ASSIGNMENT

Submitted To,

Mr. Sijo Thomas

Deapartment of MCA

Marian college Kuttikkanam

Submitted By,

Jojo joseph

17PMC227

MCA(L)

INTRODUCTION

R and Python are both open-source languages used in a wide range of data analysis fields. Their main difference is that R has traditionally been geared towards statistical analysis, while Python is more generalist. Both comprise a large collection of packages for specific tasks and have a growing community that offers support online. All in all, the Python code could easily be translated into R and was comparable in length and simplicity between the two languages. While Python's syntax is inherently cleaner/ tidier, we can use packages that implement piping in R and achieve similar results (even though Python's dot-separated syntax is still much easier to type than using the piping operator of magrittr). For plotting and visualisation I still think that R's ggplot2 is top of the line in both syntax, customizability and outcome (admittedly, I don't know matplotlib as well as ggplot)! In terms of functionality, I couldn't find major differences between the two languages and I would say they both have their merits. For me, R comes more natural as that is what I'm more fluent in, but I can see why Python holds an appeal too and I think I'll make more of an effort to use both languages in my future projects. R and Python are both open-source languages used in a wide range of data analysis fields. Their main difference is that R has traditionally been geared towards statistical analysis, while Python is more generalist. Both comprise a large collection of packages for specific tasks and have a growing community that offers support and tutorials online. Python and R have all been used successfully in teaching college students fundamentals of mathematics & statistics. In today's data driven environment, the study of data through big data analytics is very powerful, especially for the purpose of decision making and using data statistically in this data rich environment. MatLab can be used to teach introductory mathematics such as calculus and statistics. Both Python and R can be used to make decisions involving big data. On the one hand, Python is perfect for teaching introductory statistics in a data rich environment. On the other hand, while R is a little more involved, there are many customizable programs that can make somewhat involved decisions in the context of prepackaged, preprogrammed statistical analysis.

Introducing R

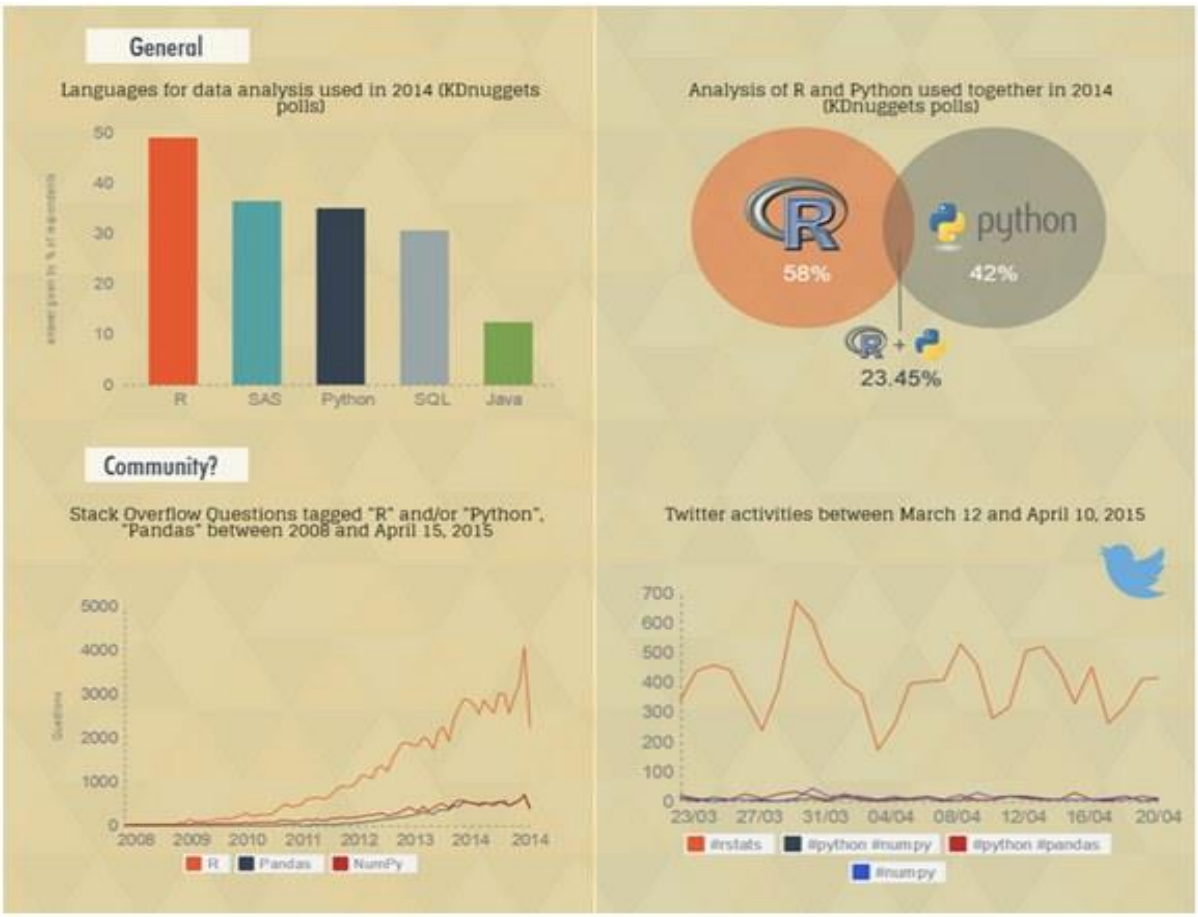
Ross Ihaka and Robert Gentleman created the open-source language R in 1995 as an implementation of the S programming language. The purpose was to develop a language that focused on delivering a better and more user-friendly way to do data analysis, statistics and graphical models. At first, R was primarily used in academics and research, but lately the enterprise world is discovering R as well. This makes R one of the fastest growing statistical languages in the corporate world. One of the main strengths of R is its huge community that provides support through mailing lists, user-contributed documentation and a very active Stack Overflow group. There is also CRAN, a huge repository of curated R packages to which users can easily contribute. These packages are a collection of R functions and data that make it easy to immediately get access to the latest techniques and functionalities without needing to develop everything from scratch yourself. To end, if you're an experienced programmer, you probably won't have a hard time to get up to speed with R. As a beginner, however, you might find yourself struggling with the steep learning curve. Luckily, there are many great learning resources you can consult nowadays.

Introducing Python

Python was created by Guido Van Rossem in 1991 and emphasizes productivity and code readability. Programmers that want to delve into data analysis or apply statistical techniques are some of the main users of Python for statistical purposes. The closer you get to working in an engineering environment, the more likely it is you might prefer Python. It's a flexible language that is great to do something novel, and given its focus on readability and simplicity, its learning curve is relatively low. Similar to R, Python has packages as well. PyPi is the Python Package index and consists of libraries to which users can contribute. Just like R, Python has a great community but it is a bit more scattered, since it's a general purpose language. Nevertheless, Python for data science is rapidly claiming a more dominant position in the Python universe: the expectations are growing and more innovative data science applications will see their origin here.

R and Python: The Data Science Numbers

If you look at recent polls that focus on programming languages used for data analysis, R often is a clear winner. If you focus specifically on Python and R's data analysis community, a similar pattern appears. There is a growing group of individuals using a combination of both languages when appropriate.



R: Pros and Cons

Pro: A picture says more than a thousands words

Visualized data can often be understood more efficiently and effectively than the raw numbers alone. R and visualization are a perfect match. Some must-see visualization packages are ggplot2, ggvis, googleVis and rCharts.

Pro: R ecosystem

R has a rich ecosystem of cutting-edge packages and active community. Packages are available at CRAN, BioConductor and Github. You can search through all R packages at Rdocumentation.

Pro: R lingua franca of data science

R is developed by statisticians for statisticians. They can communicate ideas and concepts through R code and packages, you don't necessarily need a computer science background to get started. Furthermore, it is increasingly adopted outside of academia.

Pro/Con: R is slow

R was developed to make the life of statisticians easier, not the life of your computer. Although R can be experienced as slow due to poorly written code, there are multiple packages to improve R's performance: pqR, renjin and FastR, Riposte and many more.

Con: R has a steep learning curve

R's learning curve is non-trivial, especially if you come from a GUI for your statistical analysis. Even finding packages can be time consuming if you're not familiar with it.

Python: Pros and Cons

Pro: IPython Notebook

The IPython Notebook makes it easier to work with Python and data. You can easily share notebooks with colleagues, without having them to install anything. This drastically reduces the overhead of organizing code, output and notes files. This will allow you to spend more time doing real work.

Pro: A general purpose language

Python is a general purpose language that is easy and intuitive. This gives it a relatively flat learning curve, and it increases the speed at which you can write a program. In short, you need less time to code and you have more time to play around with it! Furthermore, the Python testing framework is a built-in, low-barrier-to-entry testing framework that encourages good test coverage. This guarantees your code is reusable and dependable.

Pro: A multi purpose language

Python brings people with different backgrounds together. As a common, easy to understand language that is known by programmers and that can easily be learnt by statisticians, you can build a single tool that integrates with every part of your workflow.

Pro/Con: Visualizations

Visualizations are an important criteria when choosing data analysis software. Although Python has some nice visualization libraries, such as Seaborn, Bokeh and Pygal, there are maybe too many options to choose from. Moreover, compared to R, visualizations are usually more convoluted, and the results are not always so pleasing to the eye.

Con: Python is a challenger

Python is a challenger to R. It does not offer an alternative to the hundreds of essential R packages.

R vs Python for Data Science: Comparing on Parameters

►R vs. Python: Usability

R and Python are ranked amongst the most popular languages for data analysis, and both have their individual supporters and opponents. Python is widely admired for being a general-purpose language and comes with a syntax that is easy-to-understand. R, on the other hand, is developed keeping statisticians in mind, therefore more specific and has field-specific advantages such as great features for data visualization. R packages for a wide variety of statistical tasks using the CRAN task view; covers everything from Psychometrics to Genetics to Finance. R is good for statistics-heavy projects and one-time dives into a dataset. Take, for example, text analysis, where you want to deconstruct paragraphs into words or phrases and then identify patterns, R is the best choice. Python is more commonly used to build modules to create websites, interact with a variety of databases, and manage users. When drawing a comparison between Python and R, Python is better for building analytical tools. This is especially true if you are creating a web service to enable other people to upload datasets and find outliers.

►R vs. Python: Libraries

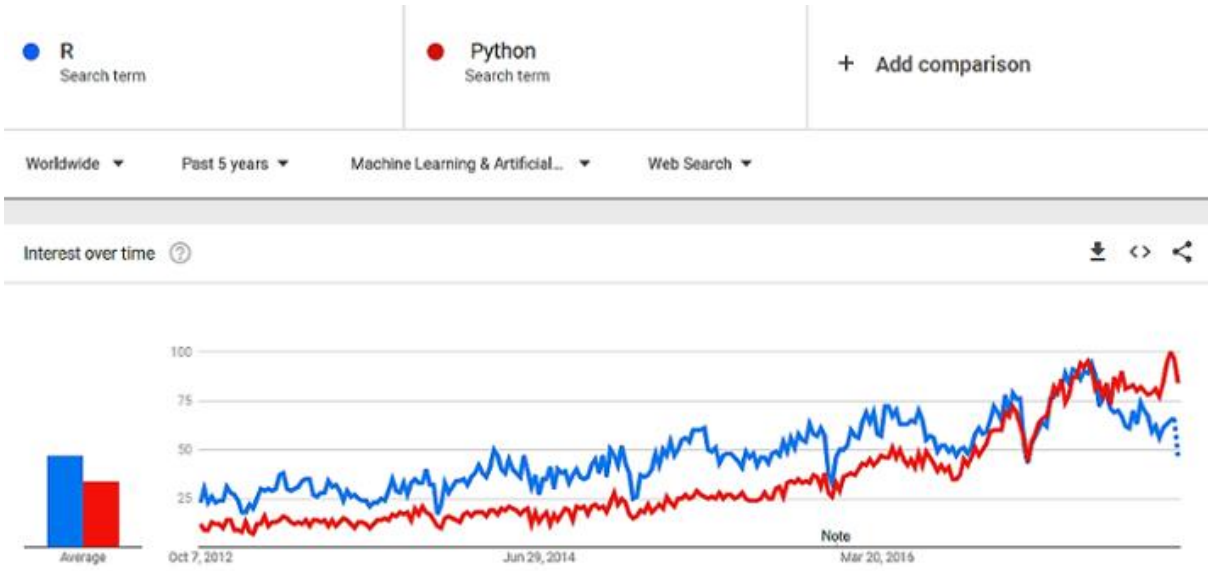
Both Python and R come with sophisticated data analysis and machine learning packages to can give you a good start. Each has its own analysis, visualization, machine learning and data manipulation packages. The same applies to IDEs. RStudio IDE is the obvious choice for working in an R development environment. R packages like dplyr, plyr and data.table are highly preferred for manipulating packages, stringr for string manipulation, ggvis and ggplot2 for data visualization, and caret for machine learning. Python, on the other hand, comes with more number of development environments. Spyder, IPython Notebook, and Rodeo are good to start with. As for popular libraries, Python gives you numerous options to choose from; NumPy /SciPy for scientific computing, matplotlib to make graphs, scikit-learn for machine learning and pandas for data manipulation.

►R vs. Python: Flexibility

Being a data analyst, one often needs to take a call; to choose Python or R, for better business value. While both languages have their own share of merits, the question of flexibility worries me a lot. R is a specific programming language is meant for complicated data analysis, comes with several packages, and is available for implementing and statistical tests and models. But the solutions must be customized and not general. Python being a general programming language comes with many libraries that are used for statistical work. It is also good for integration options and more streamlined approach to practicing novel tasks. You may write your own code for scripting a website or any web app. However, if you

would ask me which language is more suitable for approaching a data science project, my answer would be R.

►Popularity



Python is a lot more popular than R. This is primarily due to the wide-scale usability of Python in comparison to R. Python can be used for many different purposes from web development to app development to data science. R, on the other hand, is made for core statistical analysis. Being a niche player it obviously less popular. But when it comes to the landscape of data science R competes net to net with Python. According to Payscale, the average salary of R data scientist is $88,409 while that of Python is $96,616. This is because many corporates prefer Python because of is all-purpose use. That said the difference can be easily covered if the person is highly skilled in the language he chooses.

►Ease of Learning

R is the language for academicians and statisticians while Python is an all-purpose language usually preferred by programmers (people from the field of engineering and computer science). R has a very steep learning, it is difficult to start with for non-programmers while Python has a more gradual learning curve. Both languages have good documentations, courses, and books available and their communities are well committed to driving growth to their respective languages.

►Visualizations

Data scientists frequently plot data to find correlations and patterns. Thus, visualizations become important criteria while choosing a data science tool. Python data visualization libraries include Seaborn, Bokeh, and Pygal, while that of R include ggplot2, ggvis, googleVis, and rCharts. In terms of

visuals, R is way ahead of Python. R delivers stunning visuals which are much more sophisticated than the convoluted visualizations of Python.

►Usage

R is used the most when data analysis tasks require standalone computing or individual servers. Python is a glue language, therefore, it is generally used when data analysis tasks require integration with web applications or when a piece of statistical code needs to be inserted into a production database.

►Tasks

R wins hands down when it comes to performing exploratory statistical analyses. It is considered to be easy for beginners. Statistical models can be written with few lines of code. Python as a full-fledged programming language can be a great tool to deploy algorithms for production use.

►Data Handling Capabilities

R is handy when it comes to a multitude of packages for both coders and non-coders to not only perform statistical tests but also to create machine learning models. Python has had its own challenges related to data analysis. However, after the introduction of NumPy, Pandas, and a few others, it has started gaining a lot of popularity in the field of data analytics, as well.

►R vs. Python for data science: Usage

When it comes to usage in data science, experts are divided in their opinions. Some data scientists prefer R to Python because of its visualization libraries and interactive style. R comes with great abilities in data visualization, both static and interactive. Interactive visualization built with R packages like Plotly, Highcharter, Dygraphs, and Ggiraph take the interaction between the users and the data to a new level. But again, if you are looking for higher performance or structured code Python is the go-to language. It is because Python has some of the best libraries such as SciKit-Learn, IPython, numpy, scipy, matplotlib, etc. NumPy is the foundational library for scientific computing in Python, and it introduces objects for multi-dimensional arrays and matrices, as well as routines that allow developers to perform advanced mathematical and statistical functions on those arrays with fewer codes. Matplotlib is the standard Python library for creating 2D plots and graphs. Both Python and R have their individual merits. So, if you are a newbie working on a data science project, then I would advise you to use both R and Python interchangeably.

►Python vs. R for Data Science: Lingua Franca

We have arrived at an age when a data scientist is not always somebody with a computer science background, nor is he a mathematician. More often, a data scientist is an innovator or visionary, whose futuristic approach goes beyond the barriers of academia. R is the data scientist's best instrument. R codes and packages are great for communicating ideas and concepts. R is the lingua franca for data science projects today. To work in a Python development environment, one should ideally have a computer science or programming background. Learning about the different Python libraries like SciKit-Learn, IPython, numpy, scipy, matplotlib, are best for people with a coding background.

►R vs. Python: Learning Curve

Which is of the two programming languages is a better choice for learning? It is a common question that baffles many aspiring data analysts. Both R and Python, require a significant time investment, and one needs to have a thorough knowledge of either, for a promising career in data science.When making a comparison between R and Python, I may say that R has a steep learning curve, and people without prior programming experience may find it difficult to grasp at the beginning. However, with extensive learning and practice programs, you may have a strong command over R. Python, on the other hand, which focuses on readability and simplicity, is generally considered easier for programmers to pick up. Python being a more general programming language is useful building a website or making sense of command-line tools, especially for those with a background in statistics.

►R vs. Python: Community

A programming language becomes known by its usage, and yes, by its users. The richer community a language has, the better are its chance of growth and sustenance. It is because people do not just write codes, they discuss, analyze, and geniuses, we know also dream about codes. A quick look any of the language communities will give you an inkling of the goings on in the minds of these master coders. Any regular observer would know that R as a language has a rich community of more than 2 million users and that includes thousands of developers spread across the world. The community has packages widespread across actuarial analysis, finance, machine learning, web technologies, pharmaceuticals that can be of great help to predict component failure times, analyze genomic sequences, and optimize portfolios. user-generated documentation of active StackOverflow members has contributed greatly to the rapid adoption rate of R. Python community, though slightly less powerful, is also gaining acceptance of the good number of StackOverflow members. General-purpose coding in Python continues to grow with remarkable user-contributed code and documentation by developers and programmers, data scientists, researchers, and students across the world.

►R vs. Python: Licensing

When drawing a comparison between Python vs R for Data Science, one must not overlook the part on licensing. Most libraries used for Python have business-friendly distribution licenses, such as BSD or MIT that makes sharing of the software much easier. Both MIT and BSD are simple and permissive licenses, which allow people to use and distribute your code subject to with a few restrictions; the license must always be distributed with the code. R libraries, on the other hand, are GPL or CC0, which makes distribution norms slightly stricter. The chief concern with GPL-2 and GPL-3 is distribution. Both GPL-2 and GPL-3 are "copy-left" licenses. So, anyone who distributes your code in a bundle must license the whole bundle in a GPL-compatible way. Also, to distribute modified versions of your code (derivative works) the source code must also be made available. GPL-3 is a little stricter than GPL-2. So, if you are looking for easier distribution licenses, I would say Python is a simpler option.

►R vs. Python: The Winner

In the recent past, Python and R have been outdoing each other, when it comes to programming and application for Analytics, Data Science, and Machine Learning. Most of the common tasks which could be executed earlier in either of the two are now executable by both. To make a choice between R and Python you need to depend completely on the use case and abilities. If you are from a statistical background then I would advise you to start with R. On the contrary, if you are an experienced programmer, choose Python for proficiency. At times, the level of analysis and development needed becomes a deciding factor. R is an obvious choice when you have a hardcore data science requirement. Python, on the other hand, is a better alternative for application development. Thus, the best solution is to make a smart move based on the domain needs, resource availability, and cost. Both the languages are equally good. Each language has its pros and cons for different scenarios and tasks. The bottom line here is that it is difficult to place one before the other, Python or R. So, if you have already mastered Python and gained a few years of experience, you may also learn R, for more knowledge. Learning both is always a boon for a career in data science. You may also enroll for a data analytics course for more lucrative career options in Data Science.

CONCLUSION

We see the market slightly bending towards Python in today's scenario. It will be pre-mature to place bets on what will prevail, given the dynamic nature of industry. Depending on your circumstances (career stage, financials etc.) you can add your own weights and come up with what might be suitable for you. On the web, you can find many numbers comparing the adoption and popularity of R and Python. The main reason for this is that you will find R only in a data science environment; As a general purpose language, Python, on the other hand, is widely used in many fields, such as web development. The choice between R and Python depends completely on the use case and abilities. If you are from a statistical background than it is better to start with R. On the contrary, if you are from computer science than it is better to choose Python. In case of business, the choice should depend on the individual use case and availability. If data scientists of one language are more easily available to you than it is better to go with the favorable option. The choice can also vary based on the level of analysis and development needed. If the need is for hardcore data science than R comes out as a better alternative while Python is the apt solution for application development based on data science. These are the languages and libraries that have proved to be extremely useful in various data science use cases. Keep in mind, that the choice of programming language and the libraries that you will use, depends on specific tasks, so it's beneficial to know what are the strong and weak sides of each of them. Indeed, this list is not complete, many other valuable tools can and have to be examined, but it will definitely be a good starting point for your journey into data science industry.