**Between R and Python, Which Language Would I Suggest to an Aspiring Data Analytics Expert?**

In recent years I’ve seen the evolution of the use of both of these languages in the world of Data Analytics. Here are my thoughts.



(image from Unsplash)

Until now I have always avoided entering the tussle of those interested in the R vs Python battle. But recently, my peer data geek Brian Julius asked my opinion on the Python vs R debate in his LinkedIn post here:

**[Brian Julius on LinkedIn: #r #python #datascience #dataanalysis | 37 comments](https://www.linkedin.com/feed/update/urn:li:activity:6996829718033399808/" \t "_blank)**

[I've long been a conscientious objector in the R vs. Python wars, taking the view what's "best" depends on your…](https://www.linkedin.com/feed/update/urn:li:activity:6996829718033399808/" \t "_blank)

[www.linkedin.com](https://www.linkedin.com/feed/update/urn:li:activity:6996829718033399808/" \t "_blank)

I tried to answer in a comment of the post, but the text was too long 🙂 So, I decided to write a post instead.

**My thoughts about Python and R**

In recent years I have been able to observe the evolution of the use of both of these languages in the world of Data Analytics. So, I have come to the following personal conclusions:

✔ The R language is much more straightforward for *interactive data analysis* and *data exploration*, especially for analysts or those who come from the world of business intelligence, where SQL holds sway. Transforming the data with R is very reminiscent of the mental process done by those using SQL, with the advantage of being able to use specific functions that simplify complex transformations (such as data pivoting, for example), or that apply statistical operations useful for analysis.  
The approach taken by Python to transform data is more related to a programmer’s experience. Having to necessarily resort to *lambda expressions*, for example, for rather basic data manipulation tasks, disorients any analyst used to a more set-based approach (which is the right way to think when working with data!) and makes us realize that those who developed the Python packages needed for data wrangling were primarily developers rather than analysts.

✔ R is the language par excellence used in academia (Statistics, Mathematics, Data Science, and so on). It is therefore very likely to find new data science algorithms implemented directly in R, even before finding them implemented in Python. Therefore, if you need to use these new algorithms for a project, you must necessarily use R.

✔ As far as *data visualization* is concerned, R is the main tool that makes you able to produce beautiful graphics ready for professional publications. Certainly, graphs of this type are also obtainable in Python, but not as easily as you can make them in R, especially with the myriad of packages developed to add specific graphical needs.

✔ As for *professional dashboarding*, the most widely used platforms have evolved and admit the use of both languages. For example, both Plotly Dash and Shiny allow the development of enterprise-level data apps in Python and R. For those connected to the world of Microsoft’s data platform, Power BI also allows you to use both languages.

✔ Python is a very clear, general purpose programming language, very versatile due to the ecosystem of packages developed by the community, and predominantly used by programming students and developers. For this reason, it is easier to find a programmer who knows Python who also wants to go into data analytics. That is why there are so many packages dedicated to data transformation and data analysis now available in Python. As a result, the market for technologies related to Data Analysis has veered conspicuously toward Python adoption.

✔ On the whole *data engineering* side, the clear winner is certainly Python. First of all, Python is one of the few languages supported by all the APIs that allow interacting with the data services from all major cloud providers (Azure, AWS, Google). Moreover, the need to integrate the use of heterogeneous data solutions into a production architecture on the cloud pushes towards the use of Docker containers, which make life easier for those who have to engineer all this. One of the languages used to interact with Docker is Python (R is not among them).

✔ Due to the fact that all the data engineering part is handled in Python, it is often convenient to develop *machine learning models* always using Python (you can develop them just as well in R). In this way, they are easily released into production via specific pipelines developed in Python. It’s true that it is possible to integrate specific steps into Python pipelines that invoke R scripts, but often having everything in Python is more convenient and maintainable by a group of data engineers skilled in Python.

✔ *Big data* platforms, now mostly based on *Spark*, allow data transformation or machine learning pipelines to be developed in PySpark, a Python API specifically designed for Spark. It’s also possible to interact with Spark through the SparkR language (an R API for Spark). The problem is that SparkR isn’t always implemented in these platforms out-of-the-box. What’s more, SparkR doesn’t implement the integration of *dplyr* (the main package used for data transformation in R) with Spark, which is done by *sparklyr* instead. But the aforementioned data platforms often don’t support sparklyr natively. So, PySpark, along with *SparkSQL* (the SQL API for Spark), is widely used as the ultimate “big data language”.

✔ Python is the necessary choice when you need to tackle projects that require *Deep Learning* solutions (computer vision, automatic spoken language recognition, natural language processing, audio recognition using TensorFlow, Keras and PyTorch frameworks). Implicitly, we are stating that the whole world of Artificial Intelligence is heavily based on Python.

That said, I try to answer the question asked in the title of the article.

**Conclusions**

Nowadays, Python allows you to deal with pure Data Analysis and Data Science topics as well as Data Engineering and Artificial Intelligence thanks to the vast ecosystem of packages developed by the community. It’s true that specific problems related more to Statistics and Data Visualization are more easily solved using the R language, but these are still specific cases that make up a very small percentage of the cases for which Python is mostly required.

Therefore, although I myself am more fluent in R than Python,

I highly recommend that those who want to tackle Data Analytics topics devote themselves primarily to learning Python and how to best use the most important packages for data transformation and machine learning (pandas, scikit-learn, etc.).

The versatility of Python allows you to possibly switch from one job role to another (e.g., from Data Scientist to Data Engineer, or vice versa), should your interests change during your career.

This doesn’t take away from the fact that, once you have delved into Python, studying to have a basic understanding of R and its packages of the Tidyverse ecosystem is definitely a plus that will prove very useful in so many cases. You should not underestimate that the R community is large and that many solutions have been developed in R. Adding the knowledge of this language in your arsenal is therefore a winning choice. Also because, contrary to what some may think, Python and R are no longer two compartmentalized worlds. There are bilingual IDEs that allow you to develop using both languages. One example is *RStudio*, as shown in this article: