The Lish: a data model to support analysis by end user programmers

Alan Hall
The Open University, UK
alan.hall@open.ac.uk

Abstract
For end user programmers needing to carry out data analysis, the spreadsheet is an attractive choice, but has little safety net against user errors. Reducing these errors is an active research area, but one aspect rather little investigated is the role played by the underlying data model: the grid of cells. I am working on an alternative model, the "lish", based on nested lists of cells. Its theoretical advantages include fewer and more concise formulae, and easier updates to the structure. A user study is in preparation to assess its practical utility.

1. Introduction
The professional analyst needing to process tabular data has a number of options, including dedicated data science languages such as R, and libraries for general purpose languages, such as the SciPy library for Python. But for the end user seeking an interactive alternative, requiring little if any use of code, the most widespread choice is the spreadsheet. Its directness and immediacy make it accessible to non-programmers, but come at a price: spreadsheets are notorious for errors (EUSPRIG, 2018).

This situation presents a conundrum for researchers: it is tempting to try to augment the spreadsheet with features that might encourage users to adopt a more disciplined approach, but in so doing we risk losing the low barrier to entry and ease of use that made it so attractive in the first place. McCutchen et al (2016) take an alternative angle, arguing convincingly that a major flaw with the spreadsheet is that the grid of cells is simply the wrong structure for many of its use cases. It is broadly this line of attack that I pursue in my research.

If the grid of cells is the wrong structure, what is the right one? I am investigating the potential of a structure based on nested lists of cells, as the underlying data model for representing spreadsheet-like data. Like Miller & Hermans (2016) in their "gradual restructuring" work, I am seeking to allow users to capture additional structure where it would be helpful, while continuing to work in a spreadsheet-like environment. Another guiding principle is that of "tidy" data (Wickham, 2014). The new data model aims to be "tidy" on the inside, while giving the user the flexibility of layout they would associate with a spreadsheet.

2. Approach

2.1 The Lish data model
We have submitted a paper to the main conference track which gives more details of the model, called the "lish". It is based on lists of cells, which may be nested so as to contain further lists; the first element of each list has a privileged status, forming a template which provides a minimum structure for subsequent elements.

The nested nature of the model has some obvious uses in capturing natural hierarchies within the data, and in permitting formulae to have multi-cellular results, whose size may not be known at design-time. The way the templates interact with the recursive structure causes some further useful properties to emerge. It enables systems of tables that have a repeating pattern to be guaranteed to repeat consistently. It can also avoid unnecessary formula replication, because the user can define a single formula in a template to apply to a range of other cells. In a normal spreadsheet, changing a *value* updates dependent values; in a lish, changing a *structure* updates dependent structures.

Templates are composed of ordinary cells and lists – they are not a separate abstraction. Hence they support the spreadsheet-like form of programming by example, where a user may define a calculation for a specific instance and then, by making it a template, expand it to a more general case.

2.2 Research questions

My top level question is “What are the pros and cons of expressing spreadsheet-like tabular data in a nested list-of-cell form, as opposed to conventional grid form?” Some relevant sub-questions are: “Does the nested form accord with users' mental model of their data” and “How does the nested form affect users' workflow when conducting analysis”?

3. Current status

My initial work was to define the lish as a data structure and develop algorithms to operate on it. Because a lish has more constraints than an ordinary list of lists, these algorithms have to accommodate “action at a distance”, where modifying one part of the structure causes a corresponding modification elsewhere. An example would be inserting a column in a table, which might cause a separate column to be inserted in one or more related tables.

The second main piece of work was to define a “lish calculus”: a set of rules for performing arithmetic and functional transformations upon lishes. This drew heavily upon the vectorised arithmetic of the R programming language (R Core Team, 2018). The lish calculus allows calculations involving many, possibly non-adjacent, cells to be defined using a single concise formula; the structure as laid out in the templates is used by the machine to deduce which cells are to be operated upon.

I have also built a small prototype editor, which allows the user to enter and edit a lish and build formulae in a somewhat spreadsheet-like way.

4. Forthcoming work

Are the more “structured” data actually easier to work with? Are the more “concise” formulae actually more intuitive to use? These questions can only be answered empirically, so my main outstanding piece of work is a user study.

Participants will be government analysts, who are frequent spreadsheet users. They will be asked to build a specified model in lish form, and this will be followed by a semi-structured interview. The interview is intended to shed light on how well (or poorly) the nested form corresponds to users' mental model of their data. It will also elicit users' perceptions of the costs and benefits of structuring data into lish form, with regard to their everyday workflow. And it will seek to identify any system behaviour that was surprising to the user.

5. Conclusion

I have developed the “lish” on the hypothesis that it maps more closely to users' mental models of their data than does a spreadsheet grid, and hence could facilitate building and maintaining end user analyses with tabular data. The planned user study will help to assess whether these advantages can be realised in practice.

References

EuSpRIG (2018) Spreadsheet mistakes - news stories. Available at: http://www.eusprig.org/horror-stories.htm

McCutchen, M., Itzhaky, S., and Jackson, D. (2016) Object spreadsheets: a new computational model for end-user development of data-centric web applications. Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, 112–127.

Miller, G. and Hermans, F. (2016) Gradual structuring in the spreadsheet paradigm. Proceedings of the 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 240-241.

R Core Team (2018) R Language Definition v 3.5.0, section 3.3, “Elementary arithmetic operations”. Available at: https://cran.r-project.org/doc/manuals/r-release/R-lang.html

Wickham, H. (2014) Tidy data. Journal of Statistical Software, 59(10).