Code review checklist

# Code review checklist

## Readability and style

### Do you understand what the code is doing?

* What parts do you not understand?
* What type of comments or documentation would make it easier for you to understand them? What changes to the code would make it more understandable? Note that comments can be helpful to explain what the code is doing, but if the code isn’t self-explanatory, it should be made simpler.
* Are there any parts of the code that you can understand, but that are too complex? For example, were there any parts of the code that you had to read over and over to understand? Are there parts of the code that you would find difficult to change if you wanted to modify them?

### Do you understand why a task is implemented the way it is?

* Are there any tasks you would approach differently?

For example, would you use a different function or package? Would you change the order in which things are done?

* What decisions do you not understand? What type of comments or documentation would make it easier for you to understand them?

### Style

* Does the code follow a style guide?
* Are variable, file, and function names clear and communicative?
* Are lines in the script short?

As a rule of thumb, each line should have no more than 80 characters.

* Is each script in the project short?

As a rule of thumb, each script should have no more than 200 lines.

* Are white spaces and indentation used to make the code clearer and more accessible?

## Correctness

* Does the code do what it is supposed to do?

For example, does the code to construct indicators follow their definition from the documentation? Do regressions follow the model specified?

* Are there any unintended consequences for this line code? Be particularly mindful of operations that change the structure of the data, such as combining or reshaping data sets and aggregating variables or observations.

For example, is it changing the number of observations, the number of missing values, or the results? Is it replacing files or variables that should not be edited?

* Are any values hard-coded into the code? Could they be calculated in the code instead? Could this input be defined in a different part of the code to make it more salient?
* Are there any sanity checks being performed manually that could be automated?

For example, checking if the number of observations has changed, if missing values are being introduced, if values fall within a certain range.

## Efficiency

### Repetition

* Are there any tasks that are being done repeatedly that could be simplified or streamlined?

For example, is the same line of code used multiple times with only minor changes?

* Could you use functions, macros, or loops to reduce the repetition in the code?
* Is the code using pre-specified functions for all tasks where that is possible?

For example, using post-estimation commands to calculate confidence intervals instead or manually coding them.

* Do inputs have a single representation in the code?

For example, the list of outcome and control variables is defined in a single place and referred to using a function, macro, or object whenever needed?

### Automation

* Are any manual steps required to run the code? Could they be automated?
* Are outputs such as tables and figures exported through code to an accessible format (csv, tex, png, svg)?
* Are outputs formatted using code, so they are ready for publication once exported?

For example, are all the necessary labels and notes added in the code? Are the number of decimals, font sizes and other formatting automated?

## Reproducibility

* Can you run the code from start to end? If not, can you tell why?
* Is the code easy to run? Is it possible to simplify the transfer from one computer to another?

Ideally, the code should run on a new computer without any manual steps.

* Did you have to install any dependencies (packages, drivers)? If so, could this be automated? If not, how would you suggest that the author communicate that these steps are necessary?
* Were file paths easy to modify? Did you have to modify them in many different lines of the code? If so, how could this process be simplified?
* If the code is creating any outputs (such as data sets, tables, and figures), are you able to recreate them? Are they the same as in the original code?
* Are there any random processes being used in the code? If so, are they reproducible?

Sampling, treatment assignment and bootstrapping are examples of common tasks that involve random processes.

## Good things

* Did you learn about any new functions or packages reading this code?
* Are there any practices you would like to also use in your work?
* Was any task implemented in a more efficient or simpler way than you would have done?

## Content-specific items

### Reviewing data sets

* Is the data uniquely and fully identified?
* Is it clear what the unit of observation in the data is?
* Is the data saved in a file format that is suited for the software used?
* Is there suitable data documentation to undestand the contents of the data?
* Is the file encoding suitable to the data?
* Are all variables using the most efficient data type for the type of information they contain?

For example, in R, categorical variables are stored as factors and logical variables are stored as booleans. In Stata, categorical variables are stored as labeled integers and the data set is compressed.

* Are data sets stored in formats that are native to the software used?

For example, .rds and .rda for R and .dta for Stata.

* Could operations be simplified by using a tidy data set as opposed to long or wide?
* Are categorical variables correctly labeled?
* Are categorical variables consistently labeled?
* If two different variables have the same categories, do they have the same levels and labels?
* Are codes for missing values interpreted as missing?

In survey data, for example, it is common to use codes like -88 or 9999 to represent missing values. In Stata, extended missing values should be used when applicable.

* Is there a system in place to track changes to the data?

### Reviewing data analysis code

* Are categorical variables treated as categorical, not continuous?
* Is there a system in place to track changes to tables and figures?
* Can you easily identify which parts of the code create each different table or figure?
* Do tables and graphs contain detailed notes explaining the methods, such that exhibits can be read and understood on their own?
* Do the methods described in the notes correspond to those implemented in the code?
* Are output tables are [easy to interpret](https://dimewiki.worldbank.org/Checklist:_Submit_Table)
* Are output graphs are [easy to interpret](https://dimewiki.worldbank.org/Checklist:_Reviewing_Graphs)