PhUSE US Connect 2018

Paper AB04

R or Python: A Programmer's Response

Jiangtang Hu, d-Wise, Morrisville, NC, USA

ABSTRACT

We hear lots of requests on porting R to Clinical Computing Platform besides SAS. Before R really touches the ground, I'd like to share some thoughts on why R is a not a good choice, considering there is a strong alternative, Python.

In this paper, I will compare the strength and weakness of R and Python, on various tasks like data manipulation, statistical analysis, system integration. Although R is well known in our industry and has some good things, I argue that Python is a better fit:

1. Python is a better designed programming language;
2. For various programming tasks, Python offers more consistent and unified packages stack, while in R, the packages are scattered;
3. Python is better at system integration due to its reputation as glue language;
4. Considering its elegant syntax and unified ecosystem, Python is easier to learn for SAS programmers.

Introduction

It's not a language war between R and Python. The argument was settled long ago in programmer world. The requests on R are mostly from statisticians inside clinical development units. Statisticians are not programmers; they might prefer R for some statistical tasks, like sample size estimate and visualization. But for heavy users for the Clinical Computing Platform, aka, clinical/statistical programmers, they need good language(s) to import data, clean data, transform data and do analysis and reporting. In this type of programming, I'd argue Python is far way better than R, from a programmer's perspective.

Language Design

Python is simply a better language, created by real computer scientists. R was mainly created by statisticians and for statisticians. There are lots of design drawbacks in R core language. I'm not cherry-picking the downsides of R; instead, it's fundamental and it's well known. Ross Ihaka, one of the creators of R language, listed two main problems of the R system [1]: non-optimized and inherently slow interpreter, call-by-value semantics that copies large data objects multiple times. For example, the design matrix is copied 6 times during the fitting process in a linear model [2]. The suggestion by Ross Ihaka, is to develop a new language!

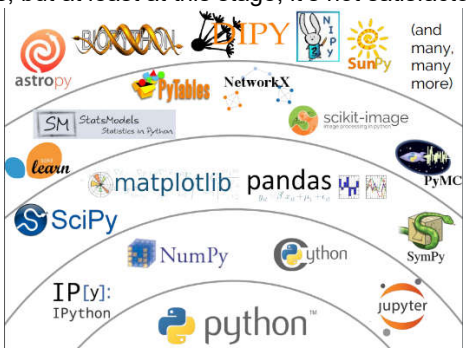
There are lots of attempt to address these problems. For example, *renjin*, is a JVM based R interpreter (developed by Alexander Bertram), aiming for better performance [3]. Microsoft reimplemented the so called enhanced R distribution: *Microsoft R Open* for improved performance [4]. Another attempt is from TIBCO, *TIBCO Enterprise Runtime for R (TERR)* [5]. The problem is, currently, no single R-core alternative is the obvious winner. Most R users are still using the default R interpreter.

To be fair, Python is not flawless language. Actually, 10 years ago, the Python core developers were so overwhelming to improve the Python 2; they decided to create a new branch, Python 3. Now the much better Python 3 is the main stream among Python users, and Python 2 will retire in 2020 [6].

Ecosystem

Although facing fundamental drawbacks in core design, R enjoys huge success due to its user contributed packages.

For programming tasks mentioned above, Python offers more consistent and unified packages stack (Numpy, Pandas, Scikit-learn, etc), while in R, the packages are scattered. It's true that there is an attempt to gather several core R packages to unify the process, but at least at this stage, it's not satisfactory.



In the Python's data science stack [7] above, from bottom to top, there are the essential, well designed, well documented and well maintained (by a large community) Python packages for data science tasks:

1. Ipython and Jupyter are the interactive programming environments
2. NumPy and Scipy are basically the matrix and high-dimensional array computation packages
3. Pandas is the main workhorse to manage data frame
4. Matplotlib is the visualization package
5. StatsModles is the traditional statistical analysis package
6. Scikit-learn is the machine learning package

I have no doubt that R can also perform the similar functionalities. The problem of R ecosystem is, there are no such centralized packages. Usually R packages are small, developed and maintained by a very small group of people. The most famous R packages, I argue, are mostly single-man's work, like Tidyverse by Hadley Wickham [8]. Here I didn't infer that single-man's work is low quality; instead, I just want to assert that compared to Python ecosystem, R is not as mature.

System Integration

Python is a general-purpose programming language. It supports object-oriented, functional, procedural and such multiple programming paradigms. Besides data sciences, Python is also popular in fields like web frameworks, multimedia, databases, networking, automation, system administration. Considering this rich feature, I think it's rather easy to argue Python is far way better than R at system integration, with current components residing in Clinical Computing Platform.

And again, I didn't say R can't do such things. But having the capacities is one thing, R nowadays is still mainly a language of statistical analysis and graphics.

Back to the Future

Python is the first-class citizen in the world of big data/artificial intelligence/machine learning/deep learning packages. Every related software ports to Python first due to its huge user base. In recent years, Rstudio company tries to port Spark [9], Tensorflow [10] and other popular packages to R.

In this sense, Python will make a happier programmer. Python will introduce SAS programmers to a wider world of machine learning, deep learning which all cool kids are talking about.

CONCLUSION

I might show strong bias toward Python. But not every software shares the same weight. In clinical world, people talks about R, most because statisticians learned R at their graduate schools. Outside academia, Python as a better software is well established. In the future clinical computation platform, I bet Python a good position.

REFERENCES (HEADER 1)

[1] Towards a New Statistical Computing System by Ross Ihaka, https://www.stat.auckland.ac.nz/~ihaka/downloads/New-System.pdf
[2] The R Project: A Brief History and Thoughts About the Future by Ross Ihaka, https://www.stat.auckland.ac.nz/~ihaka/downloads/Massey.pdf
[3] http://www.renjin.org
[4] Microsoft R Open, https://mran.microsoft.com/open
[5] TIBCO Enterprise Runtime for R (TERR) https://community.tibco.com/products/terr
[6] https://pythonclock.org/
[7] Python's Data Science Stack by Jake VanderPlas, https://speakerdeck.com/jakevdp/pythons-data-science-stack-jsm-2016
[8] https://www.tidyverse.org/
[9] http://spark.rstudio.com
[10] https://tensorflow.rstudio.com/

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Jiangtang Hu
d-Wise
1500 Perimeter Park Dr., Suite 150
Morrisville, NC 27560
Work Phone: 919.334.6096
Email: Jiangtang.hu@d-Wise.com
Web: http://jiangtanghu.com/