Combining R and Python for scientific computing

Felipe Ortega1, *, Javier M. Moguerza1 and Emilio L. Cano1

1. Universidad Rey Juan Carlos
*Contact author: felipe.ortega@urjc.es

Keywords: Scientific computing, Python, Django, lxml, programming interfaces

Over the past decade, Python has become one of the most popular programming languages for scientific computing. The large number of available libraries to extend its features, including NumPy, SciPy, as well as Pandas (for data mining) have led Python to play a central role in numerous scientific projects requiring support for mathematical and statistical operations. In the same way, the R programming language has evolved to become today a mature and solid resource for statistical computing, specially regarding the availability of hundreds of contributed packages to extend its core functionalities.

However, in many occasions developers need to combine the advantages of these two powerful programming languages to address complex challenges in data science projects. On the one hand, Python offers more flexible and intuitive alternatives to implement code for data retrieval, parsing and preparation. On the other hand, once the data are ready for the analysis, R exhibits a richer collection of packages that already implement most of the statistical and data visualization features required in practical studies. In addition to this, sometimes it is required to provide a convenient and structured way to access data and results from these analyses through a structured and simple API, so that they can be imported by other software systems that will reuse them for different purposes, or will display them on third-party GUIs.

Hence, the purpose of this presentation is to provide a summary of current practical strategies to combine R and Python code in data analysis, following a structured, maintainable and pragmatic approach. Concrete examples taken from different application domains, including software engineering, the study of open online communities and optimization in energy systems, will illustrate these strategies. Furthermore, this presentation will put special emphasis in two specific Python projects that can be of great utility for scientific computing developers:

- lxml lxml Development team (2013), one of the most advanced and feature-rich libraries for parsing XML code in Python.
- Django Django Software Foundation (2013), a high-level web development framework in Python that is specially suitable for fast and maintainable implementation of REST APIs Fielding (2000).

References

