1)

AvidBeam :

AvidBeam® is a video Big Data Analytics company that was founded with combined computer vision, Big Data and video analytics expertise. AvidBeam® aims to generate a leap in video processing capabilities with Big Data distributed architecture. AvidBeam was founded by a team of ex-Intel PhD’s and engineers with the vision of extending “big data” scope to cover not-only text data but also video processing analytics. AvidBeam made it to CIO Review’s 2016, and 2017 top worldwide Big-Data startups list for its innovative Open Video Platform that extracts real-time business intelligence information from enormous number of Video sources. In 3 years, AvidBeam team developed and delivered few video processing/analytics projects in a few verticals and geographies including Middle East, China, Europe and US. Projects spanned video surveillance, retail, perimeter protection and automotive industries.

وادى النيل، معادي الخبيري الشرقية، قسم المعادي، الشرقية

2)

Nahdet Misr:

Since more than 80 years, we have been striving to empower people’s development and growth through valuable education, knowledge and skills, thus enabling them to enjoy a lifelong learning journey. In accordance with our vision to develop knowledge that would promote the lives of the Arab people at large, we offer world-class educational and cultural content and solutions in printed and digital format.

Nahdet Misr (NM) was established in 1938 as a modest printing house that offers cultural books, has verged into the educational sector as well, thus developing educational curricula and textbooks for Ministries of Education in several Arab countries. Along the years, we have embraced change and have always been a step ahead in the cultural and educational fields. Our business has grown into seven companies that offer integrated services and solutions in the areas of education, digital learning, educational software, publishing and entrepreneurship.

المقر الرئيسي: مدينة السادس من أكتوبر

3)

AIM Technologies :

In today's connected
era, customers do
everything online.
Technology has become critical to capture massive amounts of unstructured data fully.
So, we developed our own.

1200
Registered brands

3%
Revenue Increase

87%
Arabic Sentiment Accuracy

20.000
survey questions

AIM Technologies was created with the purpose of redefining the Insights & Knowledge mining industry across the MENA region.

Stemming from a need for thorough understanding of customer behavior and sentiment, we strive to become the multilingual AI powerhouse of the Middle East.

In 2019, 6 founders with a thirst for innovation and a fusion of 15+ years of experience across Egypt, MENA, and Europe, came together in creating the first multilingual AI analytics tool in the region.

Our tools help our clients better understand their consumers’ voice and thrive in saturated markets; regardless of their size. And to this date, we’ve helped (x) brands thrive in their industries and stay ahead.

- Social listening
- Market Research

العنوان: 40 El Golf street, معادي السرايات الغربية، قسم المعادي، محافظة القاهرة

Compiled Language
vs
Interpreted Language

Comparison Chart

Compiled Language	Interpreted Language
The code of compiled languages can be executed directly by the computer's CPU.	A program written in an interpreted language is not compiled, it is interpreted.
The source code must be transformed into machine readable instructions prior to execution.	It does not compile the source code into machine language prior to running the program.
Compiled programs run faster than interpreted programs.	Interpreted programs can be modified while the program is running.
Delivers better performance.	Delivers relatively slower performance.
C, Fortran, and COBOL are languages used to produce compiled programs.	Java and C# are compiled into bytecode, the virtual interpreted language.

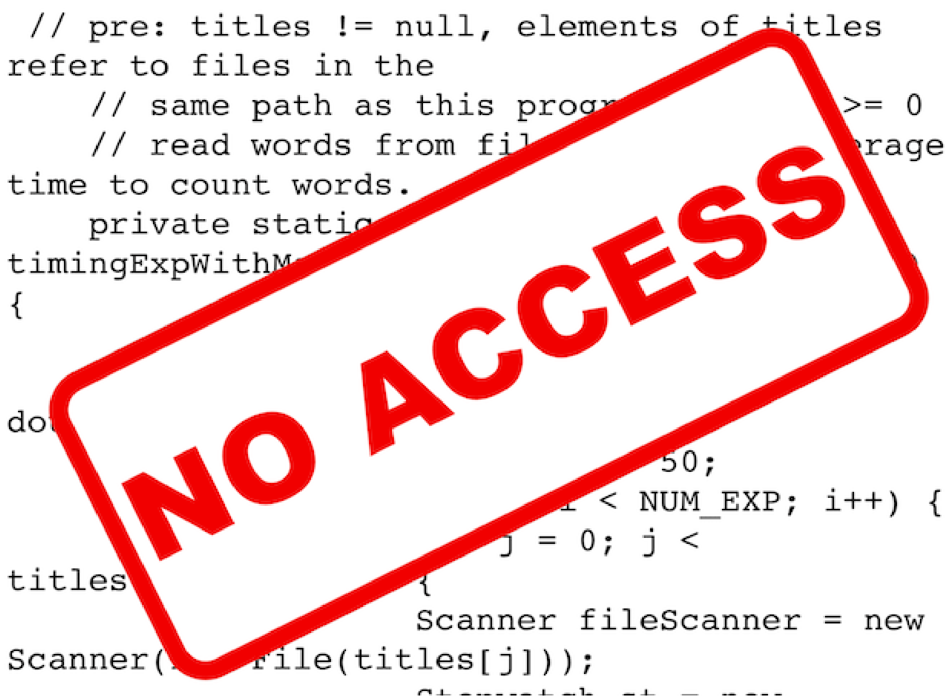
DB DifferenceBetween.net

Open source vs. closed source software

Almost every piece of computer software is created using source code, which is the technical blueprint that tells a program how to function. When creators release their finished product to the public, they must decide whether to make its code open source or closed source.

What is open source and closed source?

With closed source software (also known as proprietary software), the public is not given access to the source code, so they can't see or modify it in any way.



But with open source software, the source code is publicly available to anyone who wants it, and programmers can read or change that code if they desire. Keep in mind that you don't have to read or modify any code in order to use an open source product.

Which type is more common?

The vast majority of apps, games, and other popular software is closed source. However, there are open source options for many types of programs. If you want an open source alternative to Microsoft Office, you could use LibreOffice. Instead of using Windows, you could try an open source Linux operating system. Other common open source examples include the Firefox web browser and WordPress blogging platform.

What are the pros and cons?

One of open source's biggest advantages is that it's usually free, although some features and technical support may cost extra. Also, because the code is available to anyone who wants it, public collaboration can fix bugs, add features, and improve performance within a relatively short amount of time .

Here are the 10 open-source programming languages that are being used most for open source projects.

1.JavaScript

2.Python

3.PHP

4.Swift

5.R Programming

6.C++

7.Go

8.Kotlin

9.Scala

10.Ruby

A language isn't open-source or closed-source as such. For example, G++ is open source while MSVC++ is closed source. ISO C++ is neither, it's a non-free non-proprietary standard.

Some examples of closed source software are Skype, Google earth, Java, Adobe Flash, Virtual Box, Adobe Reader, Microsoft office, Microsoft Windows, WinRAR, mac OS, Adobe Flash Player etc.
non OOP languages include :

- Assembler.
- Fortran.
- Forth.
- Pascal.

MIT Technology Review Subscribe

ARTIFICIAL INTELLIGENCE

A radical new technique lets AI learn with practically no data

"Less than one"-shot learning can teach a model to identify more objects than the number of examples it is trained on.

By Karen Hao

October 16, 2020

The mythical rhinocorn.

MS TECH / PIXABAY

Machine learning typically requires tons of examples. To get an AI model to recognize a horse, you need to show it thousands of images of horses. This is what makes the technology computationally expensive—and very different from human learning. A child often needs to see just a few examples of an object, or even only one, before being able to recognize it for life.

In fact, children sometimes don't need *any* examples to identify something. Shown photos of a horse and a rhino, and told a unicorn is something in between, they can recognize the mythical creature in a picture book the first time they see it.

Hmm...ok, not quite.

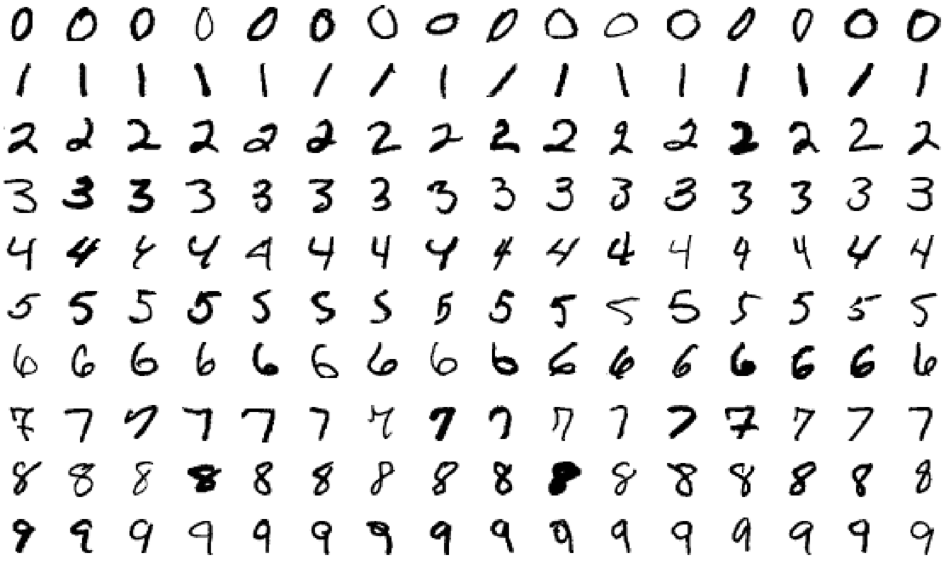
MS TECH / PIXABAY

Now a new paper from the University of Waterloo in Ontario suggests that AI models should also be able to do this—a process the researchers call “less than one”-shot, or LO-shot, learning. In other words, an AI model should be able to accurately recognize *more* objects than the number of examples it was trained on. That could be a big deal for a field that has grown increasingly expensive and inaccessible as the data sets used become ever larger.

How “less than one”-shot learning works

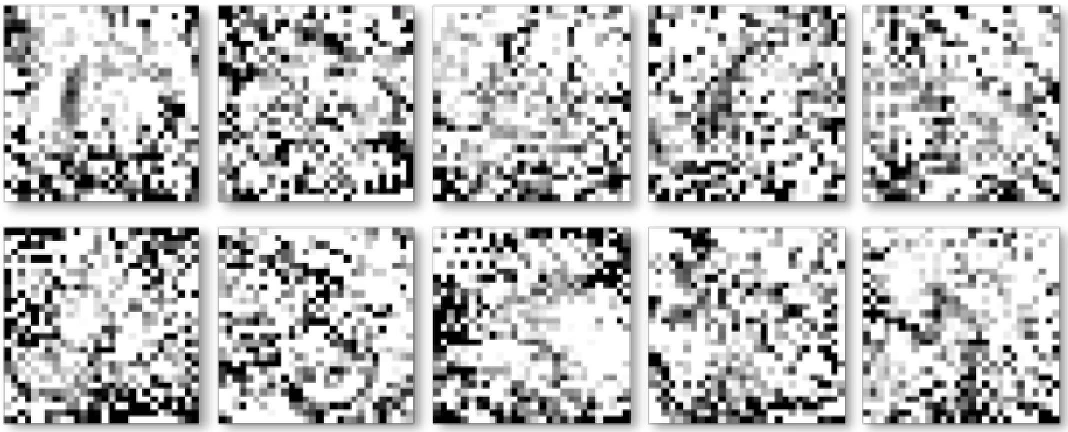
The researchers first demonstrated this idea while experimenting with the popular computer-vision data set known as MNIST. MNIST, which contains 60,000 training images of handwritten digits from 0 to 9, is often used to test out new ideas in the field.

In a previous paper, MIT researchers had introduced a technique to “distill” giant data sets into tiny ones, and as a proof of concept, they had compressed MNIST down to only 10 images. The images weren’t selected from the original data set but carefully engineered and optimized to contain an equivalent amount of information to the full set. As a result, when trained



Sample images from the MNIST dataset.

WIKIMEDIA



The 10 images "distilled" from MNIST that can train an AI model to achieve 94% recognition accuracy on handwritten digits.

TONGZHOU WANG ET AL.

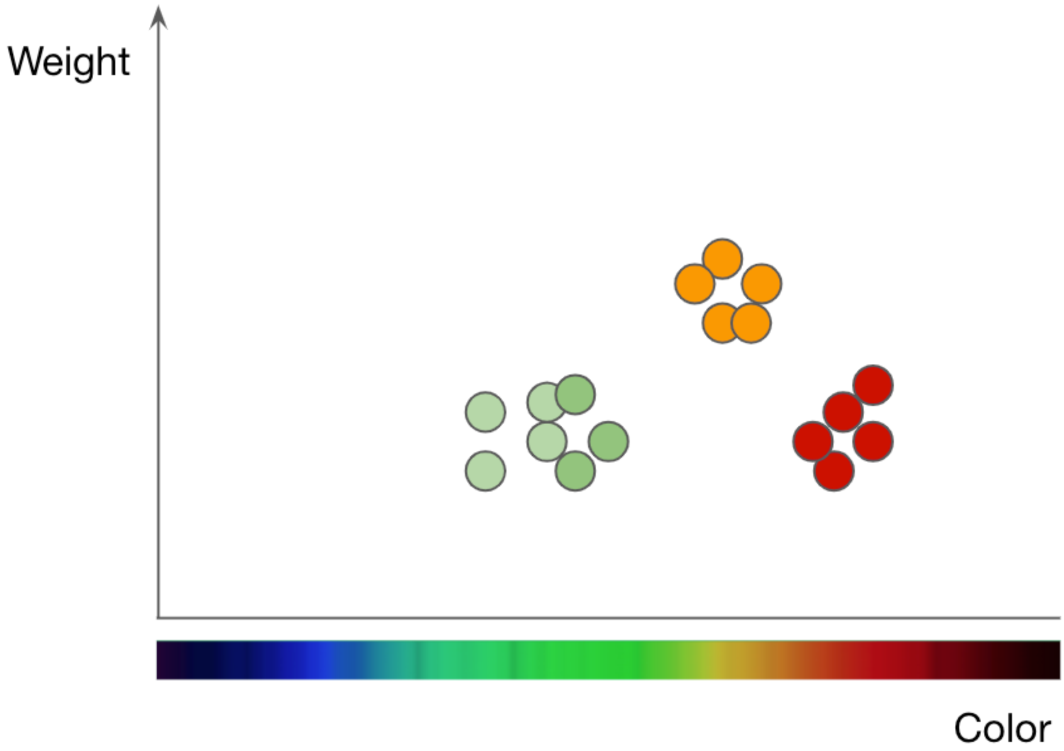
The Waterloo researchers wanted to take the distillation process further. If it's possible to shrink 60,000 images down to 10, why not squeeze them into five? The trick, they realized, was to create images that blend multiple digits together and then feed them into an AI model with hybrid, or "soft," labels. (Think back to a horse and rhino having partial features of a unicorn.)

"If you think about the digit 3, it kind of also looks like the digit 8 but nothing like the digit 7," says Ilia Sucholutsky, a PhD student at Waterloo and lead author of the paper. "Soft labels try to capture these shared features. So instead of telling the machine, 'This image is the digit 3,' we say, 'This image is 60% the digit 3, 30% the digit 8, and 10% the digit 0.'"

The limits of LO-shot learning

Once the researchers successfully used soft labels to achieve LO-shot learning on MNIST, they began to wonder how far this idea could actually go. Is there a limit to the number of categories

Surprisingly, the answer seems to be no. With carefully engineered soft labels, even two examples could theoretically encode any number of categories. “With two points, you can separate a thousand classes or 10,000 classes or a million classes,” Sucholutsky says.



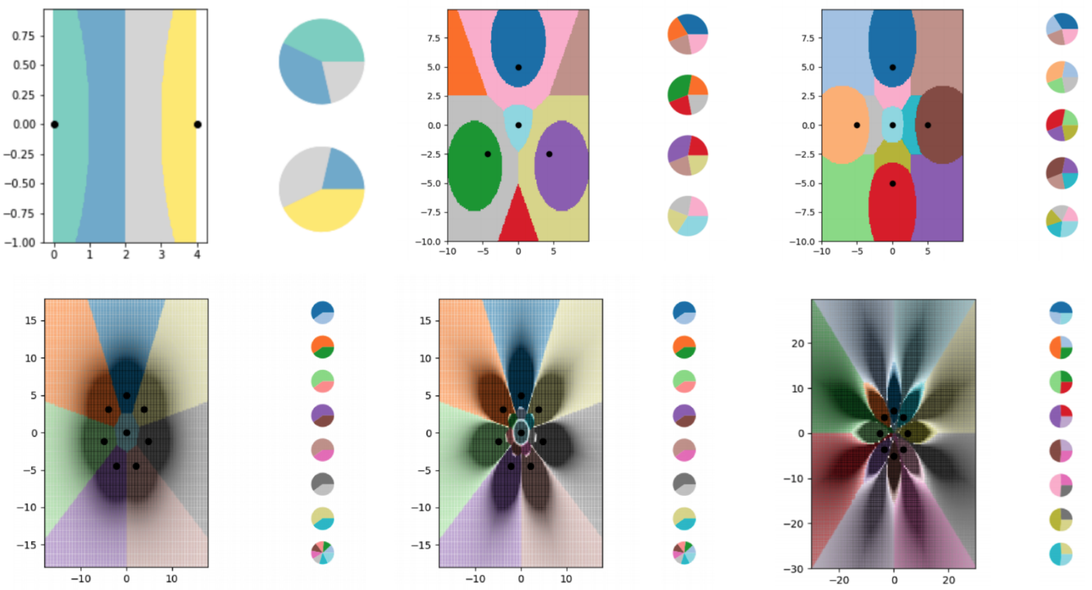
Plotting apples (green and red dots) and oranges (orange dots) by weight and color.

ADAPTED FROM JASON MAYES' "MACHINE LEARNING 101" SLIDE DECK

This is what the researchers demonstrate in their latest paper, through a purely mathematical exploration. They play out the concept with one of the simplest machine-learning algorithms, known as k-nearest neighbors (kNN), which classifies objects using a graphical approach.

To understand how kNN works, take the task of classifying fruits as an example. If you want to train a kNN model to understand the difference between apples and oranges, you must first select the features you want to use to represent each fruit. Perhaps you choose color and weight, so for each apple and orange, you feed the kNN one data point with the fruit’s color as its x-value and weight as its y-value. The kNN algorithm then plots all the data points on a 2D chart and draws a boundary line straight down the middle between the apples and the oranges. At this point the plot is split neatly into two classes, and the algorithm can now decide whether new data points represent one or the other based on which side of the line they fall on.

To explore LO-shot learning with the kNN algorithm, the researchers created a series of tiny synthetic data sets and carefully engineered their soft labels. Then they let the kNN plot the boundary lines it was seeing and found it successfully split the plot up into more classes than patterns in the shape of flowers.



The researchers used soft-labelled examples to train a kNN algorithm to encode increasingly complex boundary lines, splitting the chart into far more classes than data points. Each of the colored areas on the plots represent a different class, while the pie charts to the side of each plot show the soft label distribution for every data point.

ILIA SUCHOLUTSKY ET AL.

Of course, these theoretical explorations have some limits. While the idea of LO-shot learning should transfer to more complex algorithms, the task of engineering the soft-labeled examples grows substantially harder. The kNN algorithm is interpretable and visual, making it possible for humans to design the labels; neural networks are complicated and impenetrable, meaning the same may not be true. Data distillation, which works for designing soft-labeled examples for neural networks, also has a major disadvantage: it requires you to start with a giant data set in order to shrink it down to something more efficient.

Sucholutsky says he's now working on figuring out other ways to engineer these tiny synthetic data sets—whether that means designing them by hand or with another algorithm. Despite these additional research challenges, however, the paper provides the theoretical foundations for LO-shot learning. "The conclusion is depending on what kind of data sets you have, you can probably get massive efficiency gains," he says.

This is what most interests Tongzhou Wang, an MIT PhD student who led the earlier research on data distillation. "The paper builds upon a really novel and important goal: learning powerful models from small data sets," he says of Sucholutsky's contribution.

Ryan Khurana, a researcher at the Montreal AI Ethics Institute, echoes this sentiment: "Most significantly, 'less than one'-shot learning would radically reduce data requirements for getting a functioning model built." This could make AI more accessible to companies and industries that have thus far been hampered by the field's data requirements. It could also improve data privacy, because less information would have to be extracted from individuals to train useful models.

Sucholutsky emphasizes that the research is still early, but he is excited. Every time he begins presenting his paper to fellow researchers, their initial reaction is to say that the idea is

Radix

Machine Learning without data...why not?

By Brecht Coghe September 9th, 2021 7 minutes

MACHINE LEARNING DATA SCIENCE ARTIFICIAL INTELLIGENCE

In many minds, Machine Learning is inevitably connected with having data. And although having heaps of ‘the new gold’ is indeed often an advantage when starting your AI project, it is not always necessary or even possible to have such amounts of data. It can be very complex or expensive to gather it, and your business might just not be ready for

Radix

Data... before or after?

As Machine Learning and AI became more and more popular in the last few decades, more and more use cases in the industry popped up and proved to be of enormous value. From 'the early days' of spam classification, over demand forecasting, to state-of-the-art applications such as using transformers trained on the entirety of Wikipedia for generating natural responses to customers or using Generative Adversarial Networks to upscale your images. A lot of these applications have in common that people had data lying around and wanted to make use of it. Fed by new AI technologies from research, applications were made based on the data available. If you look at it, it is a very natural way of growth in the AI field. I call this *the data before approach* and so, in other words, the application or algorithm *after approach*. As already said, this has proven itself to unlock massive potential with AI.

However, at Radix, we notice that thinking in terms of *the data before approach* limits companies in unlocking the full potential AI has for them. If the data is not there yet, some useful AI applications might not even be considered, and possible competitive advantages might not be uncovered. That is why *the data after approach* is often also a good perspective, and as Radix, we want to encourage every one of you to consider what AI can do for you without thinking of what data you have first! You start with listing your business problems or potential use cases, you select the ones that might be solved with AI (if needed, involve an expert!) and only then you see if you have the right data, if it needs to be gathered and/or which structure or format it needs to have to solve your case. This process is how your company can build a structured AI strategy roadmap that will allow you to work towards improving your business instead of being limited by the data you have at this very moment. As you can guess, it often leads to thinking about *Machine Learning without having data.. But, Why not?*

ML with little data advancements

Of course, when you have the use cases you want to solve, still having to gather thousands to even sometimes hundreds of thousands of data points is not always feasible. Luckily, the AI field has had an exponential decrease in the amount of data needed to solve a problem. As already predicted in 2015 by Lecun, Bengio, and Hinton in their [Nature Review](), unsupervised, semi-supervised, and self-supervised learning

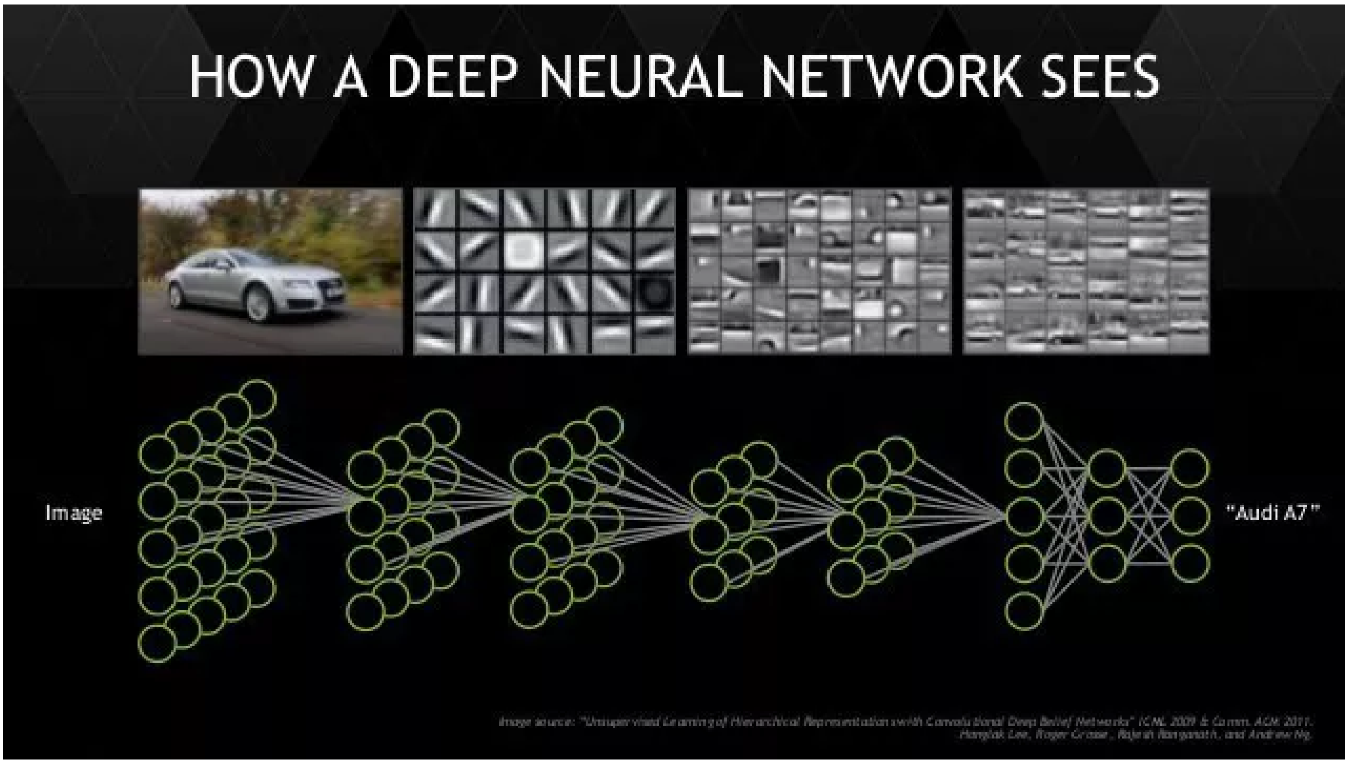
Radix

do ML with little or no data.

Transfer learning

We, humans, are good at learning new things if we have already learned something similar. If we know how a dog, horse, or cat looks, it will be easier to learn how to identify a donkey. That is because we learn lower-level concepts, f.e. Fur, paws, snout,... Machine learning algorithms actually show the same behavior. Let's look at an example of car type classification. When first trained on a huge dataset of images, f.e. they learn low-level features such as corners, circles, stripes, etc. Then higher-level features such as windows, wheels, body,... If you use the same Neural Network to then learn to classify boats or trucks, it will be able to reuse a lot of its knowledge and require only a minimal amount of data.

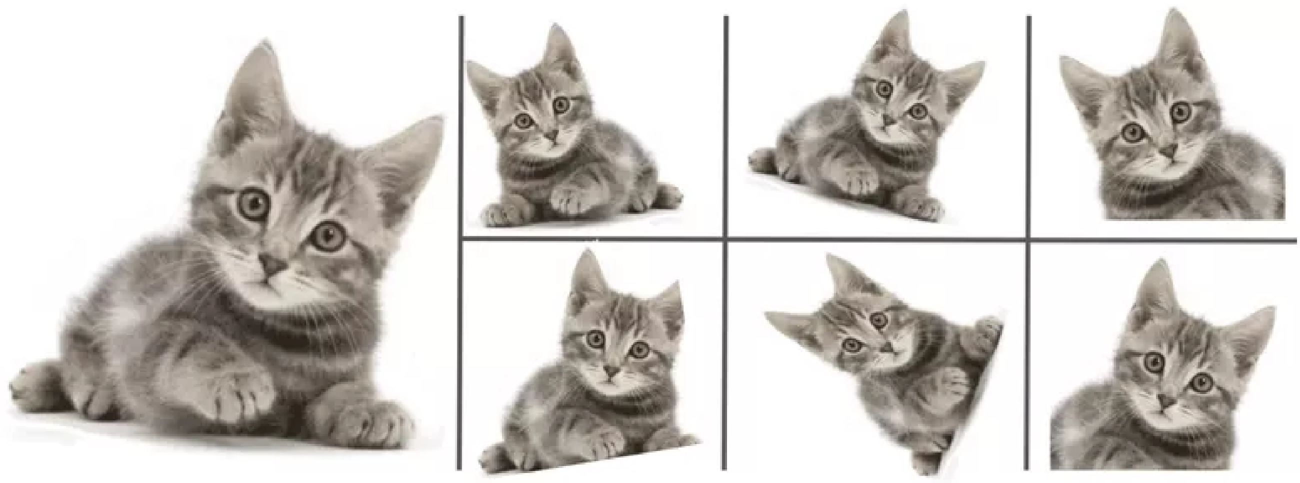
Transfer learning is often used in combination with publicly available datasets. One good example is making use of Wikipedia to train the newest state-of-the-art Language Models. But also in your use cases, some similar data might be available to start learning from. And if that is not the case, you can even use simulated data!



A neural network first learns low level-features and further in the network, it learns higher-level concepts. These features and concepts can be reused for other applications while needing fewer data examples.

Radix

Who doesn't like kittens, so what better way to explain data augmentation than to use kittens. Data augmentation is a technique where you use data points (f.e. Images of kittens) that are already labeled and do transformations on them. The assumption is that the label will not change by doing these transformations. Still, the image looks like a totally different combination of 1's and 0's and thus as a new learning opportunity to the model. These transformations can be rotations, scaling, adding noise etc for images, but it is also used in Natural Language Processing by using SOTA translating models to translate a sentence to another language and back, so small perturbations occur on it. One of the more advanced examples is using generative models to produce 'new' data points.



For us, this looks like almost the same picture of the same cat. For a ML model, this looks like a set of totally different images.

Unsupervised learning

Supervised learning uses labels to show your model what is *good* and *wrong,* and your model can learn from that. That is the most standard case of machine learning. However, some algorithms can find patterns in your data without even having labels. Unsupervised learning. The most known algorithms are clustering and anomaly detection. You can make use of it by finding groups of similar data points first and then see if these groups make sense and label the groups rather than labeling every data point. Say you have 5000 data points per group, I'd say that can save you a lot of effort!

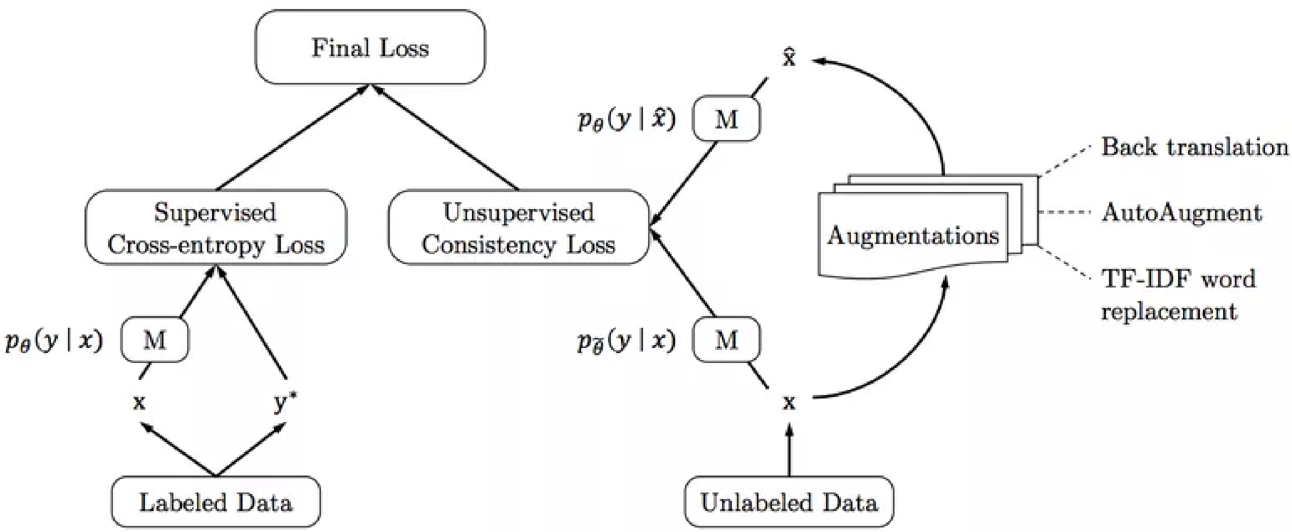
Self-supervised learning

Radix

making use of the data it already has. In NLP, for example, the model can look at a text, leave a word out, and try to predict the missing word based on its context. By doing that, it can learn structure in data, and this actually leads to the SOTA LMs that have taken a big step in the last years. These language models can then be used to do a downstream task such as question-answering or summarization based on limited sets of data.

Mix it up!

Each of the above advancements helps us to solve AI problems with less and less data. But the most interesting is the ability to combine all of those in creative ways to test Machine Learning with little data (or even no data) to its limits. One example is an algorithm called Unsupervised Data Augmentation. As the name suggests, it uses unlabelled data to learn from. The only assumption made is that the prediction of an original data point and a slightly augmented data point should be the same, regardless of the label it actually is. A simple assumption, but the resulting improvements can be enormous! In some cases, almost the same error rates can be reached with a factor of 1000 fewer labeled data points. Another State-of-the-art yet more complicated example of a semi-supervised approach is MixMatch!



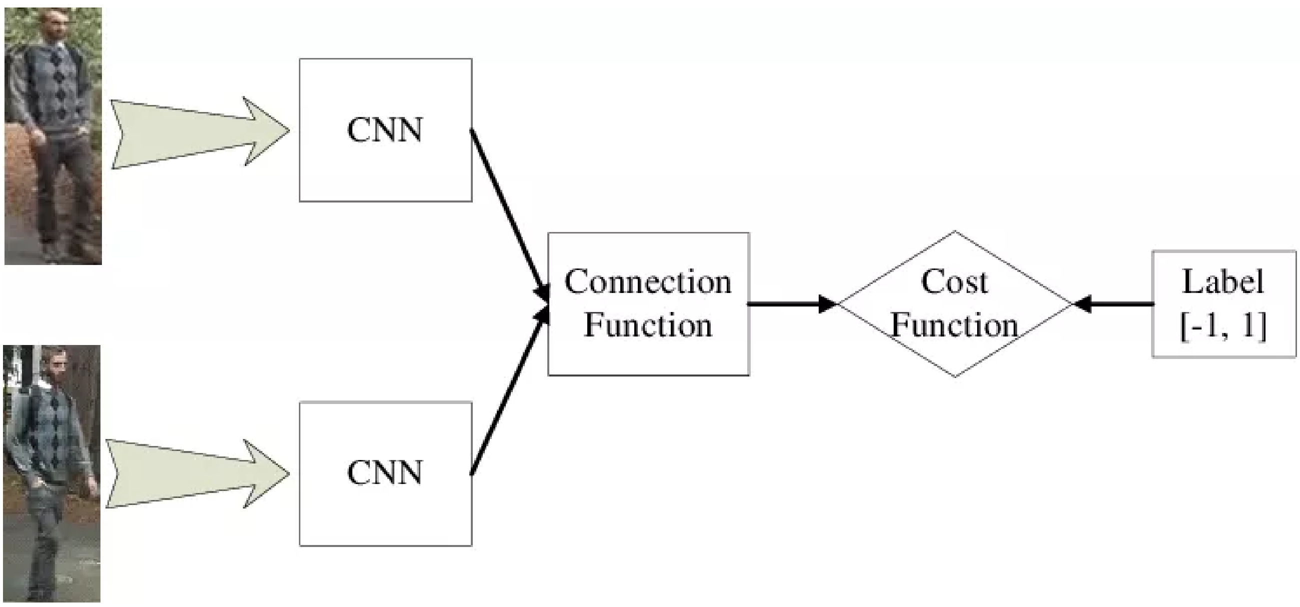
Pre-BERT SOTA		*4.32*	2.16	29.98	3.32	34.81	0.70
$\text{BERT}_{\text{LARGE}}$		4.51	*1.89*	*29.32*	*2.63*	*34.17*	*0.64*

Semi-supervised setting

Initialization	UDA	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
$\text{BERT}_{\text{BASE}}$	✗	18.40	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
$\text{BERT}_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
$\text{BERT}_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-

Unsupervised Data Augmentation: Similar error rates can be reached with up to a factor of 1000 less labeled images

And if all of the above does not help you solve your problem, you can look into posing your question slightly differently to use one of the ML with little data techniques. As the last example, imagine classifying 100 parts where you only have 5 examples per part. Even with transfer learning, five examples per class might prove to be insufficient. One way to maybe deal with this situation is to not ask the question “Which part is it?” from the start, but to ask, “Are these two images the same part or not?”. This transforms your dataset from 5 images for each class to each combination of 2 out of 500 (=125k). Using this, you might be able to extract more useful features, and you can then train a classifier on top.

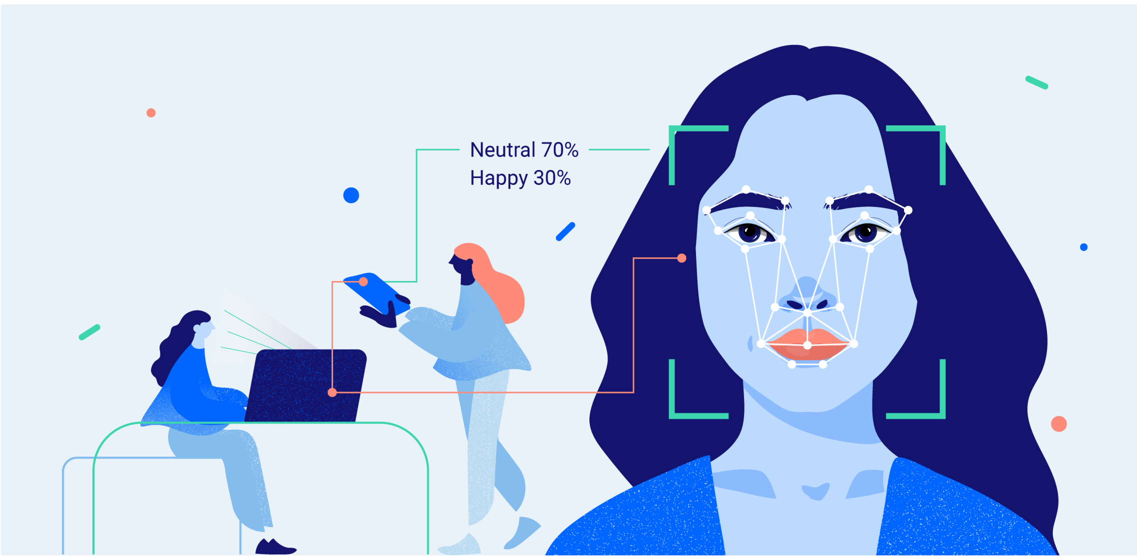


Radix

be discouraged if you don't immediately have the right data available. Instead, reflect on which of your use cases you might be able to tackle with AI and tackle the data aspect after. In that way, you can unlock the true potential of AI for your company. By being creative in the AI methods you use, you might be very surprised what you can achieve with only small efforts in your data!

If you might have some use cases worth looking into with AI but don't know where to start or how to use your data most optimally, reach out, and we'll be happy to find out together with you!

More posts by Radix



Why is Emotional AI not dead?

By Pierre Gerardi
January 16th, 2023

WIKIPEDIA
The Free Encyclopedia

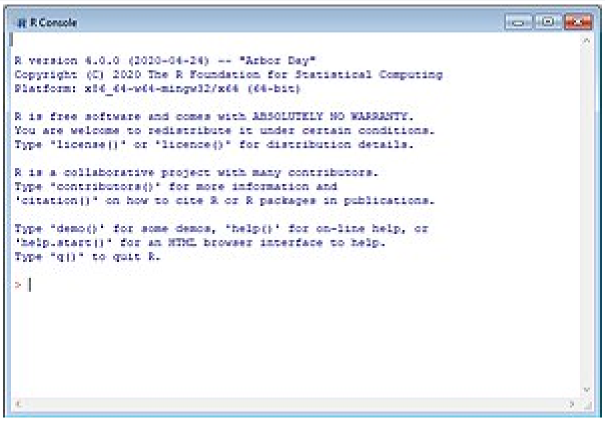
R (programming language)

R is a programming language for statistical computing and graphics supported by the R Core Team and the R Foundation for Statistical Computing. Created by statisticians Ross Ihaka and Robert Gentleman, R is used among data miners, bioinformaticians and statisticians for data analysis and developing statistical software.[7] Users have created packages to augment the functions of the R language.

According to user surveys and studies of scholarly literature databases, R is one of the most commonly used programming languages in data mining.[8] As of December 2022, R ranks 11th in the TIOBE index, a measure of programming language popularity, in which the language peaked in 8th place in August 2020.[9][10]

The official R software environment is an open-source free software environment within the GNU package, available under the GNU General Public License. It is written primarily in C, Fortran, and R itself (partially self-hosting). Precompiled executables are provided for various operating systems. R has a command line interface.[11] Multiple third-party graphical user interfaces are also available, such as RStudio, an integrated development environment, and Jupyter, a notebook interface.

R



R terminal

Paradigms	Multi-paradigm: procedural, object-oriented, functional, reflective, imperative, array[1]
Designed by	Ross Ihaka and Robert Gentleman
Developer	R Core Team
First appeared	August 1993
Stable release	4.2.3[2] / 15 March 2023
Typing discipline	Dynamic
Platform	arm64 and x86-64
License	GNU GPL v2
Filename extensions	.r[3] .rdata .rds .rda[4]
Website	www.r-project.org (https://www.r-project.org)

Influenced by

Lisp · S[5] · Scheme[1]

Influenced

History

R was started by professors Ross Ihaka and Robert Gentleman as a programming language to teach introductory statistics at the University of Auckland.[12] The language took heavy inspiration from the S programming language with most S programs able to run unaltered in R[5] as well as from Scheme's lexical scoping allowing for local variables.[1] The name of the language comes from being an S language successor and the shared first letter of the authors, Ross and Robert.[13] Ihaka and Gentleman first shared binaries of R on the data archive StatLib and the *s-news* mailing list in August 1993.[14] In June 1995, statistician Martin Mächler convinced Ihaka and Gentleman to make R free and open-source under the GNU General Public

License.[14][15] Mailing lists for the R project began on 1 April 1997 preceding the release of version 0.50.[16] R officially became a GNU project on 5 December 1997 when version 0.60 released.[17] The first official 1.0 version was released on 29 February 2000.[12]

Julia[6]
R Programming at Wikibooks

Robert Gentleman, co-originator of R

Ross Ihaka, co-originator of R

The Comprehensive R Archive Network (CRAN) was founded in 1997 by Kurt Hornik and Fritz Leisch to host R's source code, executable files, documentation, and user-created packages.[18] Its name and scope mimics the Comprehensive TeX Archive Network and the Comprehensive Perl Archive Network.[18] CRAN originally had three mirrors and 12 contributed packages.[19] As of December 2022, it has 103 mirrors[20] and 18,976 contributed packages.[21]

The R Core Team was formed in 1997 to further develop the language.[22][23] As of January 2022, it consists of Chambers, Gentleman, Ihaka, and Mächler, plus statisticians Douglas Bates, Peter Dalgaard, Kurt Hornik, Michael Lawrence, Friedrich Leisch, Uwe Ligges, Thomas Lumley, Sebastian Meyer, Paul Murrell, Martyn Plummer, Brian Ripley, Deepayan Sarkar, Duncan Temple Lang, Luke Tierney, and Simon Urbanek, as well as computer scientist Tomas Kalibera. Stefano Iacus, Guido Masarotto, Heiner Schwarte, Seth Falcon, Martin Morgan, and Duncan Murdoch were members.[14][24] In April 2003,[25] the R Foundation was founded as a non-profit organization to provide further support for the R project.[26]

Features

Data processing

R's data structures include vectors, arrays, lists, and data frames.[27] Vectors are ordered collections of values and can be mapped to arrays of one or more dimensions in a column major order. That is, given an ordered collection of dimensions, one fills in values along the first dimension first, then fills in one-dimensional arrays across the second dimension, and so on.[28] R supports array arithmetics and in this regard is like languages such as APL and MATLAB.[27][29] The special case of an array with two dimensions is called a matrix. Lists serve as collections of objects that do not necessarily have the same data type. Data frames contain a list of vectors of the same length, plus a unique set of row names.[27] R has no scalar data type.[30] Instead, a scalar is represented as a length-one vector.[31]

R and its libraries implement various statistical techniques, including linear, generalized linear and nonlinear modeling, classical statistical tests, spatial and time-series analysis, classification, clustering, and others. For computationally intensive tasks, C, C++, and Fortran code can be linked and called at run time. Another of R's strengths is static graphics; it can produce publication-quality graphs that include mathematical symbols.[32]

Programming

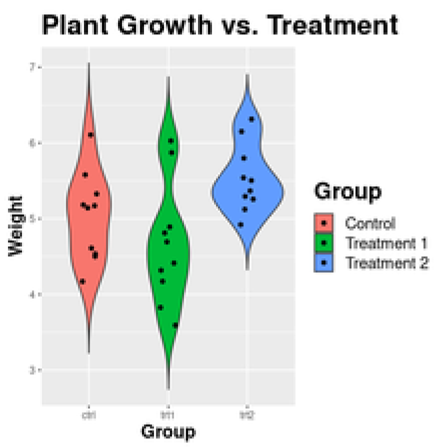
R is an interpreted language; users can access it through a command-line interpreter. If a user types `2+2` at the R command prompt and presses enter, the computer replies with `4`.

R supports procedural programming with functions and, for some functions, object-oriented programming with generic functions.[33] Due to its S heritage, R has stronger object-oriented programming facilities than most statistical computing languages. Extending it is facilitated by its lexical scoping rules, which are derived from Scheme.[34] R uses S syntax (not to be confused with S-expressions) to represent both data and code .[35] R's extensible object system includes objects for (among others): regression models, time-series and geo-spatial coordinates. Advanced users can write C, C++,[36] Java,[37] .NET[38] or Python code to manipulate R objects directly.[39]

Functions are first-class objects and can be manipulated in the same way as data objects, facilitating meta-programming that allows multiple dispatch. Function arguments are passed by value, and are lazy—that is to say, they are only evaluated when they are used, not when the function is called.[40] A generic function acts differently depending on the classes of the arguments passed to it. In other words, the generic function dispatches the method implementation specific to that object's class. For example, R has a generic `print` function that can print almost every class of object in R with `print(objectname)`.[41] R is highly extensible through the use of packages for specific functions and specific applications.

Packages

R's capabilities are extended through user-created[42] *packages*, which offer statistical techniques, graphical devices, import/export, reporting (RMarkdown, knitr, Sweave), etc. These packages and their easy installation and use has been cited as driving the language's widespread adoption in data science.[43][44][45][46][47] The packaging system is also used by researchers to organize research data, code, and report files in a systematic way for sharing and archiving.[48]



Violin plot created from the R visualization package ggplot2

Multiple packages are included with the basic installation. Additional packages are available on CRAN,[21] Bioconductor, R-Forge,[49] Omegahat,[50] GitHub, and other repositories.[51][52][53]

The "Task Views" on the CRAN website[54] lists packages in fields including Finance, Genetics, High-Performance Computing, Machine Learning, Medical Imaging, Meta-Analysis,[55] Social Sciences and Spatial Statistics.[55] R has been identified by the FDA as suitable for interpreting data from clinical research.[56] Microsoft maintains a daily snapshot of CRAN that dates back to Sept. 17, 2014.[57]

Other R package resources include R-Forge,[58][49] a platform for the collaborative development of R packages. The Bioconductor project provides packages for genomic data analysis, including object-oriented data handling and analysis tools for data from Affymetrix, cDNA microarray, and next-generation high-throughput sequencing methods.[59]

A group of packages called the Tidyverse, which can be considered a "dialect" of the R language, is increasingly popular among developers.[note 1] It strives to provide a cohesive collection of functions to deal with common data science tasks, including data import, cleaning, transformation, and visualisation (notably with the ggplot2 package). Dynamic and interactive graphics are available through additional packages.[60]

R is one of 5 languages with an Apache Spark API, along with Scala, Java, Python, and SQL.[61][62]

Interfaces

Early developers preferred to run R via the command line console,[63] succeeded by those who prefer an IDE.[64] IDEs for R include (in alphabetical order) R.app (https://cran.r-project.org/bin/macosx/) (OSX/macOS only), Rattle GUI, R Commander, RKWard, RStudio, and Tinn-R.[63] R is also supported in multi-purpose IDEs such as Eclipse via the StatET plugin,[65] and Visual Studio via the R Tools for Visual Studio.[66] Of these, RStudio is the most commonly used.[64]

Statistical frameworks which use R in the background include Jamovi and JASP.

Editors that support R include Emacs, Vim (Nvim-R plugin),[67] Kate,[68] LyX,[69] Notepad++,[70] Visual Studio Code, WinEdt,[71] and Tinn-R.[72] Jupyter Notebook can also be configured to edit and run R code.[73]

R functionality is accessible from scripting languages including Python,[74] Perl,[75] Ruby,[76] F#,[77] and Julia.[78] Interfaces to other, high-level programming languages, like Java[79] and .NET C#[80][81] are available.

Implementations

The main R implementation is written in R, C, and Fortran.[82] Several other implementations are aimed at improving speed or increasing extensibility. A closely related implementation is pqR (pretty quick R) by Radford M. Neal with improved memory management and support for automatic multithreading. Renjin and FastR are Java implementations of R for use in a Java Virtual Machine. CXXR, rho, and Riposte[83] are implementations of R in C++. Renjin, Riposte, and pqR attempt to improve performance by using multiple cores and deferred evaluation.[84] Most of these alternative implementations are experimental and incomplete, with relatively few users, compared to the main implementation maintained by the R Development Core Team.

TIBCO, who previous sold the commercial implementation S-PLUS, built a runtime engine called TERR, which is part of Spotfire.[85]

Microsoft R Open (MRO) is a fully compatible R distribution with modifications for multi-threaded computations.[86][87] As of 30 June 2021, Microsoft started to phase out MRO in favor of the CRAN distribution.[88]

Community

The R community hosts many conferences and in-person meetups. Some of these groups include:

- R-Ladies: an organization to promote gender diversity in the R community[89]
- UseR!: an annual international R user conference[90]
- SatRdays: R-focused conferences held on Saturdays[91]
- R Conference[92]
- Posit::conf (formerly known as Rstudio::conf)[93]

The R Foundation supports two conferences, useR! and Directions in Statistical Computing (DSC), and endorses several others like R@IIRSA, ConectaR, LatinR, and R Day.[90]

The R Journal

The R Journal is an open access, refereed journal of the R project. It features short to medium-length articles on the use and development of R, including packages, programming tips, CRAN news, and foundation news.

Comparison with alternatives

SAS

In January 2009, the *New York Times* ran an article charting the growth of R, noting its extensibility with user-created packages as well as R's open-source nature in contrast to SAS.[94] SAS supports Windows, UNIX, and z/OS.[95] R has precompiled binaries for Windows, macOS, and Linux with the option to compile and install R from source code.[96] SAS can only store data in rectangular data sets while R's more versatile data structures allow it to perform difficult analysis more flexibly. Completely integrating functions in SAS requires a developer's kit but, in R, user-defined functions are already on equal footing with provided functions.[97] In a technical report authored by Patrick Burns, respondents found R more convenient for periodic reports but preferred SAS for big data problems.[98]

Stata

Stata and R are designed to be easily extendable. Outputs in both software are structured to become inputs for further analysis. They hold data in main memory giving a performance boost but limiting data both can handle. R is free software while Stata is not.[99]

Python

Python and R are interpreted, dynamically typed programming languages with duck typing that can be extended by importing packages. Python is a general-purpose programming language while R is specifically designed for doing statistical analysis. Python has a BSD-like license in contrast to R's GNU General Public License but still permits modifying language implementation and tools.[100]

Commercial support

Although R is an open-source project, some companies provide commercial support and extensions.

In 2007, Richard Schultz, Martin Schultz, Steve Weston, and Kirk Mettler founded Revolution Analytics to provide commercial support for Revolution R, their distribution of R, which includes components developed by the company. Major additional components include ParallelR, the R Productivity Environment IDE, RevoScaleR (for big data analysis), RevoDeployR, web services framework, and the ability for reading and writing data in the SAS file format.[101] Revolution Analytics offers an R distribution designed to comply with established IQ/OQ/PQ criteria that enables clients in the pharmaceutical sector to validate their installation of REvolution R.[102] In 2015, Microsoft Corporation acquired Revolution Analytics[103] and integrated the R programming language into SQL Server, Power BI, Azure SQL Managed Instance, Azure Cortana Intelligence, Microsoft ML Server and Visual Studio 2017.[104]

In October 2011, Oracle announced the *Big Data Appliance*, which integrates R, Apache Hadoop, Oracle Linux, and a NoSQL database with Exadata hardware.[105] As of 2012, Oracle R Enterprise[106] became one of two components of the "Oracle Advanced Analytics Option"[107] (alongside Oracle Data Mining).

IBM offers support for in-Hadoop execution of R,[108] and provides a programming model for massively parallel in-database analytics in R.[109]

TIBCO offers a runtime-version R as a part of Spotfire.[110]

Mango Solutions offers a validation package for R, ValidR,[111][112] to comply with drug approval agencies, such as the FDA. These agencies required the use of validated software, as attested by the vendor or sponsor.[113]

Examples

Basic syntax

The following examples illustrate the basic syntax of the language and use of the command-line interface. (An expanded list of standard language features can be found in the R manual, "An Introduction to R".[114])

In R, the generally preferred assignment operator is an arrow made from two characters `<-`, although `=` can be used in some cases.[115][116]

```
> x <- 1:6 # Create a numeric vector in the current environment
> y <- x^2 # Create vector based on the values in x.
> print(y) # Print the vector’s contents.
[1]  1  4  9 16 25 36

> z <- x + y # Create a new vector that is the sum of x and y
> z # Return the contents of z to the current environment.
[1]  2  6 12 20 30 42

> z_matrix <- matrix(z, nrow=3) # Create a new matrix that turns the vector z into a 3x2 matrix object
> z_matrix
     [,1] [,2]
[1,]    2   20
[2,]    6   30
[3,]   12   42

> 2*t(z_matrix)-2 # Transpose the matrix, multiply every element by 2, subtract 2 from each element in the matrix,
and return the results to the terminal.
     [,1] [,2] [,3]
[1,]    2   10   22
[2,]   38   58   82

> new_df <- data.frame(t(z_matrix), row.names=c('A','B')) # Create a new data.frame object that contains the data
from a transposed z_matrix, with row names 'A' and 'B'
> names(new_df) <- c('X','Y','Z') # Set the column names of new_df as X, Y, and Z.
> print(new_df)  # Print the current results.
   X  Y  Z
A  2  6 12
B 20 30 42

> new_df$Z # Output the Z column
[1] 12 42
```

```
> new_df$Z==new_df['Z'] && new_df[3]==new_df$Z # The data.frame column Z can be accessed using $Z, ['Z'], or [3] syntax and the values are the same.
[1] TRUE

> attributes(new_df) # Print attributes information about the new_df object
$names
[1] "X" "Y" "Z"

$row.names
[1] "A" "B"

$class
[1] "data.frame"

> attributes(new_df)$row.names <- c('one','two') # Access and then change the row.names attribute; can also be done using rownames()
> new_df
     X  Y  Z
one  2  6 12
two 20 30 42
```

Structure of a function

One of R's strengths is the ease of creating new functions. Objects in the function body remain local to the function, and any data type may be returned.[117] Example:

```
# Declare function “f” with parameters “x”, “y“
# that returns a linear combination of x and y.
f <- function(x, y) {
  z <- 3 * x + 4 * y
  return(z) ## the return() function is optional here
}
```

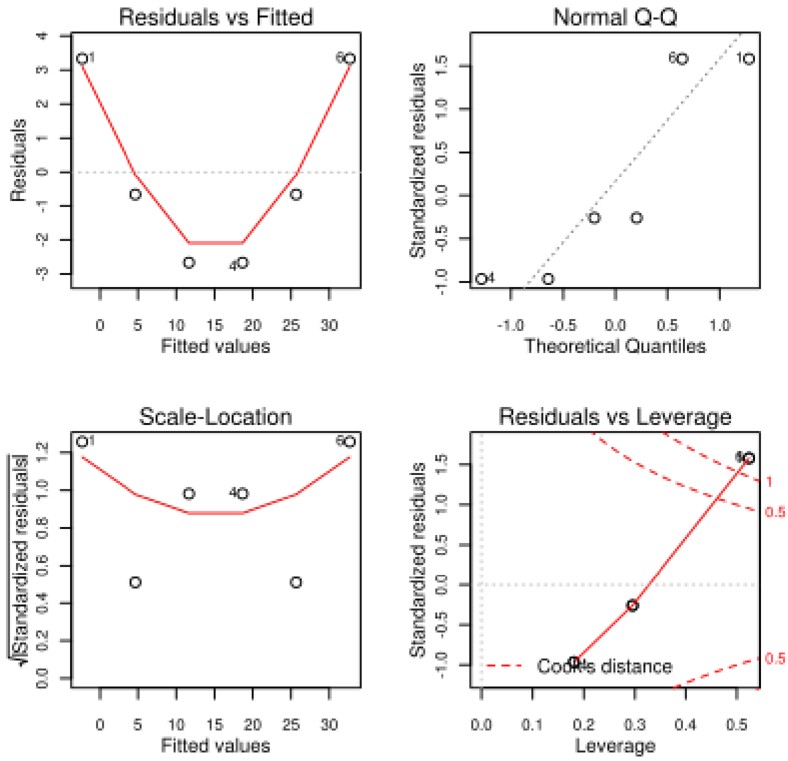
```
> f(1, 2)
[1] 11

> f(c(1,2,3), c(5,3,4))
[1] 23 18 25

> f(1:3, 4)
[1] 19 22 25
```

Modeling and plotting

The R language has built-in support for data modeling and graphics. The following example shows how R can easily generate and plot a linear model with residuals.

```
> x <- 1:6 # Create x and y values
> y <- x^2
> model <- lm(y ~ x)  # Linear regression model y = A + B * x.
> summary(model)  # Display an in-depth summary of the model.

Call:
lm(formula = y ~ x)

Residuals:
      1       2       3       4       5       6       7       8      9      10
 3.3333 -0.6667 -2.6667 -2.6667 -0.6667  3.3333

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -9.3333     2.8441  -3.282 0.030453 *
x             7.0000     0.7303   9.585 0.000662 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 3.055 on 4
degrees of freedom
Multiple R-squared:  0.9583, Adjusted R-
squared:  0.9478
F-statistic: 91.88 on 1 and 4 DF,  p-
value: 0.000662

> par(mfrow = c(2, 2))  # Create a 2 by 2
layout for figures.
> plot(model)  # Output diagnostic plots
of the model.
```



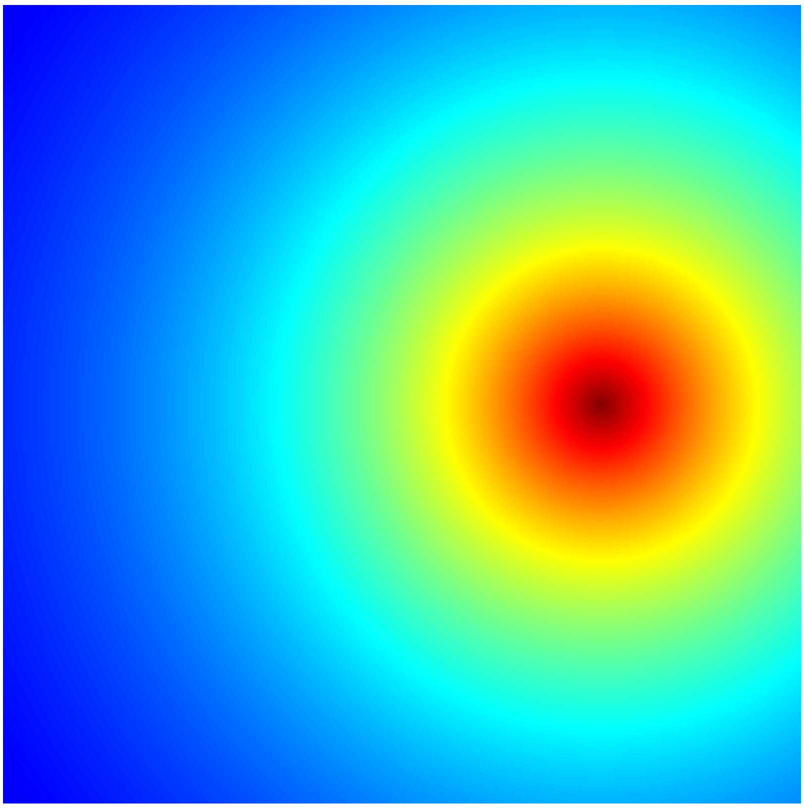
Diagnostic plots from plotting “model” (q.v. “plot.lm()” function). Notice the mathematical notation allowed in labels (lower left plot).

Mandelbrot set

Short R code calculating Mandelbrot set through the first 20 iterations of equation $z = z^2 + c$ plotted for different complex constants c. This example demonstrates:

- use of community-developed external libraries (called packages), like the caTools package
- handling of complex numbers
- multidimensional arrays of numbers used as basic data type, see variables C, Z, and X.

```
install.packages("caTools")  # install
external package
library(caTools)             # external
package providing write.gif function
jet.colors <- colorRampPalette(c("green",
"pink", "#007FFF", "cyan", "#7FFF7F",
                                 "white",
"#FF7F00", "red", "#7F0000"))
dx <- 1500                    # define
width
dy <- 1400                    # define
height
C  <- complex(real = rep(seq(-2.2, 1.0,
length.out = dx), each = dy),
              imag = rep(seq(-1.2, 1.2,
length.out = dy), dx))
C <- matrix(C, dy, dx)       # reshape as
square matrix of complex numbers
Z <- 0                       # initialize
Z to zero
X <- array(0, c(dy, dx, 20)) # initialize
output 3D array
for (k in 1:20) {            # loop with
20 iterations
  Z <- Z^2 + C               # the
central difference equation
  X[, , k] <- exp(-abs(Z))   # capture
results
}
write.gif(X, "Mandelbrot.gif", col =
jet.colors, delay = 100)
```



"Mandelbrot.gif" – graphics created in R with 14 lines of code

See also

- R package

- Comparison of numerical-analysis software
- Comparison of statistical packages
- List of numerical-analysis software
- List of statistical software
- Rmetrics

Notes

1. As of 13 June 2020, Metacran (https://www.r-pkg.org/downloaded) listed 7 of the 8 core packages of the Tidyverse in the list of most download R packages.

References

1. Morandat, Frances; Hill, Brandon; Osvald, Leo; Vitek, Jan (11 June 2012). "Evaluating the design of the R language: objects and functions for data analysis" (https://doi.org/10.1007/978-3-642-31057-7_6). *European Conference on Object-Oriented Programming*. **2012**: 104–131. doi:10.1007/978-3-642-31057-7_6 (https://doi.org/10.1007%2F978-3-642-31057-7_6). Retrieved 17 May 2016 – via SpringerLink.
2. Peter Dalgaard (15 March 2023). "R 4.2.3 is released" (https://hypatia.math.ethz.ch/pipermail/r-announce/2023/000689.html). Retrieved 15 March 2023.
3. "R scripts" (http://mercury.webster.edu/aleshunas/R_learning_infrastructure/R%20scripts.html). *mercury.webster.edu*. Retrieved 17 July 2021.
4. "R Data Format Family (.rdata, .rda)" (https://www.loc.gov/preservation/digital/formats/fdd/fdd000470.shtml). *Loc.gov*. 9 June 2017. Retrieved 17 July 2021.
5. Hornik, Kurt; The R Core Team (12 April 2022). "R FAQ" (https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-are-the-differences-between-R-and-S_003f). *The Comprehensive R Archive Network*. 3.3 What are the differences between R and S?. Archived (https://web.archive.org/web/20221228045640/https://cran.r-project.org/doc/FAQ/R-FAQ.html) from the original on 28 December 2022. Retrieved 27 December 2022.
6. "Introduction" (https://web.archive.org/web/20180620172516/https://docs.julialang.org/en/stable/manual/introduction/#man-introduction-1#man-introduction-1). *The Julia Manual*. Archived from the original (https://docs.julialang.org/en/stable/manual/introduction/#man-introduction-1) on 20 June 2018. Retrieved 5 August 2018.
7. Giorgi, Federico M.; Ceraolo, Carmine; Mercatelli, Daniele (27 April 2022). "The R Language: An Engine for Bioinformatics and Data Science" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9148156). *Life*. **12** (5): 648. doi:10.3390/life12050648 (https://doi.org/10.3390%2Flife12050648). ISSN 2075-1729 (https://www.worldcat.org/issn/2075-1729). PMC 9148156 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9148156). PMID 35629316 (https://pubmed.ncbi.nlm.nih.gov/35629316).
8. R's popularity
   - David Smith (2012); *R Tops Data Mining Software Poll* (https://www.r-bloggers.com/2012/05/r-tops-data-mining-software-poll/), R-bloggers, 31 May 2012.
   - Karl Rexer, Heather Allen, & Paul Gearan (2011); *2011 Data Miner Survey Summary* (https://www.kdnuggets.com/2012/07/rexer-analytics-2011-data-miner-survey-results.html), presented at Predictive Analytics World, Oct. 2011.
   - Robert A. Muenchen (2012). "The Popularity of Data Analysis Software" (http://r4stats.com/articles/popularity/).
   - Tippmann, Sylvia (29 December 2014). "Programming tools: Adventures with R" (https://doi.org/10.1038%2F517109a). *Nature*. **517** (7532): 109–110. doi:10.1038/517109a (https://doi.org/10.1038%2F517109a). PMID 25557714 (https://pubmed.ncbi.nlm.nih.gov/25557714).

9. "TIOBE Index - The Software Quality Company" (https://www.tiobe.com/tiobe-index/). *TIOBE*. Archived (https://web.archive.org/web/20221227091815/https://www.tiobe.com/tiobe-index/) from the original on 27 December 2022. Retrieved 27 December 2022.
10. "TIOBE Index: The R Programming Language" (https://www.tiobe.com/tiobe-index/r/). *TIOBE*. Retrieved 23 May 2022.
11. Kurt Hornik. *The R FAQ: Why R?* (https://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-is-R-named-R_003f). ISBN 3-900051-08-9. Retrieved 29 January 2008.
12. Ihaka, Ross. "The R Project: A Brief History and Thoughts About the Future" (https://www.stat.auckland.ac.nz/~ihaka/downloads/Otago.pdf) (PDF). Archived (https://web.archive.org/web/20221228043824/https://www.stat.auckland.ac.nz/~ihaka/downloads/Otago.pdf) (PDF) from the original on 27 December 2022. Retrieved 27 December 2022.
13. Hornik, Kurt; The R Core Team (12 April 2022). "R FAQ" (https://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-is-R-named-R_003f). *The Comprehensive R Archive Network*. 2.13 What is the R Foundation?. Archived (https://web.archive.org/web/20221228045640/https://cran.r-project.org/doc/FAQ/R-FAQ.html) from the original on 28 December 2022. Retrieved 28 December 2022.
14. Ihaka, Ross. "R: Past and Future History" (https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf) (PDF). Archived (https://web.archive.org/web/20221228071311/https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf) (PDF) from the original on 28 December 2022. Retrieved 28 December 2022.
15. GNU project
    - "GNU R" (http://directory.fsf.org/project/gnur/). Free Software Foundation. 23 April 2018. Archived (https://web.archive.org/web/20221228072338/https://directory.fsf.org/wiki/R) from the original on 28 December 2022. Retrieved 7 August 2018.
    - R Project. "What is R?" (https://www.r-project.org/about.html). Archived (https://web.archive.org/web/20221228072737/https://www.r-project.org/about.html) from the original on 28 December 2022. Retrieved 7 August 2018.
16. Maechler, Martin (1 April 1997). " "R-announce", "R-help", "R-devel" : 3 mailing lists for R" (https://stat.ethz.ch/pipermail/r-announce/1997/000000.html). *stat.ethz.ch*. Archived (https://web.archive.org/web/20221116161937/https://stat.ethz.ch/pipermail/r-announce/1997/000000.html) from the original on 12 February 2023. Retrieved 12 February 2023.
17. Ihaka, Ross (5 December 1997). "New R Version for Unix" (https://stat.ethz.ch/pipermail/r-announce/1997/000014.html). *stat.ethz.ch*. Archived (https://web.archive.org/web/20230212133422/https://stat.ethz.ch/pipermail/r-announce/1997/000014.html) from the original on 12 February 2023. Retrieved 12 February 2023.
18. Hornik, Kurt (2012). "The Comprehensive R Archive Network" (https://onlinelibrary.wiley.com/doi/10.1002/wics.1212). *WIREs Computational Statistics*. **4** (4): 394–398. doi:10.1002/wics.1212 (https://doi.org/10.1002%2Fwics.1212). ISSN 1939-5108 (https://www.worldcat.org/issn/1939-5108). S2CID 62231320 (https://api.semanticscholar.org/CorpusID:62231320).
19. Kurt Hornik (23 April 1997). "Announce: CRAN" (https://stat.ethz.ch/pipermail/r-announce/1997/000001.html). *r-help*. Wikidata Q101068595..
20. "The Status of CRAN Mirrors" (https://cran.r-project.org/mirmon_report.html). *cran.r-project.org*. Retrieved 30 December 2022.
21. "CRAN - Contributed Packages" (https://cran.r-project.org/web/packages/index.html). *cran.r-project.org*. Retrieved 29 December 2022.
22. Hornik, Kurt; R Core Team (12 April 2022). "R FAQ" (https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f). 2.1 What is R?. Archived (https://web.archive.org/web/20221228045640/https://cran.r-project.org/doc/FAQ/R-FAQ.html) from the original on 28 December 2022. Retrieved 29 December 2022.

23. Fox, John (2009). "Aspects of the Social Organization and Trajectory of the R Project" (https://journal.r-project.org/archive/2009/RJ-2009-014/index.html). *The R Journal*. **1** (2): 5. doi:10.32614/RJ-2009-014 (https://doi.org/10.32614%2FRJ-2009-014). ISSN 2073-4859 (https://www.worldcat.org/issn/2073-4859).
24. "R: Contributors" (https://www.r-project.org/contributors.html). *R Project*. Retrieved 14 July 2021.
25. Mächler, Martin; Hornik, Kurt (December 2014). "R Foundation News" (https://journal.r-project.org/archive/2014-2/r-foundation.pdf) (PDF). *The R Journal*. Archived (https://web.archive.org/web/20220411070133/https://journal.r-project.org/archive/2014-2/r-foundation.pdf) (PDF) from the original on 11 April 2022. Retrieved 30 December 2021.
26. Hornik, Kurt; R Core Team (12 April 2022). "R FAQ" (https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-the-R-Foundation_003f). 2.13 What is the R Foundation?. Archived (https://web.archive.org/web/20221228045640/https://cran.r-project.org/doc/FAQ/R-FAQ.html) from the original on 28 December 2022. Retrieved 29 December 2022.
27. Dalgaard, Peter (2002). *Introductory Statistics with R* (https://archive.org/details/introductorystat00dalg_885). New York, Berlin, Heidelberg: Springer-Verlag. pp. 10 (https://archive.org/details/introductorystat00dalg_885/page/n26)–18, 34. ISBN 0387954759.
28. *An Introduction to R*, Section 5.1: Arrays. Retrieved in 2010-03 from https://cran.r-project.org/doc/manuals/R-intro.html#Arrays.
29. Chen, Han-feng; Wai-mee, Ching; Da, Zheng. "A Comparison Study on Execution Performance of MATLAB and APL" (http://www.sable.mcgill.ca/~hanfeng.c/eli/res/Comparison.pdf) (PDF). *McGill University*. Retrieved 16 February 2022.
30. Ihaka, Ross; Gentlman, Robert (September 1996). "R: A Language for Data Analysis and Graphics" (https://www.stat.auckland.ac.nz/~ihaka/downloads/R-paper.pdf) (PDF). *Journal of Computational and Graphical Statistics*. American Statistical Association. **5** (3): 299–314. doi:10.2307/1390807 (https://doi.org/10.2307%2F1390807). JSTOR 1390807 (https://www.jstor.org/stable/1390807). Retrieved 12 May 2014.
31. "Data structures · Advanced R." (http://adv-r.had.co.nz/Data-structures.html) *adv-r.had.co.nz*. Retrieved 26 September 2016.
32. "R: What is R?" (https://www.r-project.org/about.html). *R-project.org*. Retrieved 17 February 2022.
33. White, Homer. *14.1 Programming Paradigms | Beginning Computer Science with R* (https://homerhanumat.github.io/r-notes/programming-paradigms.html).
34. Jackman, Simon (Spring 2003). "R For the Political Methodologist" (https://web.archive.org/web/20060721143309/http://polmeth.wustl.edu/tpm/tpm_v11_n2.pdf) (PDF). *The Political Methodologist*. Political Methodology Section, American Political Science Association. **11** (1): 20–22. Archived from the original (http://polmeth.wustl.edu/tpm/tpm_v11_n2.pdf) (PDF) on 21 July 2006. Retrieved 13 September 2018.
35. "An Introduction to R: 1.2 Related software and documentation" (https://cran.r-project.org/doc/manuals/r-release/R-intro.html#Related-software-and-documentation-1). Retrieved 14 January 2023.
36. Eddelbuettel, Dirk; Francois, Romain (2011). "Rcpp: Seamless R and C++ Integration" (https://doi.org/10.18637%2Fjss.v040.i08). *Journal of Statistical Software*. **40** (8). doi:10.18637/jss.v040.i08 (https://doi.org/10.18637%2Fjss.v040.i08).
37. "nution-j2r: Java library to invoke R native functions" (https://gitlab.nuiton.org/nuiton/nuiton-j2r). Retrieved 13 September 2018.

38. .NET Framework
    - "Making GUIs using C# and R with the help of R.NET" (http://psychwire.wordpress.com/2011/06/19/making-guis-using-c-and-r-with-the-help-of-r-net/). 19 June 2011. Retrieved 13 September 2018.
    - "R.NET homepage" (https://web.archive.org/web/20151013020201/http://rdotnet.codeplex.com/). Archived from the original (http://rdotnet.codeplex.com/) on 13 October 2015. Retrieved 13 September 2018.
    - Haynold, Oliver M. (April 2011). *An Rserve Client Implementation for CLI/.NET* (https://web.archive.org/web/20151129223447/http://www.rinfinance.com/agenda/2011/OliverHaynold.pdf) (PDF). R/Finance 2011 (http://www.rinfinance.com/RinFinance2011/agenda/). Chicago, IL, USA. Archived from the original (http://www.rinfinance.com/agenda/2011/OliverHaynold.pdf) (PDF) on 29 November 2015. Retrieved 13 September 2018.
39. R manuals. "Writing R Extensions" (https://cran.r-project.org/doc/manuals/r-release/R-exts.html). *r-project.org*. Retrieved 13 September 2018.
40. "Functions · Advanced R." (http://adv-r.had.co.nz/Functions.html#lazy-evaluation) *adv-r.had.co.nz*.
41. R Core Team. "Print Values" (https://stat.ethz.ch/R-manual/R-devel/library/base/html/print.html). *R Documentation*. R Foundation for Statistical Computing. Retrieved 30 May 2016.
42. Hadley, Wickham; Bryan, Jenny. "R packages: Organize, Test, Document, and Share Your Code" (https://r-pkgs.org/).
43. Chambers, John M. (2020). "S, R, and Data Science" (https://journal.r-project.org/archive/2020/RJ-2020-028/index.html). *The R Journal*. **12** (1): 462–476. doi:10.32614/RJ-2020-028 (https://doi.org/10.32614%2FRJ-2020-028). ISSN 2073-4859 (https://www.worldcat.org/issn/2073-4859).
44. Vance, Ashlee (6 January 2009). "Data Analysts Captivated by R's Power" (https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html). *New York Times*.
45. Tippmann, Sylvia (29 December 2014). "Programming tools: Adventures with R" (https://doi.org/10.1038%2F517109a). *Nature News*. **517** (7532): 109–110. doi:10.1038/517109a (https://doi.org/10.1038%2F517109a). PMID 25557714 (https://pubmed.ncbi.nlm.nih.gov/25557714).
46. Thieme, Nick (2018). "R generation" (https://doi.org/10.1111%2Fj.1740-9713.2018.01169.x). *Significance*. **15** (4): 14–19. doi:10.1111/j.1740-9713.2018.01169.x (https://doi.org/10.1111%2Fj.1740-9713.2018.01169.x). ISSN 1740-9713 (https://www.worldcat.org/issn/1740-9713).
47. widely used
    - Fox, John & Andersen, Robert (January 2005). "Using the R Statistical Computing Environment to Teach Social Statistics Courses" (https://socialsciences.mcmaster.ca/jfox/Teaching-with-R.pdf) (PDF). Department of Sociology, McMaster University. Retrieved 6 August 2018.
    - Vance, Ashlee (6 January 2009). "Data Analysts Captivated by R's Power" (https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html). *New York Times*. Retrieved 6 August 2018. "R is also the name of a popular programming language used by a growing number of data analysts inside corporations and academia. It is becoming their lingua franca..."
48. Marwick, Ben; Boettiger, Carl; Mullen, Lincoln (26 August 2017). "Packaging data analytical work reproducibly using R (and friends)" (https://peerj.com/preprints/3192/?td=wk). *PeerJ Preprints*. doi:10.7287/peerj.preprints.3192v1 (https://doi.org/10.7287%2Fpeerj.preprints.3192v1). ISSN 2167-9843 (https://www.worldcat.org/issn/2167-9843).
49. Theußl, Stefan; Zeileis, Achim (2009). "Collaborative Software Development Using R-Forge" (https://journal.r-project.org/archive/2009/RJ-2009-007/index.html). *The R Journal*. **1** (1): 9. doi:10.32614/RJ-2009-007 (https://doi.org/10.32614%2FRJ-2009-007). ISSN 2073-4859 (https://www.worldcat.org/issn/2073-4859).
50. "Omegahat.net" (http://www.omegahat.net/). Omegahat.net. Retrieved 16 September 2018.

51. packages available from repositories
   - Robert A. Muenchen (2012). "The Popularity of Data Analysis Software" (http://r4stats.com/articles/popularity/).
   - Tippmann, Sylvia (29 December 2014). "Programming tools: Adventures with R" (https://doi.org/10.1038%2F517109a). *Nature*. **517** (7532): 109–110. doi:10.1038/517109a (https://doi.org/10.1038%2F517109a). PMID 25557714 (https://pubmed.ncbi.nlm.nih.gov/25557714).
   - "Search all R packages and function manuals | Rdocumentation" (http://www.rdocumentation.org/). *Rdocumentation*. 16 June 2014. Retrieved 16 September 2018.
52. Wickham, Hadley; Bryan, Jennifer. *Chapter 10 Object documentation | R Packages* (https://r-pkgs.org/man.html).
53. "Rd formatting" (https://cran.r-project.org/web/packages/roxygen2/vignettes/rd-formatting.html). *cran.r-project.org*. Retrieved 16 August 2021.
54. "CRAN Task Views" (https://cran.r-project.org/web/views/). *cran.r-project.org*. Retrieved 16 September 2018.
55. Hornik, Kurt; Zeileis, Achim (2013). "Changes on CRAN" (http://journal.r-project.org/archive/2013-1/cran.pdf) (PDF). *The R Journal*. **5** (1): 239–264.
56. "FDA: R OK for drug trials" (http://blog.revolutionanalytics.com/2012/06/fda-r-ok.html). Retrieved 16 September 2018.
57. "CRAN Time Machine. MRAN" (https://mran.microsoft.com/timemachine). Retrieved 26 December 2019.
58. "R-Forge: Welcome" (https://r-forge.r-project.org/). Retrieved 16 September 2018.
59. Huber, W; Carey, VJ; Gentleman, R; Anders, S; Carlson, M; Carvalho, BS; Bravo, HC; Davis, S; Gatto, L; Girke, T; Gottardo, R; Hahne, F; Hansen, KD; Irizarry, RA; Lawrence, M; Love, MI; MacDonald, J; Obenchain, V; Oleś, AK; Pagès, H; Reyes, A; Shannon, P; Smyth, GK; Tenenbaum, D; Waldron, L; Morgan, M (2015). "Orchestrating high-throughput genomic analysis with Bioconductor" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4509590). *Nature Methods*. Nature Publishing Group. **12** (2): 115–121. doi:10.1038/nmeth.3252 (https://doi.org/10.1038%2Fnmeth.3252). PMC 4509590 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4509590). PMID 25633503 (https://pubmed.ncbi.nlm.nih.gov/25633503).
60. Lewin-Koh, Nicholas (7 January 2015). "CRAN Task View: Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization" (https://web.archive.org/web/20160926233106/https://cran.r-project.org/web/views/Graphics.html). Archived from the original (https://cran.r-project.org/web/views/Graphics.html) on 26 September 2016. Retrieved 27 December 2022.
61. "Spark API Documentation" (https://spark.apache.org/docs/latest/api.html). *Spark*.
62. "SparkR (R on Spark)" (https://spark.apache.org/docs/latest/sparkr.html). *Spark*.
63. "Poll: R GUIs you use frequently (2011)" (http://www.kdnuggets.com/polls/2011/r-gui-used.html). *kdnuggets.com*. Retrieved 18 September 2018.
64. "R Programming - The State of Developer Ecosystem in 2020 Infographic" (https://www.jetbrains.com/lp/devecosystem-2020/). *JetBrains: Developer Tools for Professionals and Teams*. Retrieved 16 August 2021.
65. Stephan Wahlbrink. "StatET for R" (https://marketplace.eclipse.org/content/statet-r/).
66. "Work with R in Visual Studio" (https://docs.microsoft.com/en-us/visualstudio/rtvs/?view=vs-2017). Retrieved 14 December 2020.
67. "Nvim-R - Plugin to work with R : vim online" (https://www.vim.org/scripts/script.php?script_id=2628). *Vim.org*. Retrieved 6 March 2019.
68. "Syntax Highlighting" (https://web.archive.org/web/20080707062903/http://www.kate-editor.org/downloads/syntax_highlighting). Kate Development Team. Archived from the original (http://kate-editor.org/downloads/syntax_highlighting) on 7 July 2008. Retrieved 9 July 2008.
69. Paul E. Johnson & Gregor Gorjanc. "LyX with R through Sweave" (http://wiki.lyx.org/LyX/LyxWithRThroughSweave). Retrieved 4 April 2017.

70. "NppToR: R in Notepad++" (http://sourceforge.net/projects/npptor/). sourceforge.net. 8 May 2013. Retrieved 18 September 2013.
71. Uwe Ligges (5 January 2017). "RWinEdt: R Interface to 'WinEdt' " (https://cran.r-project.org/web/packages/RWinEdt/index.html). Retrieved 4 April 2017.
72. "Tinn-R" (https://nbcgib.uesc.br/tinnr/en/). Retrieved 5 March 2019.
73. "Using the R programming language in Jupyter Notebook" (https://docs.anaconda.com/anaconda/navigator/tutorials/r-lang/). *Anaconda*. Retrieved 14 September 2020.
74. Gautier, Laurent. "rpy2 - R in Python" (https://rpy2.github.io). Archived (https://web.archive.org/web/20221227092503/https://rpy2.github.io/) from the original on 27 December 2022. Retrieved 30 November 2021.
75. Florent Angly. "Statistics::R - Perl interface with the R statistical program" (https://metacpan.org/module/Statistics::R). *Metacpan.org*.
76. Alex Gutteridge (15 July 2021). "GitHub - alexgutteridge/rsruby: Ruby - R bridge" (https://github.com/alexgutteridge/rsruby). *Github.com*.
77. BlueMountain Capital. "F# R Type Provider" (https://bluemountaincapital.github.io/FSharpRProvider/).
78. "JuliaInterop/RCall.jl" (https://github.com/JuliaInterop/RCall.jl). *Github.com*. 2 June 2021.
79. "Rserve - Binary R server - RForge.net" (https://www.rforge.net/Rserve/). *Rforge.net*.
80. "konne/RserveCLI2" (https://github.com/konne/RserveCLI2). *Github.com*. 8 March 2021.
81. "R.NET" (https://jmp75.github.io/rdotnet/). *Jmp75.github.io*. Retrieved 18 February 2022.
82. "r-source: Read only mirror of R source code on GitHub" (https://github.com/wch/r-source). *GitHub*. Retrieved 14 September 2019.
83. Talbot, Justin; DeVito, Zachary; Hanrahan, Pat (1 January 2012). "Riposte: A Trace-driven Compiler and Parallel VM for Vector Code in R". *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*. ACM: 43–52. doi:10.1145/2370816.2370825 (https://doi.org/10.1145%2F2370816.2370825). S2CID 1989369 (https://api.semanticscholar.org/CorpusID:1989369).
84. Neal, Radford (25 July 2013). "Deferred evaluation in Renjin, Riposte, and pqR" (https://radfordneal.wordpress.com/2013/07/24/deferred-evaluation-in-renjin-riposte-and-pqr/). *Radford Neal's blog*. Retrieved 6 March 2017.
85. Jackson, Joab (16 May 2013). TIBCO offers free R to the enterprise (http://www.pcworld.com/article/2038944/tibco-offers-free-r-to-the-enterprise.html). *PC World*. Retrieved 20 July 2015.
86. "Home" (https://mran.microsoft.com/). *mran.microsoft.com*. Retrieved 22 November 2021.
87. "Microsoft R Open: The Enhanced R Distribution" (https://mran.microsoft.com/open). Retrieved 30 June 2018.
88. "Looking to the future for R in Azure SQL and SQL Server" (https://cloudblogs.microsoft.com/sqlserver/2021/06/30/looking-to-the-future-for-r-in-azure-sql-and-sql-server/). 30 June 2021. Retrieved 7 November 2021.
89. "R Ladies" (https://rladies.org). *R Ladies*. Retrieved 12 May 2018.
90. "Conferences" (https://www.r-project.org/conferences/). *The R Project for Statistical Computing*. Archived (https://web.archive.org/web/20221229210509/https://www.r-project.org/conferences/) from the original on 29 December 2022. Retrieved 29 December 2022.
91. "satRdays" (https://satrdays.org/). *satrdays.org*. Archived (https://web.archive.org/web/20221229222603/https://satrdays.org/) from the original on 29 December 2022. Retrieved 29 December 2022.
92. Bighetti, Nelson. "R Conference" (https://rstats.ai/). *R Conference*. Archived (https://web.archive.org/web/20221229222952/https://rstats.ai/) from the original on 29 December 2022. Retrieved 29 December 2022.
93. "posit::conf" (https://www.posit.co/conference/). *Posit*. Archived (https://web.archive.org/web/20221229215714/https://posit.co/conference/) from the original on 29 December 2022. Retrieved 29 December 2022.

94. R as competition for commercial statistical packages

   - Vance, Ashlee (7 January 2009). "Data Analysts Are Mesmerized by the Power of Program R: [Business/Financial Desk]" (https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html). *The New York Times*.
   - Vance, Ashlee (8 January 2009). "R You Ready for R?" (http://bits.blogs.nytimes.com/2009/01/08/r-you-ready-for-r/). *The New York Times*.

95. "SAS Supported Operating Systems" (https://support.sas.com/supportos/list). *support.sas.com*. Retrieved 31 December 2022.
96. Hornik, Kurt; R Core Team (12 April 2022). "R FAQ" (https://cran.r-project.org/doc/FAQ/R-FAQ.html#How-can-R-be-installed_003f). How can R be installed?. Archived (https://web.archive.org/web/20221229022750/https://cran.r-project.org/doc/FAQ/R-FAQ.html) from the original on 29 December 2022. Retrieved 31 December 2022.
97. Muenchen, Robert A. (2011). *R for SAS and SPSS Users* (http://link.springer.com/10.1007/978-1-4614-0685-3). Statistics and Computing (2nd ed.). New York, NY: Springer New York. pp. 2–3. doi:10.1007/978-1-4614-0685-3 (https://doi.org/10.1007%2F978-1-4614-0685-3). ISBN 978-1-4614-0684-6.
98. Burns, Patrick (27 February 2007). "R Relative to Statistical Packages: Comment 1 on Technical Report Number 1 (Version 1.0) Strategically using General Purpose Statistics Packages: A Look at Stata, SAS and SPSS" (https://www.burns-stat.com/pages/Tutor/R_relative_statpack.pdf) (PDF). Archived (https://web.archive.org/web/20221231060847/https://www.burns-stat.com/pages/Tutor/R_relative_statpack.pdf) (PDF) from the original on 31 December 2022. Retrieved 31 December 2022.
99. Muenchen, Robert A.; Hilbe, Joseph M. (2010). *R for Stata Users* (http://link.springer.com/10.1007/978-1-4419-1318-0). Statistics and Computing. New York, NY: Springer New York. pp. 2–3. doi:10.1007/978-1-4419-1318-0 (https://doi.org/10.1007%2F978-1-4419-1318-0). ISBN 978-1-4419-1317-3.
100. Grogan, Michael (2018). *Python vs. R for Data Science*. O'Reilly Media, Inc.
101. Morgan, Timothy Prickett (2011-02-07). "'Red Hat for stats' goes toe-to-toe with SAS". The Register, 7 February 2011. Retrieved from https://www.theregister.co.uk/2011/02/07/revolution_r_sas_challenge/.
102. "Analyzing clinical trial data for FDA submissions with R" (http://blog.revolutionanalytics.com/2009/01/analyzing-clinical-trial-data-with-r.html). Revolution Analytics. 14 January 2009. Retrieved 20 September 2018.
103. Sirosh, Joseph. "Microsoft Closes Acquisition of Revolution Analytics" (http://blogs.technet.com/b/machinelearning/archive/2015/04/06/microsoft-closes-acquisition-of-revolution-analytics.aspx). *blogs.technet.com*. Microsoft. Retrieved 20 September 2018.
104. "Introducing R Tools for Visual Studio" (https://blogs.msdn.microsoft.com/visualstudio/2016/03/22/introducing-r-tools-for-visual-studio-3/). Retrieved 20 September 2018.
105. Oracle Corporation's Big Data Appliance

   - Doug Henschen (2012); *Oracle Makes Big Data Appliance Move With Cloudera* (http://www.informationweek.com/software/information-management/oracle-makes-big-data-appliance-move-wit/232400021), InformationWeek, 10 January 2012.
   - Jaikumar Vijayan (2012); *Oracle's Big Data Appliance brings focus to bundled approach* (http://www.computerworld.com/s/article/9223325/Oracle_s_Big_Data_Appliance_brings_focus_to_bundled_approach), ComputerWorld, 11 January 2012.
   - Timothy Prickett Morgan (2011); Oracle rolls its own NoSQL and Hadoop *Oracle rolls its own NoSQL and Hadoop* (https://www.theregister.co.uk/2011/10/03/oracle_big_data_appliance/), The Register, 3 October 2011.

106. Chris Kanaracus (2012); *Oracle Stakes Claim in R With Advanced Analytics Launch* (http://www.pcworld.com/article/249509/oracle_stakes_claim_in_r_with_advanced_analytics_launch.html), PC World, February 8, 2012.

107. Doug Henschen (2012); *Oracle Stakes Claim in R With Advanced Analytics Launch* (http://www.informationweek.com/software/business-intelligence/oracle-makes-its-big-play-for-analytics/232800252), InformationWeek, April 4, 2012.
108. "What's New in IBM InfoSphere BigInsights v2.1.2" (https://web.archive.org/web/20140906200802/http://www-01.ibm.com/software/data/infosphere/biginsights/whats_new.html). IBM. Archived from the original (http://www-01.ibm.com/software/data/infosphere/biginsights/whats_new.html) on 6 September 2014. Retrieved 8 May 2014.
109. "IBM PureData System for Analytics" (https://web.archive.org/web/20140517153029/http://mainline.com/_web/_shared/pdfs/brochures/IBM-PureData-System-Overview.pdf) (PDF). IBM. Archived from the original (http://mainline.com/_web/_shared/pdfs/brochures/IBM-PureData-System-Overview.pdf) (PDF) on 17 May 2014. Retrieved 8 May 2014.
110. Tibco. "Unleash the agility of R for the Enterprise" (http://spotfire.tibco.com/discover-spotfire/what-does-spotfire-do/predictive-analytics/tibco-enterprise-runtime-for-r-terr). Retrieved 15 May 2014.
111. "ValidR on Mango website" (https://www.mango-solutions.com/products/validr/). Retrieved 24 September 2018.
112. Andy Nicholls at Mango Solutions. "ValidR Enterprise: Developing an R Validation Framework" (https://www.lexjansen.com/phuse/2016/ad/AD12.pdf) (PDF). Retrieved 24 September 2018.
113. FDA. "Statistical Software Clarifying Statement" (https://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM587506.pdf) (PDF). *Food and Drug Administration*. Retrieved 24 September 2018.
114. "An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics" (https://cran.r-project.org/doc/manuals/R-intro.pdf) (PDF). Retrieved 3 January 2021.
115. R Development Core Team. "Assignments with the = Operator" (https://developer.r-project.org/equalAssign.html). Retrieved 11 September 2018.
116. most used assignment operator in R is `<-`
    - R Development Core Team. "Writing R Extensions" (https://cran.r-project.org/doc/manuals/R-exts.html#Tidying-R-code). Retrieved 11 September 2018. "[...] we recommend the consistent use of the preferred assignment operator '<-' (rather than '=') for assignment."
    - "Google's R Style Guide" (https://google.github.io/styleguide/Rguide.xml#assignment). Retrieved 11 September 2018.
    - Wickham, Hadley. "Style Guide" (http://stat405.had.co.nz/r-style.html). Retrieved 11 September 2018.
    - Bengtsson, Henrik (January 2009). "R Coding Conventions (RCC) – a draft" (https://docs.google.com/document/preview?id=1esDVxyWvH8AsX-VJa-8oqWaHLs4stGlIbk8kLc5VlII&pli=1). Retrieved 11 September 2018.
117. Kabacoff, Robert (2012). "Quick-R: User-Defined Functions" (http://www.statmethods.net/management/userfunctions.html). *statmethods.net*. Retrieved 28 September 2018.

External links

- Official website (https://www.r-project.org) of the R project
- R Technical Papers (https://developer.r-project.org/TechDocs/)

Retrieved from "https://en.wikipedia.org/w/index.php?title=R_(programming_language)&oldid=1144363470"