assess Computational Reproducibility and Possibilities for Improvement

### Definitions

#### Stages of the exercises

scope --> assemnt --> imprv --> verif --> exttn  
 ▲ |  
 | |  
 |\_\_\_\_\_\_\_\_\_|

##### Scope

###### Suggestions on how to prioritize

* First main estimates from paper.
* Then main 2 main tables and/or two main figures.
* Once done with improvements stage of these outcomes, then go back to rest of the paper.

Another approach: - get everything to run up to analytic data, then go to improvements. Once completed go back for raw data (or in parallel?)

##### Assessment

##### Improvement

##### Verification/robustness

##### Extension

#### Types of reproduction

##### By outputs reproduced:

* **Main results:** a successful reproduction of main results would have obtained the same estimates as in the original paper, for the estimates that the authors highlight the most in either the abstract, introduction or conclusion of the paper. If no estimate is highlighted, then the researcher conducting the reproduction should choose the main result and declare it at the begging of the reproduction.
* **Complete:** a successful reproduction is complete when it is possible to obtain the same estimates as the original study for all the outputs presented in the paper. This includes: tables, figures, and inline estimates both in the main body of the paper and all the appendicess.

##### By sources used:

* **Computationally Reproducible from Analytic data (CRA):** an output is CRA if it can be reproduced with minimal effort starting from the analytic data sets.
* **Computationally Reproducible from Raw data (CRR):** an output is CRR if it can be reproduced with minimal effort from the raw data sets.

#### Types of data:

* **Raw data:** A data set is considered to be raw, if it corresponds to the a unmodified file that was obtained by the authors from the sources cited in the paper. The only possible modification that can be made to raw data, without changing its category to processed data, is that of deleting personally identifiable information.
* **Processed data:** a raw data set that has gone through any transformation should be defined as processed data. Processed data can be separated into intermediate data and analytic data.
  + *Intermediate data:* a processed dataset is defined as intermediate if it is not directly used as final input for an analysis in the final paper (including appendix). Intermediate data should not contain direct identifiers.
  + *Analytic data:* data will be defined as analytic if it will be used as the last input in the workflow, to produce a statistic displayed in the final paper (including the appendix).

##### How to classify a data set?

To classify a data set as **raw** you can use some of the following strategies:  
- Check if the data is stored in a folder denominated as “raw”  
- Check if the filename has the word “raw” in some part of it  
- Verify that the data set is not the output of any code script in the reproduction materials  
- Verify that the same file can be independently obtained from the data source cited in the paper

To classify a data set as **analytic** you can use some of the following strategies:  
- Check if the data is stored in a folder denominated as “analytic” or “analysis”  
- Check if the filename has the word “analytic” or “analysis” in some part of it  
- Verify that the data set is the last input required to produce some of the output (formatted or unformatted) of the paper

#### Types of code:

* **Cleaning code:** a script should be classified as primarily data cleaning if most of its content is dedicated to actions such as: deleting variables or observations, merging data sets, removing outliers, and reshaping the structure of the data (from long to wide or vice versa).
* **Analysis code:** a script should be classified as primarily analysis code if most of its content is dedicated to actions such as: running regressions, running hypothesis tests, computing standard errors, and imputing missing values.

#### Types of output:

* **Tables:**
* **Figures:**
* **In-line:**

#### Other definitions:

* **Minimal effort:** the definition of minimal effort we will use here is that of spending five minutes or less in getting the code running. This five minutes do not include the computing time.
* **Coding errors:** a coding error will occur when a section in the code, of the reproduction package, executes a procedure that is in direct contradiction with the intended procedure expressed in the documentation (paper or comments of the code). For example an error happens if the paper specify that the analysis is perform on the population of males, but the code restricts the analysis to females only. Please follow the [ACRE procedure to report coding errors](ADD%20LINK).

### Possible improvements:

* **Add missing analysis data files (+AD):** if found missing from the reproduction materials, analysis data sets could be added. Researchers performing the reproduction are encourage to follow these steps:
  + 1 - Identify specific name of the missing data set. Typically this information can be found in some of the analysis code that calls such data in order to perform an analysis (eg analysis\_data\_03.csv).
  + 2 - Verify that such data cannot be obtained by running the data cleaning code over the raw data.
  + 3 - [Verify the ACRE database](ADD%20LINK) for previous attemps to contact the authors on this topic.
  + 4 - [Contact the authors](contact_authors.md) and request the specific data set.
* **Add missing raw data files (+RD):** most reproductions packages do not include all the original raw datasets. To obtain any missing raw data, follow the same steps (1-4) recommended for the adding analysis data sets. If the data sets are not available due to confidentiality or proprietary issues, the researcher conducting the reproduction can still improve the reproduction package by including a detailed set of instructions, including contact information and possible costs, for future researchers to follow.
* **Add missing analysis code (+AC):** analysis code can be added when there are analytic data files, but some or all the methodological steps are missing from the code. In this case researchers conducting the reproduction to follow the these steps:
  + 1 - Identify specific line/paragraph in the paper that describes the missing analytic step in the code (eg “we impute missing values to…,” or “we estimate this regressions using a bandwidth of …”).
  + 2 - Identify the code file and approximate line in code where the analysis could be carried out.
  + 3 - [Verify the ACRE database](ADD%20LINK) for previous attempts to contact the authors on this issue.
  + 4 - [Contact the authors](contact_authors.md) and request the specific code files, following the ACRE sample language to request the specific analysis.
  + 5 - If no response from the authors, researchers reproducing the paper are encourage to attempt to recreate the analysis based on their interpretation of the paper, and filling in any missing piece by making explicit assumptions.
* **Add missing data cleaning code (+CC):** data cleaning/processing code can be added when there are certain steps missing in the creation/recoding of variables, merging, subsetting of the data sets, and other cleaning related processes. Researchers conducting the reproduction should follow the same steps (1-5) as when adding missing code.
* **Debug analysis code (DAC) or debug cleaning code (DCC):** whenever any code is available in the reproduction package, the researcher performing the reproduction will be able to debug those scripts. Here four types of debugging are suggested to label the different modifications perform in the reproduction:
  + *Code cleaning:* whenever set of instructions is simplified (e.g. by wrapping repetitive steps in a function or a loop) or when redundant code is removed (eg. old code that was commented out) but the original output remains intact.
  + *Performance improvement:* whenever a set of instructions is replace by other that perform the same tasks but take less time (eg. choosing one numerical optimization algorithm over another, but obtaining the same results).
  + *Environment set up:* whenever the code has to be modified to include correct paths to files, specific versions of software, and instructions to install missing packages or libraries.
  + *Correcting errors:* whenever a coding error is detected, please follow the [ACRE procedure to fix and report coding errors](ADD%20LINK).

### Levels of Computational Reproducibility

With the definitions and dimensions for improvement provided above, we can now outline different levels of computational reproducibility. Each level is defined on the basis of data and code availability, possible improvements, and weather or not to achieves some type of reproducibility.

The assessment is made at the output level. Hence a paper can be highly reproducible for its main results, but suffer of low reproducibility for other outputs (like tables and figures).

The following figure summarizes the different levels of computational reproducibility (for any given output). For each level, there will be possible improvements that have been done already (-), that can be done to move up one level of reproducibility (✔) or that are out of reach given the current level of reproducibility (x).

Figure 1: Levels of Computational Reproducibility and Possible Improvements   
  
 | | Possible improvements |  
 | Level |+AD|+RD|+AC|+CC|DAC|DCC|  
 --------|---|---|---|---|---|---|  
What data are available? | | | | | | | |  
├── None ..................................| L1 | ✔ | ✔ | x | x | x | x |  
├── Analytic data only. Code? | | | | | | | |  
| ├── No code or cleaning code only......| L2 | - | ✔ | ✔ | x | x | x |  
| └── Analysis code only. Is it CRA? | | | | | | | |  
| ├── No..............................| L3 | - | ✔ | - | x | ✔ | x |  
| └── Yes.............................| L4 | - | ✔ | - | x | - | x |  
└── Raw & Analytic data. Code? | | | | | | | |  
 ├── None ...............................| L5 | - | - | ✔ | ✔ | x | x |  
 ├── Analysis code only. CRA? | | | | | | | |  
 | ├── No...............................| L6 | - | - | - | ✔ | ✔ | x |  
 | └── Yes..............................| L7 | - | - | - | ✔ | - | x |  
 └── A. and cleaning code. Is it CRR? | | | | | | | |  
 ├── No. CRA? | | | | | | | |  
 | ├── No...........................| L8 | - | - | - | - | ✔ | ✔ |  
 | └── Yes..........................| L9 | - | - | - | - | - | ✔ |  
 └── Yes.............................| L10 | - | - | - | - | - | - |

**Level 1 (L1):** There are no data or code available of any type. Possible improvements include: adding raw data (+AD) and adding analysis data (+RD).

**Level 2 (L2):** There are analytic data available, but no raw data or any type of code. Possible improvements include: adding raw data (+RD) and adding analysis code (+AC).

**Level 3 (L3):** Both analytic data sets and analysis code are available. However the code does not run or produces different results that those of the paper (not CRA). Possible improvements include obtaining raw data (+RD) or debugging the analysis code (DAC).

**Level 4 (L4):** Both analytic data sets and analysis code are available, and they produce the same output as in the paper (yes CRA). The reproducibility package can still be improved by obtaining the original raw data sets, or by documenting the steps required to obtain those files.

**Level 5 (L5):** All data, analytic and raw, are available. However some or all the codes for cleaning and analysis are missing. Steps for improvement include adding analysis code (+AC) and/or cleaning code (+CC).

**Level 6 (L6):** All data and analysis code are available. However the code does not run or produces different results that those of the paper (not CRA). Possible improvements include adding the missing cleaning code (+CD) or debugging the analysis code (DAC).

**Level 7 (L7):** All data and analysis code are available, and they produce the same output as in the paper (yes CRA). The reproducibility package can still be improved by adding adding the missing cleaning code (+CD).

**Level 8 (L8):** All materials (raw and analysis data, and cleaning and analysis code) are available. However the code does not run or produces different results that those of the paper (not CRR and not CRA). Possible improvements include debugging the cleaning code (DCC) or debugging the analysis code (DAC).

**Level 9 (L9):** All materials (raw and analysis data, and cleaning and analysis code) are available, and the analysis code produces the same output as in the paper (yes CRA). However the cleaning code does not run or produces different results that those of the paper (not CRR). Possible improvements include debugging the cleaning code (DCC).

**Level 10 (L10):** All materials are available and produce the same results as in the paper with minimal effort, starting from the analytic data sets (yes CRA) and from the raw data (yes CRR).

The next section describes improvements that can be applied across levels and are meant to improve the readability of the reproducibility package.

### Additional Paper-Level Improvements Across Categories

In addition to the different levels of computational reproducibility described in the previous sections, this section highlights six additional improvements that researchers conducting the reproduction can do to improve the overall reproducibility of a paper. This additional improvements can be apply across levels (including level 10). - Set up the replication package using version control software (Git). - Improve documentation: - Add extensive comments to the code. - Adapt the paper into a literate programming environment (eg: Jupyter notebook, RMarkdown, Stata Dynamic Doc). - Re-write the code from a proprietary statistical software (eg Stata, Matlab) into a open source statistical software (eg R, Python, Julia). - File organization and master script: re-organize the reproduction materials into a set of folders and sub-folders that follow [standardized best practices](https://www.projecttier.org/tier-protocol/specifications/#overview-of-the-documentation), and add a master script that executes all the code in order and with no further modifications. ADD LINK TO A NICE MASTER SCRIPT. LARS?  
- Set up a computing capsule that executes all the reproduction in the browser without the need to install any software. See for examples [Binder](https://mybinder.org/) and [Code Ocean](https://codeocean.com/).

### Identifying Analytical Choices

As part of the requirements to [demonstrate comprehension of the paper and the code](requirements_comprehension.md) researchers conducting the reproduction will be asked to record all the analytical choices identified during the code review process. This is done in two steps: first adding comment lines into the code files where an analytic choice are found, and second, compiling those analytic choices into a standardized data set.

In your copy of the replication code, add the comment “# ANALYTICAL CHOICE OF TYPE \_\_\_\_. RECORDED FOR THE FIRST TIME [HERE or IN "FILE\_NAME-LINE\_NUMBER"]” above each analytical choice detected in the code. Possible types of analytical choices include (but are not limited to):

* Analytical choices in data cleaning code:
  + Variable definition
  + Data sub-setting
  + Data reshaping (merge, append, long/gather, wide/spread)
  + Others (specify as “processing - other”)
* Analytical choices in analysis code:
  + Regression function (link function)
  + Key parameters (tuning, tolerance parameters, etc.)
  + Controls
  + Adjustment of standard errors
  + Choice of weights
  + Treatment of missing values
  + Imputations
  + Other (specify as “methods - other”)

Once finished, transcribe all the information on analytical choices into a data set. For the source field type “original” whenever the analytical choice is identified for the first time, and file\_name-line number every time that the same analytical choice is applied subsequently (for example if a analytic choice is identified for the first time in line 103 and for a second in line 122 their respective values for the source field should be original and code\_01.do-L103 respectively).

The resulting data base should have the [following structure](https://docs.google.com/spreadsheets/d/1nZuJSHswbZgaaIfBcyIUGPwG-WIP8zE1Oambud-WoDc/edit?usp=sharing):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| file\_name | line\_number | choice\_type | choice\_value | Source |
| code\_01.do | 73 | data subsetting | males | original |
| code\_01.do | 122 | variable definition | income = wages + capital gains | “code\_01.do-L103” |
| code\_05.R | 143 | controls | age, income, education | original |
| … | … | … | … | … |

### Additional resources

* [Project TIER](https://www.projecttier.org/tier-protocol/)
* [IDB’s cheatsheet for transparency, reproducibility and ethics](http://idbdocs.iadb.org/wsdocs/getdocument.aspx?docnum=EZSHARE-1350314980-383)
* [Lars Vilhuber LDI’s Wiki for Reproducibility](https://github.com/labordynamicsinstitute/replicability-training/wiki). Particularly [this section](https://github.com/labordynamicsinstitute/replicability-training/wiki/Prepare_and_run_replication).
* World Bank [DIME’s Wiki for transparent and reproducible research](https://dimewiki.worldbank.org/wiki/Main_Page).
* Dynamic documents in [R](https://rmarkdown.rstudio.com/gallery.html), [Python](https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#economics-and-finance) and [Stata](https://github.com/BITSS/CEGA2019/blob/master/03-extra_dynamic_docs/02b-Stata-markdown/Stata%20Markdown.pdf)
* Git resources:
  + [Jenny Bryan’s book](https://happygitwithr.com) and [video](https://www.rstudio.com/resources/videos/happy-git-and-gihub-for-the-user-tutorial/)
  + [Github learning lab](https://lab.github.com/)
  + [Udacity’s intro](https://www.udacity.com/course/how-to-use-git-and-github--ud775)
  + [Git for poets](https://www.youtube.com/playlist?list=PLRqwX-V7Uu6ZF9C0YMKuns9sLDzK6zoiV)
  + [Combining GitHub and Dropbox](https://github.com/kbjarkefur/GitHubDropBox)
  + [Atlassian intro to Git](https://www.atlassian.com/git/tutorials)
  + [Software Carpentry tutorial from the command line](https://swcarpentry.github.io/git-novice/)
* [Open Science Framework (OSF)](https://osf.io)
* [R for Stata users](https://github.com/hblackburn/R4Econ/blob/master/Resources.md)