Why I heart (not) parentheses, a journeyman's toolkit path from S to R

Colin Goodall[1,*]

1. AT&T Labs - Research
*Contact author: cgoodall@att.com

Keywords: "data model" "standalone tools" jt Perl Python

The data model introduced in S [1], which includes vectors (no scalars), multi-dimensional arrays, lists, data frames, functions as objects, and metadata in object attributes, is clearly one of the most successful and long-lived in data analysis. The introduction of open source R, r-project.org [2], and CRAN, set in motion the pervasive use of the R software environment for data analysis and statistical computation we participate in today. Indeed, the early limitations of S and R, notably in the analysis of very large data sets, in highly interactive graphics, and in providing an integrated collaborative development environment, are even now being effectively addressed and overcome.

A different approach to computational data analysis is to shed R (for now) as a software environment. Instead, computation is by discrete lightweight standalone tools in a shell environment. Data objects sit in distinct files instead of as R objects. UNIX tools for system administration, such as sort, cat, paste, and join, are a starting point in assembling a large set of standalone executables for data analysis, the journeyman's toolkit. The functions are programmed in Perl, Python, Korn shell, and Cymbal [3]. The specific language used is not important, except that each language supports some constructs more easily than does R, which in turn influences how a data analysis task is designed. For example, in Perl, each array may contain data of different types (as in, rows of a data frame in R), text manipulation is primary, numerical analysis is secondary, and hierarchical hashes and arrays are very general.

The data model is at core that of S and R, and the journeyman's toolkit has analogs of `apply`, `aggregate`, `reshape`, `duplicated`, `print`, `subset`, `seq`, etc., as one would expect. However, compared to their R counterparts these standalone functions typically do more operations (with one or more versions possibly of the base function), accept more complex inputs, and offer more output options. For example, duplicated can return duplicated values in one or more variables conditional on a set of conditioning variables (`tapply` anyone?). Or print will return formatted output with multi-row column headers, empty lines for spacing, and exception formatting for values with unusual width.

To take this a step further, the toolkit contains novel handling of metadata (for example multi-valued names attributes), list-valued fields (support for multiple field separators in the same record), analysis of arbitrarily large data files (including parallelization opportunities), and mini-languages (used for example in building Excel spreadsheets from flat files).

I describe an R package **jt** which brings these modes of standalone analysis back into R.

References

[1] Becker, R.A., Chambers, J.M., and Wilks, A.R. (1988). The New S Language. Wadsworth and Brooks/Cole, Pacific Grove CA.

[2] R Project (…2014). The R Project for Statistical Computing, http://www.r-project.org

[3] Greer, R (1999). Daytona and the Fourth-Generation Language Cymbal. In *Proceedings ACM SIGMOD Intl Conf on Management of Data*, **28**(2): 525-526.