Journal of Statistical Software

June 2020, Volume 94, Book Review 1. *doi: 10.18637/jss.v094.b01*

Reviewer: James E. Helmreich
Marist College

Data Science with Julia

Paul D. McNicholas, Peter A. Tait
Chapman & Hall/CRC Press, Boca Raton, 2019.
ISBN 978-1-138-49998-0. 220 pp. USD 149.95 (H), 52.18 (P).
`http://github.com/paTait/dswj`

Introduction

The Julia language (Bezanson, Edelman, Karpinski, and Shah 2017) is relatively new, and as Charles Bouveyron says in the forward to *Data Science with Julia*, is designed "for efficient and parallel numerical computing while keeping a high level of human readability." The authors Paul McNicholas and Peter Tait do not intend this book to be a course text, but to help those who want to learn the Julia language and apply it to data science problems. Coding in Julia is somewhat similar to R (R Core Team 2020) and Python (Van Rossum and Drake Jr 1995). Knowledge of R or Python is not required to use *Data Science with Julia*, but McNicholas and Tait try to highlight these similarities, with R especially. That may well be, but I judge that Julia should not be the first language one encounters.

Structure

There are seven chapters to the book, starting with an introduction to data science, some discussion of Julia, and descriptions of the data sets that will be used for examples. The introductory discussion of the field of data science is quite nice. The authors provide some historical perspective going all the way back to 1998, highlighting the newness of the field conceived as separate from statistics. There is no discussion of choosing an IDE (integrated development environment), and while that choice is a personal one some guidance would have been appreciated. We are given a brief description of the main data sets that are used in the text, as well as a few well chosen graphics in full color.

In the second chapter we are introduced to the basic syntax, types, and dataframes. Standard programming structures such as looping, conditional execution etc. are well done. There is a good section on writing functions and their use. Chapter 3 continues the basics of working with dataframes. Two large code chunks provide good examples of cleaning large dataframes; there is a discussion of manipulation of the data structures that allows querying several data sources using syntax similar to SQL.

In Chapter 4 we get into data visualization. There are several plotting packages available for Julia. McNicholas and Tait use the package **Gadfly** (Jones *et al.* 2018), which follows Wilkinson's *Grammar of Graphics* (Wilkinson 2005) and is quite similar to Wickham's R package **ggplot2** (Wickham 2016). We are shown a sprinkling of standard plots (scatter, boxplot, facet plots etc.) in full color, but a quick internet search will bring up many good Julia graphics that go beyond these fairly standard plots.

Chapters 5 and 6 introduce the reader to supervised and unsupervised learning techniques. For each technique there is a careful exposition of the mathematics of the various procedures covered, followed by detailed code chunks showing their implementation. While I strongly recommend typing the code in as an exercise in (personal) learning, most of the code is available at the GitHub site for the text. Unfortunately, many file names are somewhat opaque, and the alphabetical listing is not in order of appearance in the book. In a short introductory text, not all data science techniques will or could be covered, but there are examples of some of the more frequently used methods. These include k-nearest neighbors classification, classification and regression trees, principle components analysis and probabilistic PCA, as well as clustering and dimension reduction. The final chapter covers techniques to import R dataframes, as well as calling R from within Julia. Examples include additional methods in data reduction and random forests.

Discussion

The Julia language is indeed similar to R and Python, though I find it a bit more finicky than R. The text provides a good introduction to the language, and the diligent student who types in their own code will find it fairly straightforward to get going with Julia. As always, working with your own data in similar ways will reinforce the material. The code for the various learning techniques is informative, and provides the reader with a nice jumping off point for other examples of their own. The authors argue strongly for certain structural techniques that enable Julia to take better advantage of multiple cores and speed up evaluations. One or two timed example comparisons might have been informative but were not included.

Unfortunately, there were some major editing and melding issues. Perhaps the biggest is their use of a large kaggle dataframe on beer types and characteristics. In the first chapter, these `beer` data are discussed, and presented as cleaned and ready to use. In the third chapter however, these same `df_recipe` data are anything but cleaned and ready for investigation (and yes, there is a different name for the dataframe). There is a long block of code that performs import and cleaning operations that is very informative, so I am not sure why the pre-cleaned `beer` version from Chapter 1 is needed. But it gets more confusing: in Chapter 4 on Data Visualization, we are back to the `beer` dataframe, as well as a related but undiscussed dataframe `beer1`. Indeed, at one point (pages 116–118) the narrative discusses `beer` but the code uses `df_recipe`. This is more problematic than simply a name change as the non-informative column names in `beer` (e.g. "`c3`") are different from those used in `df_recipe`. Clearly this is a problem of two authors not melding their individual chapters together properly into one narrative. It is not an isolated occurrence. In the second chapter they create a simulated dataframe called `df1`. This dataframe is used throughout that chapter, but in later chapters it is called `df_1` where from the context it is a similar but somewhat larger (in both rows and columns) dataframe. This makes those latter chapter usages a bit difficult to follow.

In the later chapters on supervised and unsupervised learning, other omissions mar the presentation. For example, several helper functions are presented and defined, and then there is code (page 100) for a k-nearest neighbors classification. This latter code uses a helper function (`cityblock`) which is not defined and so breaks the code if you try to run it yourself. This is not an isolated problem: a few pages later (page 118) another helper function is left undefined (`N_array`). These are oversights, perhaps not critical, but emblematic of lack of attention to detail. This sort of thing occurs in other places as well, for example on page 82 they use and plot a datafrme `df_ecgf` that is never discussed or defined; on page 144 a block of code calls to `include("chp6_ppca_functions.jl")` which should be included among the code at their GitHub site but is not. A file `chp6_ppca1.jl` exists and is what the authors need.

In other cases, material is simply out of order. For instance, a code block in Chapter 6 (page 134) works well if you have obtained the `crabs.csv` dataframe already. It is fairly easy to save it from R, then load it using methods discussed in Chapter 3. More problematically, later in the same chapter (page 149) we also need to import `x2.rda`, where Chapter 3 methods fail. Reading on though, the next chapter, Chapter 7, leads off with methods to import the crabs dataframe from the **RDatasets.jl** package, as well as methods for other example dataframes `x2, coffee, wine`. It seems an unfortunate choice of the order to do things in.

One finds a number of smaller annoyances. For instance, early on the authors show that (some) of the Greek alphabet can be used in the REPL (Julia console), but neglect to show precisely how. It takes a bit of exploration to realize that `\mu<tab>` will produce the μ symbol. Another: code for graphic 4.23 is inexplicably omitted. And another: on multiple occasions when running code the authors neglect to include the first `using Package1, ...` line that explicitly details the packages that must be loaded for the commands used. Of course, Julia will throw an error, but you still have to figure out which package the commands came from. As an example, in one small block McNicholas and Tait generate some random values with the `Pareto` distribution, but neglect to include `using Distributions`, which is where `Pareto` lives. Continuing: page 36 "In Julia, function names are all lowercase, without underscores, ...", yet on page 39: "For clarity ... we use a `_` in the function name..." Presumably formal Julia functions do not use the underscore, but user-defined ones are fine. The graphics package used, **Gadfly**, by default will not show the plot in the plot window. There is no mention of this quirk by McNicholas and Tait, and I had to stumble upon a fix at the package's website `https://gadflyjl.org/stable`. In sum, there were more than a reasonable number of frustrations in working through the text that the authors could have been more careful about. None of these issues strike me as fatal, but are certainly irksome. I would hope an errata page could be added to the book's website.

Some frustrations were not the fault of the authors. Most significantly, the help files for functions were often anything but helpful. Many did not include code examples. Frequently not all optional arguments would be listed. Very frequently I had to opt for the web to obtain correct syntax. But this is probably more a symptom of a young language that has not had the kinks worked out than of serious flaws. Additionally, there were annoyingly frequent warnings about deprecated functions, almost always not explicitly used in the code that was run.

Conclusion

It is clear to me that Julia is an important tool for the data scientist, and could well become a predominant language in the field. Tests show it is for some applications at least an order of magnitude faster than R. The book under review here, *Data Science with Julia*, though flawed is a reasonable introduction to working with Julia. It will reward your efforts, and provide a good jumping off point for real work with the language. The graphics package **Gadfly** McNicholas and Tait use is very good, and shows the plotting utilites for the language to be far from rudimentary. The authors give examples of calling R from within Julia, but it is also possible to call Julia from within R, see Chambers (2016). Thus the cutting edge techniques of R as the lingua franca of working statisticians can usefully be combined with the speed of Julia for working with the large data sets with which data scientists must routinely contend.

References

Bezanson J, Edelman A, Karpinski S, Shah VB (2017). "Julia: A Fresh Approach to Numerical Computing." *SIAM Review*, **59**(1), 65–98. doi:10.1137/141000671.

Chambers JM (2016). *Extending R*. The R Series. Chapman & Hall/CRC, Boca Raton.

Jones DC, *et al.* (2018). "GiovineItalia/Gadfly.jl: v0.7.0." doi:10.5281/zenodo.1284282.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Van Rossum G, Drake Jr FL (1995). *Python Reference Manual*. Centrum voor Wiskunde en Informatica Amsterdam. URL https://docs.python.org/2.0/ref/ref.html.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Wilkinson L (2005). *The Grammar of Graphics*. 2nd edition. Springer-Verlag.

Reviewer:

James E. Helmreich
Marist College
Department of Mathematics
3399 North Road
Poughkeepsie, NY 12601, United States of America
E-mail: James.Helmreich@Marist.edu
URL: http://foxweb.marist.edu/users/james.helmreich/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
June 2020, Volume 94, Book Review 1 *Published:* 2020-06-30
doi:10.18637/jss.v094.b01