



Follow

518K Followers



From 'R vs Python' to 'R and Python'

Leveraging the best of both 'Python and R' in a single project.

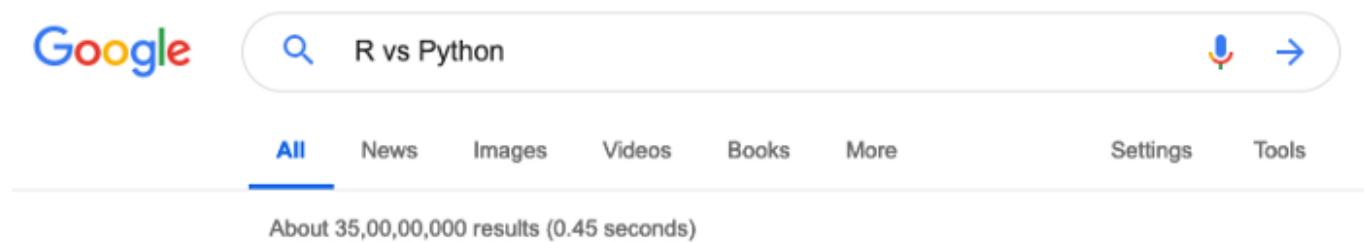


Parul Pandey Mar 7, 2019 · 7 min read ★



Focus should be on the Skills , not on the Tools

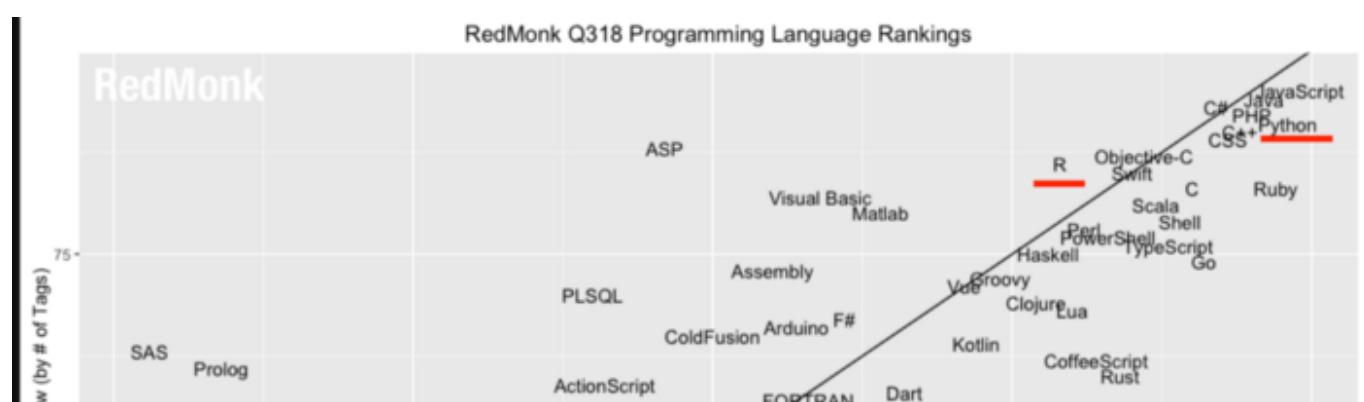
If you are into Data Science, the two programming languages that might immediately come to mind are **R** and **Python**. However, instead of considering them as two options, more often than not, we end up comparing the two. R and Python, are excellent tools in their own right but are very often conceived as rivals. If you type `R vs Python`, in your Google search bar, you instantly get a plethora of resources on topics which talk about the supremacy of one over the other.

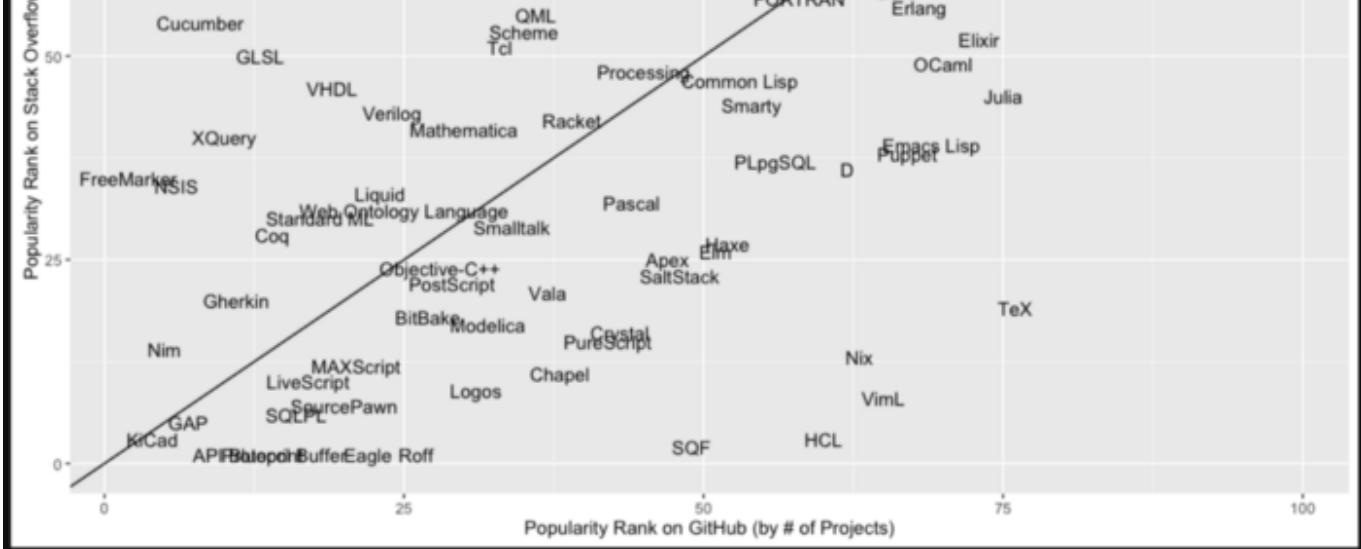


One of the reasons for such an outlook is because people have divided the Data Science field into camps based on the choice of the programming language they use. There is an **R camp** and a **Python camp** and history is a testimony to the fact that camps cannot live in harmony. Members of both the camps fervently believe that their choice of language is superior to the other. So, in a way, divergence doesn't lie with the tools but with the people using those tools

Why not use Both?

There are people in the Data Science community who are using both Python and R, but their percentage is small. On the other hand, there are a lot of people who are committed to only one programming language but wished they had access to some of the capabilities of their adversary. For instance, R users sometimes yearn for the object-oriented capacities that are native to Python and similarly, some Python users long for the wide range of the statistical distributions that are available within R.





The figure above shows the results of the survey conducted by [Red Monk](#) in the third quarter of 2018. These results are based on the popularity of the languages on **Stack Overflow** as well as on **Github** and clearly show that both R and Python are rated quite high. Therefore, there is no inherent reason as to why we cannot work with both of them on the same project. Our ultimate goal should be to do better analytics and derive better insights and choice of a programming language should not be a hindrance in achieving that.

Overview of R and Python

Let's have a look at the various aspects of these languages and what's good and not so good about them.

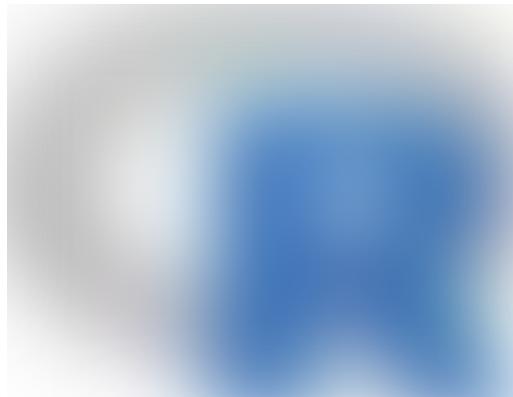
Python

Since its release in 1991, Python has been extremely popular and is widely used in data processing. Some of the reasons for its wide popularity are:

- Object-oriented language
- General Purpose

- Has a lot of extensions and incredible community support
- Simple and easy to understand and learn
- packages like pandas, numpy and scikit-learn, make Python an excellent choice for machine learning activities.

However, Python doesn't have specialized packages for statistical computing, unlike R.



R

R's first release came in 1995 and since then it has gone on to become one of the most used tools for data science in the industry.

- Consists of packages for almost any statistical application one can think of. CRAN currently hosts more than 10k packages.
- Comes equipped with excellent visualization libraries like ggplot2.
- Capable of standalone analyses

Performance wise R is not the fastest language and can be a memory glutton sometimes when dealing with large datasets.

Leveraging the best of Both Worlds

Could we utilize the statistical prowess of R along with the programming capabilities of Python? Well, when we can easily embed SQL code within either R or Python script, why not blend R and Python together?

There are basically two **approaches** by which we can use both Python and R side by side in a single project.

R within Python

- PypeR

PypeR provides a simple way to access R from Python through pipes. PypeR is also included in Python's Package Index which provides a more convenient way for installation. PypeR is especially useful when there is no need for frequent interactive data transfers between Python and R. By running R through pipe, the Python program gains flexibility in sub-process controls, memory control, and portability across popular operating system platforms, including Windows, GNU Linux and Mac OS



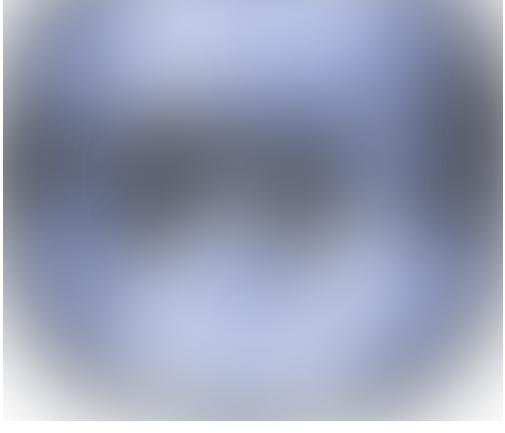
Conventions for conversion of Python objects to R objects

- pyRserve

pyRserve uses Rserve as an RPC connection gateway. Through such a connection, variables can be set in R from Python, and also R-functions can be called remotely. R objects are exposed as instances of Python-implemented classes, with R functions as bound methods to those objects in a number of cases.

- rpy2

rpy2 runs embedded R in a Python process. It creates a framework that can translate Python objects into R objects, pass them into R functions, and convert R output back into Python objects. rpy2 is used more often since it is one which is being actively developed.



One advantage of using R within Python is that we would able to use R's awesome packages like ggplot2, tidyr, dplyr et al easily in Python. As an example let's see how we can easily use ggplot2 for mapping in Python.

- **Basic Plot**

<https://rpy2.github.io/doc/latest/html/graphics.html#plot>

- **Geometry**

<https://rpy2.github.io/doc/latest/html/graphics.html#geometry>

Resources

You may want to have a look at the following resources for more in-depth review of **rpy2**:

- [rpy2's Official Documentation](#)
- [RPy2: Combining the Power of R + Python for Data Science](#)
- [Accessing R from Python using RPy2](#)

Python within R

We can run R scripts in Python by using one of the alternatives below:

- [rJython](#)

This package implements an interface to Python via Jython. It is intended for other packages to be able to embed python code along with R.

- [rPython](#)

rPython is again a Package Allowing R to Call Python. It makes it possible to run Python code, make function calls, assign and retrieve variables, etc. from R.

- [SnakeCharmR](#)

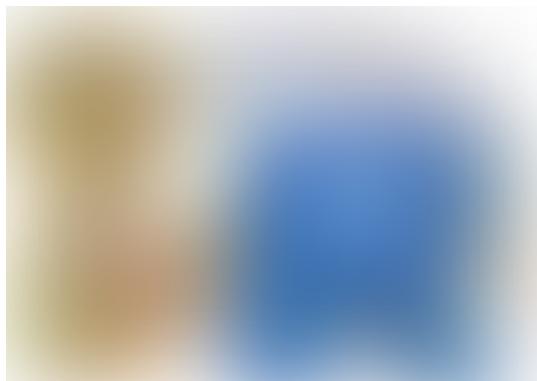
SnakeCharmR is a modern overhauled version of rPython. It is a fork from ‘rPython’ which uses ‘jsonlite’ and has a lot of improvements over rPython.

- [PythonInR](#)

PythonInR makes accessing Python from within R very easy by providing functions to interact with Python from within R.

- reticulate

The **reticulate** package provides a comprehensive set of tools for interoperability between Python and R. Out of all the above alternatives, this one is the most widely used, more so because it is being aggressively developed by Rstudio. Reticulate embeds a Python session within the R session, enabling seamless, high-performance interoperability. The package enables you to *reticulate* Python code into R, creating a new breed of a project that weaves together the two languages.



The reticulate package provides the following facilities:

- Calling Python from R in a variety of ways including R Markdown, sourcing Python scripts, importing Python modules, and using Python interactively within an R session.
- Translation between R and Python objects (for example, between R and Pandas data frames, or between R matrices and NumPy arrays).
- Flexible binding to different versions of Python including virtual environments and Conda environments.

Resources

Some great resources on using the **reticulate** package are:

- The Documentation is pretty robust and has a lot of examples and use cases to help you get started.
- <https://longhowlam.wordpress.com/2017/04/10/test-driving-python-integration-in-r-using-the-reticulate-package/>
- SNAKES IN A PACKAGE: COMBINING PYTHON AND R WITH RETICULATE

Conclusion

Both R and Python are quite robust languages and either one of them is actually sufficient to carry out the Data Analysis task. However, there are definitely some high and low points for both of them and if we could utilize the strengths of both, we could end up doing a much better job. Either way, having knowledge of both will make us more flexible thereby increasing our chances of being able to work in multiple environments.

Reference

The article is an adaptation of this [excellent talk](#) by Andrew Collier which he gave at Pydata Berlin in 2018.

<http://blog.yhat.com/tutorials/rpy2-combing-the-power-of-r-and-python.html>

Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)

Get this newsletter

Emails will be sent to pritalagala@gmail.com.

[Not you?](#)

[Python](#) [R](#) [Data Science](#) [Programming Languages](#) [Towards Data Science](#)

[About](#) [Help](#) [Legal](#)

