5Rs Comparison

Luke Fairley

March 2022

1 Re-runnable

- Python: 4/5. Most version information was given, but more could have been provided.
- C++: 4/5. Most version information was given, perhaps more could be given.
- R: 4/5. Rcpp. C++ and R version all given in package description.

2 Repeatable

- All 4/5. All algorithms are deterministic, so repeat-ability is limited only by re-runnability.

3 Reproducible

- Python and C++ 3/5. Both include plenty of information about the environment in which the code was originally run, but this information is not exhaustive. Both also include unit tests of some sort, although these are not enforced at run-time. In the case of the C++, this unit test is only in the jupyter notebook, and not the .cpp file. We also don't provide a fixed output file for more rigorous testing. The use of version control was also a bit loose for these two projects.
- R: 3/5. Version information for C++, R, and Rcpp are included in the package description. The package is publicly hosted on GitHub, and contact details are provided. However, as my university email address is given, this may expire at some point. The example code provided in the package *could* serve a dual use as a unit-test, but is not automatic and is by no means exhaustive. An R equivalent of the *isStable* function from the Python notebook could be used to check validity of results from the main algorithm.

4 Reusable

- Python and C++ 3/5. Both versions provide a decent, but not exhaustive, amount of documentation and description, and reasoning is provided as to why the objects such as preference tables and matchings are represented in the way that they are.
- R 2.5/5. Plenty of documentation, plus commentary in the raw code, is provided. However, the documentation is not as thorough as it is for Python and C++, lacking explanation as to why objects are represented in a certain way. To an extent, the "sweeping under the rug" of the specifics of C++ implementation, and the

simple representation of preference tables, may actually make the package more approachable to those less familiar with C++. However, this may not be thorough enough for those more interested. A link to the C++ForPythonProgrammers repository is given for those who are interested, but it does require the user to have C++ incorporated into a jupyter notebook.

5 Replicable

- All 2.5/5. All 3 implementations follow the same original algorithm, which fully describes how the algorithm works in the abstract. Implementations also don't use too many additional libraries, but their version numbers are given where needed. However, it's fair to say that none of the implementations provide documentation specific enough that any researcher with that documentation could fully replicate (give or take some details and allowing room for incremental changes in platform over time) the source code. Data structures are discussed sufficiently, and the code does have comments and discussion, but full design documentation and pseudo-code is not provided.