
Make Your Code Great Again!

Hauke Sonnenberg^{*1}, Michael Rustler¹, and Christoph Sprenger¹

¹Kompetenzzentrum Wasser Berlin gGmbH – Germany

Abstract

What is good code? What is bad code? What happened to good code that turned out to be bad? And if it's bad, what can we do to make it good, or even, great again? We will outline Best Practices for good and maintainable source code writing.

A couple of analytical R-functions will be presented that aim at assessing code quality and disclosing weaknesses of existing R code. One function identifies duplicates of function definitions. It extracts the bodies of functions with identical name and helps the user compare their differences. Duplicated functions indicate that code has been copied and pasted. Exact duplicates can be removed and the function can be defined at a single place, e.g. in a package. Another function identifies string constants. It is used to find hard-coded file paths. File paths can be defined in a redundant-free way to make them easily adaptable to changing locations of scripts or input files. We propose a method to define proper paths. A third function looks for package dependencies and logs the most frequently used packages. Ultimately, a script summary is generated that calculates metrics supposed to be possible indicators for the quality of code: e.g. number of rows/functions per file, number of expressions per function, number of for-loops, if-else-statements, etc.

With our presentation, we would like to discuss about R-code quality and maintainability and about making code great again.

Keywords: clean code, best practice, dry, code analysis, duplicates, metrics, reproducibility

^{*}Speaker