DIME Analytics
Peer Code Review - Reviewer Feedback Checklist
v2.0

This checklist lists important factors to consider while reviewing your code review partner's code package. Please use this checklist while reviewing the code package.

NOTE: Your deliverable from this exercise will be this online survey form which replicates the checklist below.

Overview - First Impression

Once you have received a package from your code review partner, please check for the following general recommended practices:

Data cleaning, variable creation, and analysis are all done in separate scripts

Analysis scripts do not include any data processing, unless necessary for the creation of a table or graphic

Each script is fully modular - that is to say that the scripts can be run independently from the master script, and do not depend on having the results of other scripts in memory.

All scripts are well-commented

All scripts are well-organized and formatted, such that one can easily identify functional chunks of code and evaluate whether they correctly implement the econometric or statistical process described

Tables and charts are exported (preferably as raw text files)

Computational Reproducibility (applicable only if de-identified data was shared with you)

Next step: If you received data as part of the peer review package, attempt to run the code files. Ideally, you will be able to do so by changing only the top-level directory in the main script. Take note of any issues in reproducing code outputs (constructed indicators, cleaned data, graphs, and tables) as you will need to record them in this review summary form.

I was able to run all code files provided in the review package

Code did not run initially, but I was able to identify the reason

Outputs shared (if any) reproduced exactly

Outputs did not reproduce but I was able to determine the reason

Ease of Use and Understanding

Now, assess how easy it is to understand the code you are reviewing. Consider whether the documentation provided is sufficient, and whether the code is organized in such a way that it would be easy for someone else to work on it. Make note of what additional documentation is needed as you will be asked to share detailed feedback in this review summary form.

I understood the code from original documentation and README file

The code is organized such that if I had to change a core task I could do it by making changes in only one place

A new RA joining the project would be able to understand and contribute to the code

How many days are needed to understand the code well enough to contribute? ***(0-3 days)***

Coding Practices

Next, select the box next to each practice if the mentioned coding practice was used in some form in the code package. Also make note of whether it was implemented perfectly, or if there is room for improvement. You will need to provide this information in this detailed review summary form.

Master script

Use of comments

Documentation

Folder organisation

Good naming practices - consistent variable and file names

Code organization - sections, tracking inputs and outputs for each script

Use of white spaces

Use of loops, or functions, to handle repetitive tasks in R **(map, apply, sapply)**

Abstraction - using functions to do commonly repeated tasks

Final Comments

Finally, make note of the main strengths of the code, suggestions on how the coder can improve, and any additional feedback from your side. You will need to provide this information in this detailed review summary form.)

1.) Three main strengths of the code reviewed:

(For example, use of new functions, efficient scripts with proper use of loops, modular scripts that run directly through a main script)

2.) Suggestions on how the coder can improve:

(For example, suggestions on using loops in specific places to make the code more efficient, suggestions on functions that could simplify certain tasks, using commands like `ieboilstart` to specify software version)

3.) Any additional feedback for the coder:

(For example, implementing folder structures using `iefolder`, additional recommendations for creating reproducible outputs, or fixing specific issues highlighted in the previous section)