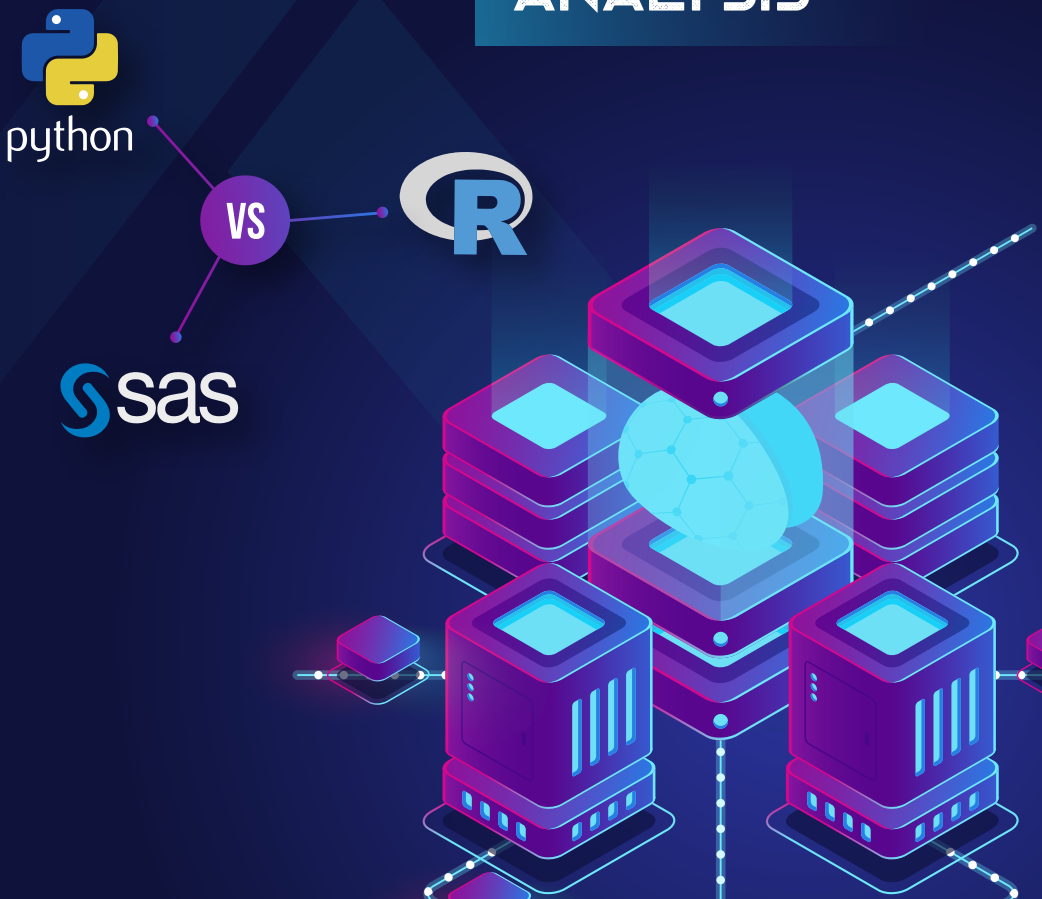
ANALYTIX LABS

THE BEST MACHINE LEARNING TOOLS:

PYTHON VS R VS SAS

A DETAILED ANALYSIS



1. TOOLS FOR HANDLING DATA

The saying 'you are as good as your tools' is true for all the people belonging to various professions and it is most certainly holds true for the people who are working in the field of Data Science. With the various kinds of data in place which can be differentiated through their format, sensitivity or sheer size requires specialized tools to deal with them. For many years, the bulk of data during the nascent stages of Data Science was made up of pre-dominantly Survey data and later Banking and other Financial data. In order to deal with these kind of data which was highly sensitive and not much in quantity, tools were created in order to respond to the requirement. These data solving data requirements were either made by research based institutions or banking and financial organizations. However, as the domains and the size of data generation expanded along with the requirement to analyze it, new tools were created in order to respond to the new data driven landscape.

> You are as good as your Tools.

Till now, the most used tools (apart from various spreadsheet based tools such as IBM's Lotus 1-2-3 and Microsoft's Excel) in the field of Data Analysis or Data Science are SQL, SPSS, SAS, MATLAB, R and Python. In today's day and age, 3 tools are required to be assessed and scrutinized which are also among the most widely used tools, they are SAS, R and Python. With the constant changes happening in the realm of data, it become imperative and of utmost importance to find which tool is the best or to put it more coherently, which tool should be used where, when and for whom it's worth is. Still, we can try to find out that, by and large and of course as of date, has won the great battle of data science tools.

> Which tool should be used where, when and for whom it's worth is.

2. ROUND 1:

PROPRIETARY STATISTICAL SOFTWARE V/S OPEN SOURCE FREE PROGRAMS

The various tools that can be used to deal with the diverse forms of data are nothing but software's. These software are different from each other in the way they deal and process the data and especially the language the user has to know in order to operate them. However, from the point of you of adopting any tool by an organization, the most important and basic difference between them can be if the tool is open source or closed source. Close Source software are Proprietary software which has the common characteristics of being not free to use and having strict control over the inner working of the software through the retention of intellectual property rights which are mainly the source codes and the copyrights. Initially, such software which had data handling capabilities were heavily relied upon by various institutions and organizations. Among these software were the famous **Matlab, SPSS, Stata and SAS.**

On the other hand, we have Open Source software which along with other characteristics, are free and most importantly can be modified (i.e. the source code is not retained by the creator) and can also be shared (as there are are little or no copyright restrcitions by the user) which in turn leads to collaboration, community formation and participation in creating various version of the software often enhancing the capabilities of the software. This democratization of the use of a particular software leads to its rapid and far reaching adoption across various domains and communities.

> R and Python are open source, free to download

In the battle between SAS, R and Python, the first casualty is of SAS and the reason lies in the fact that it is greatly and exclusively controlled by a group. For a long period of time SAS was the industry's standard tool for working on Data Science projects especially in the space of enterprise analytics. It all started to change with the advent of R. The first switch happened with companies deploying R as their standard data science tool kit in place of SAS. To comprehend why this happened, we first need to understand the fundamental difference between SAS and R which is that while software such as R and Python are open source, free to download, available for use for everyone, **software such as SAS, MATLAB, SPSS etc. are expensive commercial suites.** This fundamental difference manifests itself in various aspects of these software and we first need to understand them one by one.

2.1. BACKGROUND

SAS stands for Statistical Analysis System. This software was designed keeping in mind the industry's requirements so that large datasets can be handled. Its aim was always to capture the big, multi-billion, multi-national companies that required Data Analysis. Obviously, among the first users were the **BFSI (Banking, Financial Services and Insurance)** companies and later companies engaged in Human Capital Management.

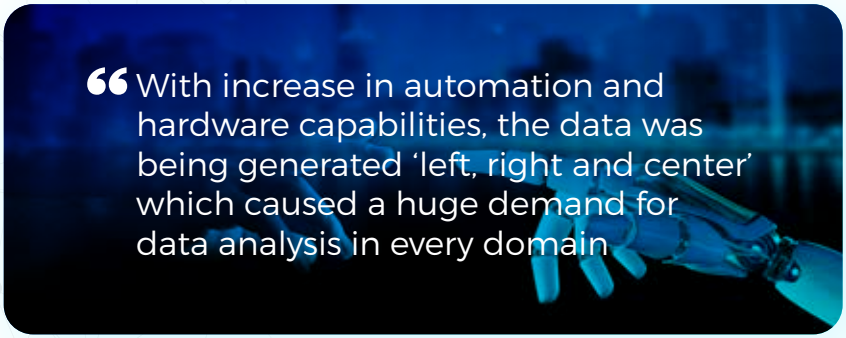
This lead to the introduction of various vertical products such as the SAS Financial Management (erstwhile CFO Vision) and SAS Human Capital Management (erstwhile HR Vision). Not only such companies but many government institutions also started using SAS because of its high level of reliability, accuracy and scientific backing. This caused administrative units such as FDA (United States Food and Drug Administration (a federal agency for control and supervision of food and other health related products) started examining new drug applications by standardizing themselves on SAS/PH-Clinical which was a SAS component for pharmaceutical users.

On the other hand, R and Python are programming languages that are free and open source. Any statistical software based on them had no backing of any corporation which made many agencies anxious to adopt them. Also, because of these software were open sourced, they heavily relied on community or 3rd party created packages in order to solve specialized domain related task (in reference to the above examples: such as Financial Management, Human Capital Management or Pharmaceutics). Because of this, the workings were not standardized and not vetted by any answerable agency, which further caused reluctance among the users working in sensitive domains.

> This software was designed keeping in mind the industry's requirements so that large datasets can be handled.

2.2. COST

Things started to change, especially when the need of analyzing data and create products based on data science began to reach domains that were not essentially multi-billion dollar companies dealing with large or highly sensitive data. Now, with increase in automation and hardware capabilities, the data was being generated 'left, right and center' which caused a huge demand for data analysis in every domain, so much so, that the task of analyzing data was actively being outsourced to small and medium level companies for whom the cost of acquiring a tool was of paramount importance.



The foremost and biggest jolt that SAS received was because of this very reason, cost. SAS was and remains to be an expensive software. Where R is absolutely free, SAS charges subscription fees through which the software can be used albeit only for a limited period of time. Along with this there is the cost of technical support and the fact that there are different versions providing different functionalities with the version providing more or specific functionalities costing significantly higher.

This was in stark contrast to R which was free and has now had a lot many packages to perform certain very peculiar, industry specific tasks. The working of these packages were slowly and gradually were being backed by research papers as R was able to penetrate research institutions which gave sanctity to its working and various packages that it deployed. This lead to building more confidence among the new companies to adopt R as their tools to analyze data and create data driven products out of it.

> “SAS was and remains to be an expensive software. Where R is absolutely free

2.3. SYNTAX AND LEARNING CURVE

There is no doubt that SAS is an easy language. Unlike R and Python, there are **drag and drop GUI,** Pre-determined standard procedures, integration of SQL commands with SQL also providing database access making it useful for professionals having no prior knowledge of programming. This was among the reasons for the early adoption of SAS as a lot many people working in BFSI domains were familiar with SQL as a language and with the SQL queries and adopting SAS was thus easier.

On the other hand, R and Python are tough when compared to SAS. People sometimes are required to have some coding background to properly use them (through it is not necessary). This initially acted as a hurdle in the widespread adoption of R and python. As the community of these languages grew, there were much more people to help with the understanding of these languages. Also, people in this field felt the compulsion to go through the learning curve and leave the ease of learning providing by SAS because this ease of learning comes at a heavy cost which is underutilization of what all can be done on the data and limitations to handle and manage the data.

> “R and Python are tough when compared to SAS. People sometimes are required to have some coding background to properly use them (through it is not necessary).

2.4. SUPPORT AND SECURITY

The stakes for any company performing data analysis become high when sensitive data, multiple clients and high dependence on the outcomes for future course of action is involved. This is where Support is required in order to understanding the inner working of the tool at play and sometimes personal assistance is required for troubleshooting. This was something where SAS had and still has the edge over other software.

SAS has a professional full-fledged technical support, something a multi-billion dollar company would want. It also has an online community which is active and they also have well documented resources on their websites and blogs. Also, SAS has the option of providing servers where data could be safely and securely stored. This helped in further building confidence among the users performing critical data driven operations.

However, the reason people felt confident to switch R or Python is because still all this doesn't come close to the community experience provided by them. Their community is not only huge but is vibrant, diverse and evolving with people from various walks of life- from students to programmers and analysts to academicians, all providing inputs which allows the inflow of multiple ideas to come in and solve a problem. This open source community is the backbone of these programs as they lack the 'proper' support structure provided by SAS, but again, SAS doesn't have an open source community and solely depends on their already established support system. As far as the server services are concerned, R has successfully provided servers for their users, again at a cost which is much less than of R. As far as Python and other languages are concerned, with the advent of cloud based computing and the level of assistance and functionalities provided by cloud computing services such as Amazon Web Services (AWS), Microsoft's Azure, Google Cloud Platform etc. has stripped SAS of the edge it had over other languages. The aspect of security is sometimes work in favor of open source software. As the source code is available to everyone, the new developers can identify errors and other mistakes in the source code which were slipped from the sight of the original developers. The leads to an environment where a software gets constantly improves and gets better and more potent over time.

> with the advent of cloud based computing and the level of assistance and functionalities provided by cloud computing services such as Amazon Web Services (AWS), Microsoft's Azure, Google Cloud Platform etc. has stripped SAS of the edge it had over other languages.

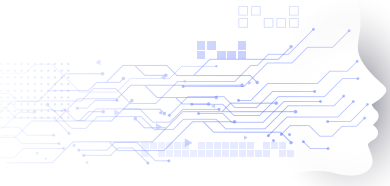
2.5. CAPABILITIES

This is the heart of decision making. The capabilities of any software or language is something that ultimately causes one to decide to which software is to be used. Let us understand one thing first- all the things that SAS can do, can be done in R and Python but the same cannot be said for the other way around. SAS allows sophisticated analysis, high quality professional graphs, array of statistical functions, ready to use GUI, however, it fails to move beyond this and match with the times we live in.

While SAS can create almost publish worthy graphs, it is difficult to create dynamic, complex graphs in SAS.

The standard statistical, mathematical and modeling capabilities that were once provided by SAS are now provided by R and Python too.
When it comes to Machine Learning which at this day and age is much needed, SAS pales in front of R and Python while Deep Learning is out of bounds for SAS and as far as to Big Data is concerned, others undoubtedly come out as a winner. Even though SAS has provided options to use Hadoop, one thing that stops it from going all out is flexibility. SAS is nowhere as flexible as other open source programming languages with its output providing more than necessary results and the user having little control over what and how somethings are to be done.

The above factors now make it understandable how and why the need for a new open source, flexible and cost effective software was created. Even though SAS is still preferred by big financial and marketing companies and is still very much indispensable because many times there is less room for experimentation especially when dealing in critical scenarios with delicate data, still, at the end of the day there are only a limited number of big companies that can afford it. Also, startups and medium-sized companies that deal with Big Data, intend to provide end to end service and analyze data using sophisticated methods such as Machine, Deep Learning require Python and R and is the reason that SAS only holds 3% support from the Data Science community. The switch from SAS to R became visible when Telecom, Tech and Research companies that had to deal with massive amount of unstructured data started switching to R. The final blow came when financial companies that were once considered as the stronghold for SAS begin to switch. It is undeniable that the cost of switching is massive which includes converting a lot of data which is stored in SAS format files (sas7bdat), training new workforce along with setting up new framework's and standard operating procedures but still the fact remains that R took the place that SAS once held for a long period of time (albeit to a certain extent).



> Machine, Deep Learning require Python and R and is the reason that SAS only holds **3%** support from the Data Science community.

3.
ROUND 2: 
BASIC DIFFERENCE BETWEEN R V/S PYTHON

The only thing that is permanent is change itself. R effectively took place of SAS and became the lingua franca for Data Science and this is something that is no more debated and contested, however, it is was a matter of time when it was pushed from its position by Python. Interestingly, the same set of reasons that caused R to take place of SAS has caused Python to dethrone R. Also, the way people resisted the fact that R has taken over SAS and this became a matter of much debate until some years passed and this thing became a fact, a similar debate is ongoing while it is becoming clearer that Python has decisively won the battle or has won some in some key areas.

History

R was created by Ross Ihaka and Robert Gentleman and was created in 1995 and released in 1997. A low-level open source language, R was created to implement the S programming language to ease out the method for creating statistical, graphical and data analysis models. R first was mostly used in the academic and research fields though slowly it spread across the business market and was being used by companies such as Google,

Facebook to financial institutions situated at the Wall Street. It is supported by the R Foundation of Statistical Computing. Python on the other hand was created 4 years earlier, in 1991 by Guido Van Rossem. Python was envisaged as a high level, simple, clean, easy to debug programming language for multiple general purposes. Later libraries began to come up that could perform the functions that were performed by R.

Community and Support

Some can argue that the community and support of R is better than Python with 2 million users, having 125 active online groups with support provided through mailing lists, Stack Overflow groups and the good documentation provided by its supporters. On top of that R has CRAN which is a huge repository which makes the process of user contribution very much centralized. On the other hand it can be argued that the support for Python is scattered with few of the 1600+ user groups being dedicated to data analysis. However, all of this was true until a few years back as now the online support and community for Python is almost at par with that of the R and could overrun it.

Speed

Speed matters a lot especially when it comes to dealing with data. With the recent revolution in hardware and with the spread of IoF (Internet of Things), the amount of data that is being generated has sky rocketed. This has caused the competition between software regarding the speed stiffer.

The answer to which one is faster is not as straight forward as it seems. As some studies show, when it comes to performing loops, R during very high levels of iterations has an edge over python. However, there are other factors at play that contribute towards making R slow when compared to Python. Being a non-standardized language, codes in R can often be written in a poor and inefficient manner making the process slow causing the need of using additional packages such as FastR, pqR, Riposte etc in order to speed up the process. Also, as mentioned earlier, Python is non-sequential which makes the process speedy from the user end. In addition to all this, as R runs solely on RAM, smaller tasks sometimes take a lot more time.

The other aspect is not to compare the core languages but rather the performance of their packages in order to deploy them to perform various data related operations. As far as the major Machine Learning and Deep Learning operations are concerned, Python is decisively faster than R because of the fact that a lot of these packages are written in lower level languages making the processing time faster. Still, when it comes stats based operations, visualization and day to day operations, R sometimes take the lead.

IDE

IDE stands for Integrated Development Environment and forms a very important aspect of any software / language. IDE helps a great deal in the ease of writing, testing and debugging codes. When it comes to the different IDE offered by R and Python then as far as Python is concerned, it provides with a range of IDE (integrated development environment) such as Spyder, Rodeo and IPython Notebook whereas Rstudio is the only commonly used IDE for R. Because of the wide range of IDEs provided by python, it gives more flexibility to the user to decide which environment to work in according to their preference and needs. R on the other hand has only on major Ide which is R studio. Even though R Studio is apt for performing various data related tasks, it doesn't give the kind of options that python provides with its variety of IDEs.
Also, the concept of IDE is getting new definition with the increasing use of Notebooks especially in the field of Data Science in recent times. Notebook's such as Jupyter Notebook have the various debugging and other IDE oriented features with them along with the range of option of adding formatted text, adding multimedia and getting multiple kinds of output from simple text output to graphs etc. This advantage of having more options for documentation helps in sharing the code as they are easy to understand and re-implement. These notebooks especially Jupyter notebook can work with multiple languages, however, so far the most common combination is of Jupyter notebook with

python as the engine or kernel. This leads to all the advantages of Jupyter notebook being assigned to python.
Thus, as Python provides with a range of environments to work in with notebooks such as Jupyter Notebook being easily integrating with it providing useful documentation functionalities that can even use HTML inputs, makes python take the lead in this aspect whereas R only has R Studio whose environment is somewhat close to an IDE of python known as Spyder (Scientific Python Environment) which are decreasingly being used for Data Science projects.

Packages

There is fierce competition between R and Python when it comes to packages. Back in the day it was successfully argued that Python had a limited number of packages with limited capabilities, however today, the number argument still holds true but as far as capabilities are concerned, packages available for Python has outmaneuvered the packages of R. Python has Numpy/SciPy for scientific computing, pandas for data manipulation, matplotlib for graphics, Seaborn to visualize statistical models, scikit-learn for machine learning and TenserFlow and Keras for Deep Learning.

On the other hand R has a very large range of packages that sometimes have very specific capabilities for particular domains such as finance, genetics etc. This strength of R is also its weakness as R provides with so many of packages that finding the right package becomes time consuming coupled with the fact that the packages are inconsistent as they are provided by third parties making the user to slow down the process and understand the methodology from scratch. As Python works on the philosophy that “there should be one-- and preferably only one --obvious way to do it”, the inconsistencies are very limited. For R, the most famous user-contributed packages include caret, dplyr, ggplot and Nnet with caret performing the machine learning and Nnet the deep learning functionalities which are no way as sophisticated as sci-kit-learn, TenserFlow etc. R’s answer to Pandas is dplyr and when compared, it is limited in its usage.

Routine Tasks (Automation)

There is fierce competition between R and Python when it comes to packages. Back in the day it was successfully argued that Python had a limited number of packages with limited capabilities, however today, the number argument still holds true but as far as capabilities are concerned, packages available for Python has outmaneuvered the packages of R. Python has Numpy/SciPy for scientific computing, pandas for data manipulation, matplotlib for graphics, Seaborn to visualize statistical models, scikit-learn for machine learning and TenserFlow and Keras for Deep Learning. On the other hand R has a very large range of packages that sometimes have very specific capabilities for particular domains such as finance, genetics etc. This strength of R is also its weakness as R provides with so

many of packages that finding the right package becomes time consuming coupled with the fact that the packages are inconsistent as they are provided by third parties making the user to slow down the process and understand the methodology from scratch. As Python works on the philosophy that "there should be one-- and preferably only one --obvious way to do it", the inconsistencies are very limited. For R, the most famous user-contributed packages include caret, dplyr, ggplot and Nnet with caret performing the machine learning and Nnet the deep learning functionalities which are no way as sophisticated as sci-kit-learn, TenserFlow etc. R's answer to Pandas is dplyr and when compared, it is limited in its usage.

Licensing

R was able to displace SAS due to the fact that it was cheap and open source, however, ironically enough today when compared to Python, R is not as business-friendly when It comes to the distribution of libraries. Python use MIT or BSD licenses which make sharing easier comparatively to R which uses CC0 and GPL licenses which place more restriction on the distribution of code.

Integration

Even though R is able to integrate with high-level languages such as C, C++ and Java, Python's integration is even better as it is a general purpose scripting language and better supports multiple systems. Python is able to adapt itself depending upon the problem. For example if there is a need to integrate with Web applications for data mining, Python can do that. R, however, will be limited in its usage such as it can't interface with the OS and it is designed with an ad-hoc, one time dive mindset while Python is more versatile and capable as it can interact with a range of databases and ease the process of building analytical tools for the specific requirements.

It is important to understand that the work environment is changing and there is an ever increasing need for a tool that provides end-to-end integration rather than a tool that does a specific job and can't move beyond statistics. Also, the need and scale of producing analytical applications are also increasing. A lot of times there is a need for automating the tasks and develop data related products where only high-level languages can be useful. Here Python is able to contain and link the entire workflow, is able to implement algorithms for production purposes and as companies already have established production systems based on Python, knowing Python makes more sense.

Today, the domain of data science is not limited for statisticians and mathematicians and often requires innovator and visionaries who can think out of the box to solve problems related to multiple domains. Python is not only able to integrate multiple systems but is able to integrate multiple people belonging to different backgrounds. While R became the lingua franca of statistics, Python was able to integrate people belonging from different academic backgrounds, from beginners to software engineers to already established data scientist, all because of its shorter learning curve.

This integration of not only of system and platform but of people from various fields has lead to the diversification of the usage of python which becomes very important for companies employing people with diverse background and dealing with a range of products and clients.
Diversification is a key to success and this is why Python has excelled.

4. ROUND 3: USAGE

This is where SAS lost to R and this is where R loses to Python – the range of usage. There are a number of areas where R and Python compete and while in some areas R takes the lead, overall it is Python that comes out to be as a winner.

Sharing, Organization and Management

As far as SAS is concerned, the aspects of sharing, organizing and managing SAS are not independent and they work in the tight framework of SAS regulations so there are facilities of perform all such tasks but there is little scope of rapid improvement.
With a range of IDE available for Python, especially IPython and Jupyter notebooks which provide extensive options for proper documentation, sharing, managing and organizing the codes becomes very easy while with R, documentation is still very ordinary.

Also, codes written in Python are more consistent and hence are more dependable and reusable when shared. The ease with which user defined modules can be created and shared leads to a very high code reusability for python users and this has befitted python greatly whereas R is still not yet developed in terms of code reusability.

Statistical and Predictive Models

This is a whole other issue where the debate or competition between R and Python becomes fierce. SAS one hand is great with the statistical and classic predictive models and is the reason that still a lot of companies with the requirement limited to such models continue to use SAS. The world, however, has moved from traditional Statistical and Classic Predictive models and we are now in the times of Machine Learning and Deep Learning models but still, for a range of organizations, the traditional statistical models are the need of the hour.

R was created keeping statisticians in mind and hence is able to undertake statically complex model with ease. The outcomes of various statistics related packages or tasks performed by it are backed statisticians as the language is, still today, used in prestigious institutions. On the other hand when it comes to classic predictive modeling, both r and Python are able to create predictive models with much effectiveness.

Academic Researches

R has always been the favorite for academic research which requires deep diving into a subject while Python is not as much effective in the academic setting. One of the reason is that R precedes python when it comes to data related tasks. The first group of people that adopted R were those who earlier relied on SAS, SPSS or Starta in order to perform their research work and as R is backed by the statistician community, it became easier for R to make a name for itself.

In recent times, Academic Research based outside the scope of traditional statistics based research has chosen Python over R as it is helpful to perform the complex often Machine Learning related tasks on Python.

4. ROUND 3: MACHINE LEARNING V/S DEEP LEARNING V/S ARTIFICIAL INTELLIGENCE

Definitions

In recent times, terminologies such as Machine Learning, Deep Learning and Artificial Intelligence have become quite popular. In order to discuss the battle of tools in order to implement these concepts, it is important to understand what these concepts stand for.

Whenever we achieve something without explicitly hardcoding it that is creating rules and condition based programs and let the machine find its own course in order to find the best result is a concept used under Machine Learning.

Now the important question is, what is Artificial Intelligence, the term we often hear every now and then. Artifical intelligence is nothing but all the ceoncepts of machine learning where computers learn automatically without much human intervention but acts more like a human where it learns from its mistake and rather than creating a new model altogether, the model has a self-healing mechanism and it is the reason it is also called Machine Intelligence i.e. Machine Learning with wit the capabilities of human learning and problem solving.

Deep Learning is another concept which is closely related to both the Machine and Artificial Learning. It is basically an architecture through which Artificial Intelligence can be achieved where we use a concept known as Neural Networks where the calculations are performed by mimicking the functioning of the human brain.

Tool of Choice

SAS do have some support for machine learning but when it comes to Deep Learning and any other Artificial Intelligence related concepts, it trails way behind R and Python. Sometimes the fight between R and Python can translate into the fight between data analysts and data scientist depending how these terms are defined. Still with data analysts requiring more of statistical exploration, R comes in handy while Data Science nowadays requires Machine Learning tools where because of scikit learn, Python comes as the undeniable winner. Because of Python's flexibility, sophisticated predictive models can be created using machine learning tools that can be then plugged into production systems. Also, R is not able to scarp and analyze unstructured data the way Python is able to do and lacks especially in the field of NLP etc. As far as Deep-Learning is concerned, there is no doubt that Python makes the process hassle-free while R struggles to implement deep learning algorithms with the ease and sophistication Python's TendorFlow and Keras do.

> SAS trails way behind R and Python. Sometimes the fight between R and Python can translate into the fight between data analysts and data scientist depending how these terms are defined

5.
ROUND 4:
BIG DATA

In recent times, the amount of data being generated has increased and is increasing rapidly. This has led to the further amplify the 'curse of dimensionality'. Because of the sheet amount of data and at the frequency it is being generated, the tools have to adopt to various platform through which such a data can be stored and accessed.

> The most suitable contender here is **Python** which is then able to use machine learning algorithms using the vast amount of data at hand.

Integrating the data science tools with such platforms is very important and the learning curve of integration R with such platform is very steep. Also integration of the Big Data architecture with R is simply not feasible. The most suitable contender here is Python which is then able to use machine learning algorithms using the vast amount of data at hand. This is facilitated because of the fact that it has multiple areas of application and easy to read and write making it possible to create scalable applications out of it. Also, there are lot many packages available for Python in order for it to integrate with Big Data platforms.

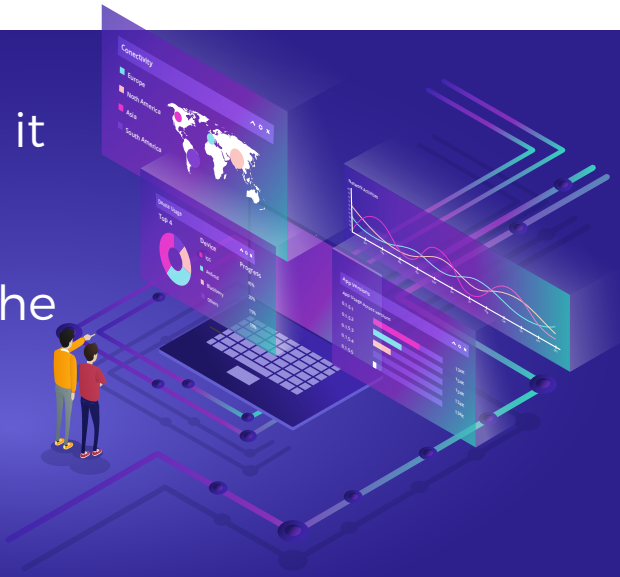
6. ROUND 5: VISUALIZATION

Visualization is one of the most important aspects when dealing with data as it allows very complicated information to be understood in a single glance. SAS, as mentioned earlier, can create publish worthy graphs but the cost and its limited application had caused others to get ahead of it.

Traditionally and till now to some extent, R has been attributed as the better candidate. Python seems a bit behind with packages such as Seaborn, Pygal, Bokeh in front of R that has packages such as ggplot2, googleVis and rCharts. Also, some of the visualization packages of python are unstable and hence unreliable.
R, on the other hand, the has ggplot which can create almost kind of graphs and is the backbone of the visualization in R. Also, creating dashboards is also very easy in R. One of the major advantage R has over Python is because of the package known as Shiny which helps in creating lightweight web application where user interactive charts can be created. Also, Shiny is very easy to understand and implement and so far Python has no package to counter is successful.

However, things have started to change with Python having packages such as Altair that has reduced the gap. Also, there Seaborn is a machine learning oriented visualization package of Python that can create very sophisticated and complex Machine Learning related plots which R also can but not to the extent of Seaborn. Sill, so far when it comes to visualization, R seems to take the head.

> “Sill, so far when it comes to visualization, R seems to take the head.



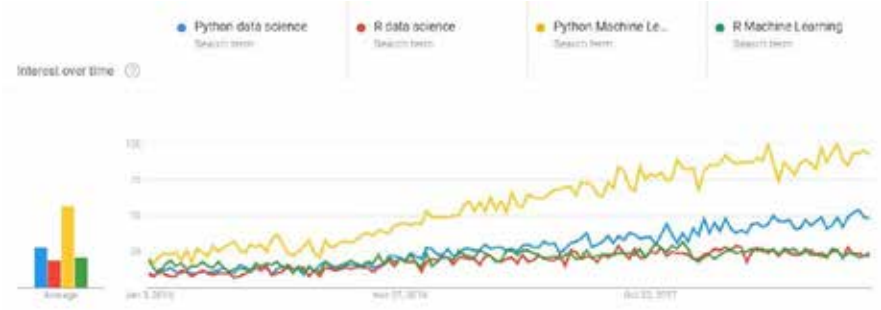
7.
ROUND 6:
MEN LIE, WOMEN LIE, NUMBER'S DON'T

Till now, it has become evident that the main rivals are R and Python and SAS is out of the race because of its bleak future. Here are few of the numbers that will clear if there is still any doubt about who has won this Great Battle of Data Science. As of 2016, the % of matching jobs posting for R was 0.16% while for Python it was around 0.30% *(source: r4stats.com)*

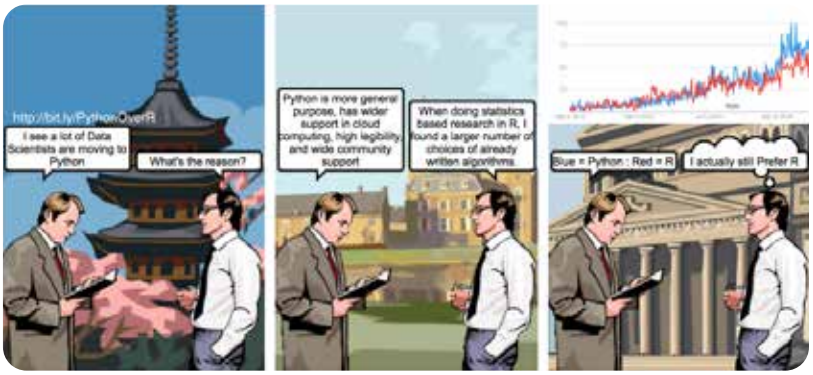
As of 2017, Python has maintained the lead in number of users over R. Also, with 74% users being loyal to R and 97% being loyal to Python and with 10% switching from R to Python and only 5% switching the other way around, Python seems to have won the race. In another poll, R showed a negative 14% change while Python showed a positive 11% change with as of 2018, 65% share of Data Science/ML Software being of Python. *(source: kdnuggets.com).*

Software	2018 % share	% change 2018 vs 2017
Python	65.60%	11%
RapidMiner	52.70%	65%
R	48.50%	-14%
SQL	39.60%	1%
Excel	39.10%	24%
Anaconda	33.40%	37%
Tensorflow	29.90%	32%
Tableau	26.40%	21%
scikit-learn	24.40%	11%
Keras	22.20%	108%

As per Google Trends, as of 2018, 'Python Data Science' was higher than 'R Data Science' and if we compare 'Python Machine Learning' with 'R Machine Learning' then the lead Python has over R is almost of 4.5 times *(source: trends.google.com)*.



With all the above given numbers, it is very much clear that Python has won over R, still it is easy to understand why someone may choose R over Python. Python is the first choice by developers and programmers while R is loved by statisticians. Also, as R had been active in the Data Science fields prior to Python, many of the professionals learnt R first and accepting and trying something new always feels uneasy, at least initially.



8. ROUND 6:

THE LANGUAGE TO START OFF WITH

Syntax and Learning Curve

Whenever learning a new languages and especially if someone is new to field of Data and has to understand the various concepts for Dealing with data, it becomes imperative that the language as a catalyst rather than as something that causes hindrance in the learning process.

R is low-level language created by keeping statisticians in mind and is often said that R makes the life of a statistician easier, not of a computer. It is created specifically for the task of complex statistical computing and data analysis. Python on the other hand is a high-level language that has an easy to understand, flexible, well structured, concise syntax for increasing code readability, simplicity, performance and productivity and it works to make the life of both the programmer and computer easy. When pitted against each other, several flaws of R tend to surface.

Firstly, Python is simple and readable when compared to R. It even has sort of 'code of conduct' known as the Zen of Python which is a set of principles that is responsible for the way the code is designed which causes Python to have a standard code making it extremely easy to read while R is devoid of any such discipline and is pretty much nonstandard causing obstacles in the process of programming.

Secondly, Python is object-oriented while R is not and is a procedural language, meaning that it does the process in a very step by step manner making the user to write much more lines of code for a task when compared to Python. The time saved here can be used for exploration and experimentation providing better understanding of the data.
All of this corresponds to the fact the learning curve for R is steeper than Python. Without a programming background the process of learning R can be exhausting while on the other hand Python has a very flat learning curve.

> All of this corresponds to the fact the learning curve for R is steeper than Python.

Data Analytics and Data Science

The various tasks performed using data have led to two major disciplines which are Data Analytics and Data Science. There are a number of nuance difference and some major difference between the two fields with often definitions being different depending upon the person being asked to explain the features of these disciplines. Still, one of the definition can be that Dana Analytics is something where the major chunk of the traditional data analysis tasks lies where we get the data and perform typical tasks which tell us about the various static features about the data. This includes a lot of descriptive analytics, valuations and the use of inferential statistics to a certain extent.On the other hand Data Science is a broader term that encompasses the aspects of Data Analytics also but also has the concepts of complex model creation and other data-based driven products using Machine and Deep Learning.

Over a period of time, the languages that specialize in each of their fields have become more and more distinct. SAS, SPSS were, are and will be the tools that can easily be implemented for the routine typical data analytics. With the lack of support for the modern concepts of Machine and Deep Learning, we are left with R and Python and this is where things become interesting and complicated.
Technically, both R and Python can work in both the disciplines of Data Analytics and Data Science. However, for Data Analytics still, R can be the right choice as because of its sound statistical inner workings and the ease with which it performs the various aspects of data manipulation and visualization which form the backbone of data analytics. Now, python can also perform the same tasks with the ever-increasing of its modules such as pandas and other visualization modules still one can be inclined towards R for performing such tasks.

> Technically, both R and Python can work in both the disciplines of Data Analytics and Data Science.

Things become interesting when we are in the field of Data Science as, again, technically both can perform the various Machine Learning tasks but when it comes to the range of options and the ease of implementation, the difference between R and Python becomes stark with Python taking the lead. Almost all the Machine Learning models that can be created in R can be created in Python and vice-a-versa but the sheer complexity and lack of cohesiveness in R makes people choose Python over R and these reasons get more highlighted when we deal with Deep Learning modes with the R are having no match for the sophistication of the method Deep Learning Modules such as TensorFlow work in Python.
The use of Python for data science is becoming common and if both the languages continue to develop at the current pace then in some time it will become very clear that Python is the language for the future as the future developments are going to be in Machine Learning, Deep Learning, Augmented Intelligence and Artificial Intelligence where Python has excelled over R.

Courses

In order to start a career or to enhance the skill set, there are a number of data science courses and by and large, these courses are either taught in R or are taught in Python with SAS forming a smaller proportion. As far as data analytics courses are concerned, the proportion of courses taught in SAS tend to increase. However, Python gives very stiff completion to R when it comes to machine learning courses as mentioned earlier, the machine learning models can be created using both R and Python. The decisive win of python becomes evident with the widespread and almost monopoly of python as the preferred tool in any major artificial intelligence course. Artificial Intelligence or for that matter deep learning, which are two closely related concepts rely heavily upon python and if someone has plans of venturing into these fields that Python can be the tool to learn and start with.

“ Artificial Intelligence or for that matter deep learning, which are two closely related concepts rely heavily upon python and if someone has plans of venturing into these fields that Python can be the tool to learn and start with.

CONCLUSION

The extreme division in opinion is visible when it comes to decide the tool best for Data Science. Today the term data science is no more limited to exploring data or creating statical models but requires production of algorithms and attaining data from various unconventional sources. It also requires automation and this is where R seems rigid, too specialized and Python seems more inclusive and diverse and makes it the ideal candidate for many technology driven companies causing people to switch. Just like SAS had, R also has the stronghold in the financial sector and this is the toughest wall to break for any new tool as this sector is less open to frequent changes and experimentation, still, recent developments such as Bank of America opting for Python shows that the wall is cracking and lingua franca for data scientist is no more unopposedly R. An unsaid benefit of knowing Python is that once it is learnt for data analysis, one can move beyond into the field to programming and development if wishes while R sort of hits a dead end and can't be made to move beyond data analysis. R seems to have stuck with solving the statistician's problem and the 'by the statisticians for the statisticians' approach has caused it to lag behind whereas Python is able to undertake a range of problems. However, it must be remembered that increasingly more people are using a combination of both the languages and with Python providing RPy2 which can make the user to have all the R's major functionality along with the fact that Python is easy to collaborate with, Python has decisively won the battle.

ANALYTIX LABS

Contact us

Need help to know more about career in **Analytics?** Reach us out from following medium and let our counselor guide you

Gurugram

GF 382, Sector 29, Adjoining IFFCO Chowk Metro Station (Gate 2), Next to Vasan Eye Care Hospital, Gurgaon, Haryana 122001, India

Bengaluru

Bldg 41, First floor, 14th Main Road, Near BDA complex, Sector 7, HSR Layout Bengaluru - 560102 India.

+91 95-55-219007

info@analytixlabs.co.in

www.analytixlabs.co.in