Translating Older R Data Manipulation Workflows to the Modern Equivalents

Devin Pastoor, Vijay Ivaturi

1Center for Translational Medicine, School of Pharmacy, University of Maryland

Objectives: In the past 2 years, an explosion of new packages have been developed for use in the R ecosystem. These packages build on the new tooling such as leveraging C++, continuous build systems, and the author’s previous experiences to introduce faster, easier to use, and better tested packages well-suited for incorporation into a pharmacometrician’s workflow. The objectives of this work were to provide a reference guide to facilitate translation of old scripts to the new equivalents, and provide a set of examples to introduce the capabilities these packages provide.

Methods: Three model datasets and one data assembly challenge were used to demonstrate how the datasets could be manipulated to perform common tasks such as calculation of summary statistics, conversion from long-to-wide format, and preparation for modeling or visualization softwares. Each task was accomplished using base R and older packages such as `plyr` and `reshape2`, then again using their modern counterparts `dplyr`, `tidyr` and `stringr`.

Results: Use of `dplyr`, `tidyr`, and `stringr` provide simpler API’s to allow for more readable code, with the added benefit of running up to 100’s of time faster than their older counterparts. The packages do not map one-to-one in usage, as the older packages often play many roles (eg. reshaping and manipulating at the same time), whereas the modern versions take the approach of each handling one role (eg. `dplyr` for data manipulation, `tidyr` for reshaping). Given their focus on performing one task well, the new packages can be more quickly updated, better qualified, and more easily understood.

Conclusions: It was demonstrated that a traditional analysis workflow in R using packages such as `reshape2` and `plyr` can be quickly and easily updated to use `dplyr` and `tidyr`. This improves the raw analysis performance, increases code legibility, and improves the overall flow of data through the entire analysis pipeline. The tasks presented are representative of common data manipulation objectives, and the code presented can be easily adapted across a variety of additional situations common in pharmacometrician’s workflows.