

Learn R Programming in 1 Day

By Krishna Rungta

Copyright 2019 - All Rights Reserved – Krishna Rungta

ALL RIGHTS RESERVED. No part of this publication may be reproduced or transmitted in any form whatsoever, electronic, or mechanical, including photocopying, recording, or by any informational storage or retrieval system without express written, dated and signed permission from the author.

Table of content

Chapter 1.What is R Programming? Introduction Basics

1. [What is R?](#)
2. [R by industry](#)
3. [R package](#)
4. [Communicate with R](#)
5. [Why use R?](#)
6. [Is R difficult?](#)

Chapter 2.How to Download and Install R, RStudio, and Anaconda on Mac/Windows

1. [Install Anaconda](#)
2. [For MAC User](#)
3. [For Windows User](#)
4. [Install R](#)
5. [Install RStudio](#)
6. [Install Package](#)

Chapter 3. R Data Types & Operator

1. [Basic Data Types](#)
2. [Variables](#)
3. [Vectors](#)
4. [Arithmetic Operations](#)
5. [Logical Statements](#)

Chapter 4. R Matrix Tutorial: Create,Print,add Column,Slice

1. [What is a Matrix?](#)
2. [Slice a Matrix](#)

Chapter 5. What is Factor in R? Categorical &Continuous

1. [What is Factor in R?](#)
2. [Categorical Variable](#)
3. [Continuous Variables](#)

Chapter 6. R Data Frames: Create, Append, Select, Subset

1. [What is a Data Frame?](#)
2. [How to create a data frame?](#)
3. [Slice Data Frame](#)
4. [Subset a data frame](#)

Chapter 7. Lists in R: Create, Select [Example]

1. [What is a List?](#)
2. [Built-in Data Frame](#)

Chapter 8. Sort a data frame using Order

Chapter 9. R Dplyr Tutorial: Merge, Join, Spread

1. [Introduction to Data Analysis](#)
2. [Merge with dplyr\(\)](#)
3. [Multiple keys pairs](#)
4. [Data cleaning Functions](#)

Chapter 10. Merge data Frames in R: Full and Partial Match

1. [Full Match](#)
2. [Partial Match](#)

Chapter 11. Function in R Programming [With Example]

1. [What is a Function in R?](#)
2. [R important built-in functions](#)
3. [General Functions](#)
4. [Math Functions](#)

5. [Statistical Function](#)
6. [Write function in R](#)
7. [Function with condition](#)

Chapter 12. IF,ELSE,ELIF, Statement in R

1. [The If-Else Statement](#)
2. [The Else-If Statement](#)

Chapter 13. For loop in R with Examples for List and Matrix

1. [For Loop Syntax and Examples](#)
2. [For Loop over a List](#)
3. [For Loop over a Matrix](#)

Chapter 14. While loop in R with Example

Chapter 15. apply(),sapply(),tapply() in R with Example

1. [What occasions to use apply\(\) Collection](#)
2. [lapply\(\) Function](#)
3. [use sapply\(\) Function](#)
4. [tapply\(\) Function](#)

Chapter 16. Import data into R: Read CSV, Excel, SPSS, Stata, SAS Files

1. [Read CSV](#)
2. [Read Excel files](#)
3. [Read SAS](#)

Chapter 17. Replace missing Values(NA)in R

1. [Mutate\(\)](#)
2. [Exclude missing values](#)
3. [Impute missing values](#)

Chapter 18. R Exporting Data to CSV, Excel, SAS, STATA, and Text File

1. [How to Export Data from R](#)
2. [Export to Hard Drive](#)
3. [Export CSV](#)
4. [Export Excel File](#)
5. [Export to different software](#)
6. [Export SAS File](#)
7. [Interact with the Cloud Services](#)
8. [Import From Google Drive](#)
9. [Export to Dropbox](#)

Chapter 19. Pearson, Spearman Correlation in R and Correlation Matrix

1. [Pearson](#)
2. [Spearman Rank Correlation](#)
3. [Correlation matrix](#)
4. [Visualize correlation matrix](#)
5. [The Function ggcorr](#)

Chapter 20. R Aggregate Function: Summarise & Group by() [Example]

1. [Group by VS noGroup by](#)
2. [Function withsummarise\(\)](#)
3. [Multiple Groups](#)

Chapter 21. R Select(), Filter(), Arrange(), Pipeline [Example]

1. [Select\(\)](#)
2. [Filter\(\)](#)
3. [One Criteria](#)
4. [Multiple Criteria](#)

5. [Pipeline](#)

Chapter 22. Scatter plot with ggplot2 [Examples]

1. [Ggplot2 Package](#)
2. [Scatterplot](#)
3. [Change Axis](#)
4. [Add information to Graph](#)
5. [Theme](#)
6. [Save Plots](#)

Chapter 23. How to make Boxplot in R (with EXAMPLE)

1. [Basic Box Plot](#)
2. [Box Plot with Dots](#)
3. [Control Aesthetic of The Box Plot](#)
4. [Box Plot with Multiple Groups](#)
5. [Box Plot with jittered dots](#)
6. [Notched Box Plot](#)

Chapter 24. Bar Chart & Histogram in R [Examples]

1. [Bar Chat](#)
2. [Customize the Graph](#)
3. [Bar chart in percentage](#)
4. [Side by Side Bars](#)
5. [Histogram](#)

Chapter 25. T-Test in R: One-Sample and Paired [Example]

1. [What is Statistical Inference?](#)
2. [What is T-Test?](#)
3. [One sample t-test](#)
4. [Paired t-test](#)

Chapter 26. R ANOVA Tutorial: One-way & Two-way [with Examples]

1. [What is ANOVA?](#)
2. [One-way ANOVA](#)
3. [Pairwise Comparison](#)

Chapter 27. R Simple, Multiple Linear and Stepwise Regression [with Example]

1. [Machine Learning](#)
2. [Supervised Learning](#)
3. [Unsupervised Learning](#)
4. [Simple Linear Regression](#)
5. [Continuous Variable](#)
6. [Factors Regression](#)
7. [Stepwise Regression](#)

Chapter 28. Decision trees in R with Example

1. [What are Decision trees?](#)
2. [Training and Visualizing a decision trees](#)

Chapter 29. R Random Forest Tutorial [Examples]

1. [Random Forests](#)
2. [Grid Search Definition](#)
3. [Random Search Definition](#)
4. [Appendix](#)

Chapter 30. Generalized Linear Model (GLM) in R with Example

1. [What is Logistic Regression?](#)
2. [Train the Model](#)
3. [Non-Linearity](#)
4. [Correlation](#)
5. [The ROC Curve](#)

Chapter 31. K-means clustering in R with Example

1. [What is cluster Analysis?](#)
2. [K-Means Algorithm](#)
3. [Optimal K](#)
4. [Examining the Cluster](#)

Chapter 32. R Vs Python: What's the difference?

1. [R](#)
2. [Python](#)
3. [Popularity Index](#)
4. [Job Opportunity](#)
5. [Analysis done By R and Python](#)
6. [Difference between R and Python](#)
7. [R or Python Usage](#)

Chapter 1: What is R Programming? Introduction Basics

What is R?

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalogue of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, statistical inference to name few. Most of the R libraries are written in R, but for heavy computational task, C, C++ and Fortran codes are preferred.

R is not only entrusted by academic, but many large companies also use R programming language, including Uber, Google, Airbnb, Facebook and so on.

Data analysis with R rotates around a succession of steps, programming, transforming, discovering, modelizing and communicate the results

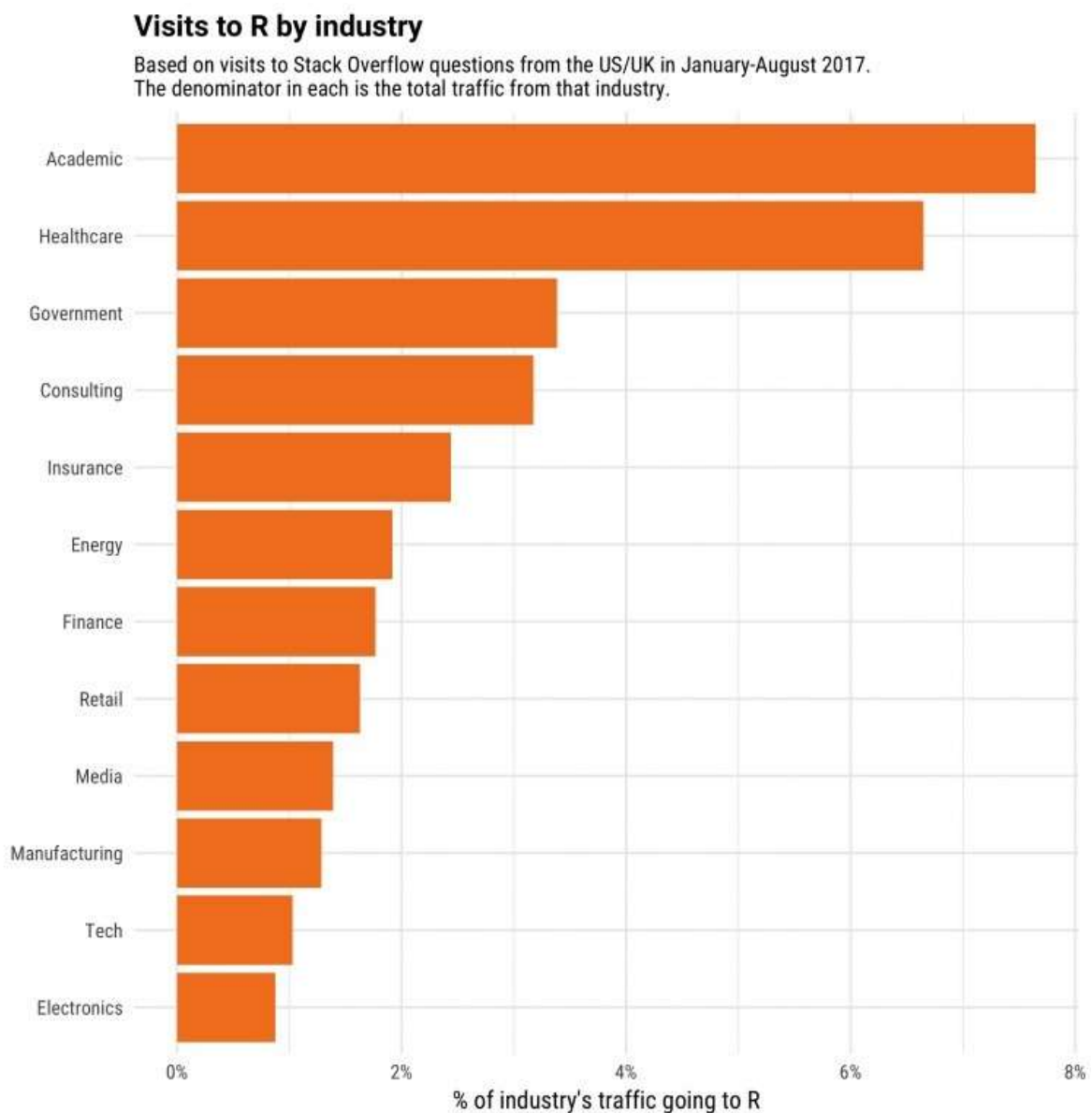
- **Program:** R is a clear and accessible programming tool
- **Transform:** R is made up of a collection of libraries designed purposely for data science
- **Discover:** Empower R environment to dig the data
- **Model:** R provides a wide array of tools to capture the right message from the data
- **Communicate:** Integrate codes, graphs and outputs to a report with R Markdown or build Shiny apps to share with the world

What is R used for?

- Statistical inference
- Data analysis
- Machine learning algorithm

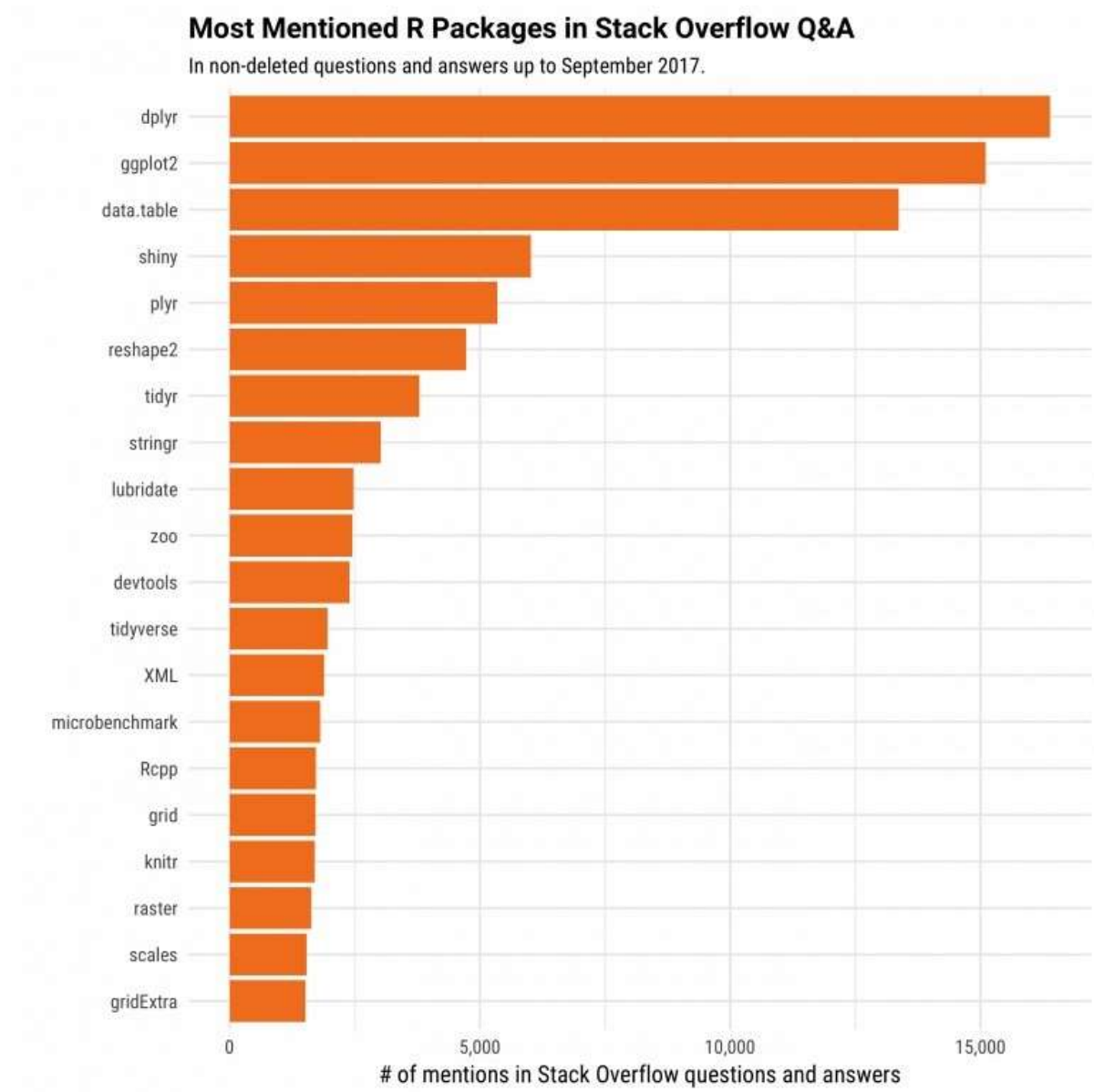
R by industry

If we break down the use of R by industry, we see that academics come first. R is a language to do statistic. R is the first choice in the healthcare industry, followed by government and consulting.

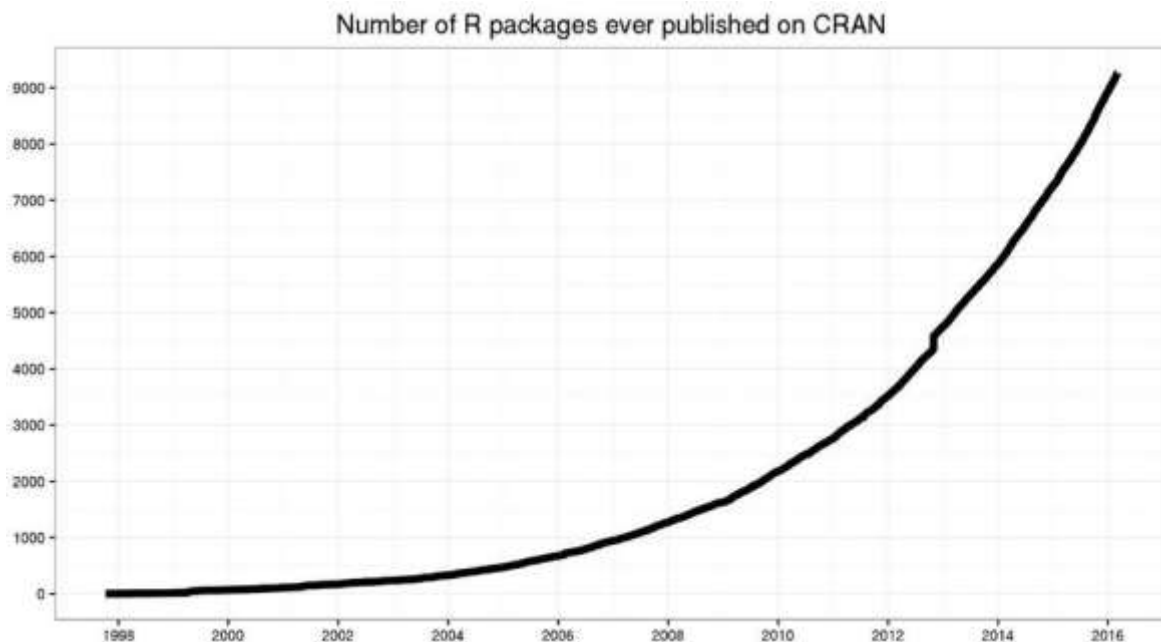


R package

The primary uses of R is and will always be, statistic, visualization, and machine learning. The picture below shows which R package got the most questions in Stack Overflow. In the top 10, most of them are related to the workflow of a data scientist: data preparation and communicate the results.



All the libraries of R, almost 12k, are stored in CRAN. CRAN is a free and open source. You can download and use the numerous libraries to perform Machine Learning or time series analysis.



Communicate with R

R has multiple ways to present and share work, either through a markdown document or a shiny app. Everything can be hosted in Rpub, GitHub or the business's website. Below is an example of a presentation hosted on Rpub

The screenshot shows a web interface for an R Markdown presentation hosted on Rpubs. The title is "R Markdown". The content includes a roadmap of the session:

- Why R
- What Can We Do with R

Annotations with green dashed arrows point to various features:

- "Host your work on Rpubs" points to the Rpubs logo in the top left.
- "Allow users to share your work" points to the "Share" button in the bottom right.
- "Allow users to comment your work" points to the "Comments" button in the bottom right.

The bottom of the interface shows a footer with "Publish Document", "by Thomas", "Last updated 2 days ago", and buttons for "Comments", "Share", and "Hide Toolbar".

Rstudio accepts markdown to write a document. You can export the documents in different formats:

- PDF beamer

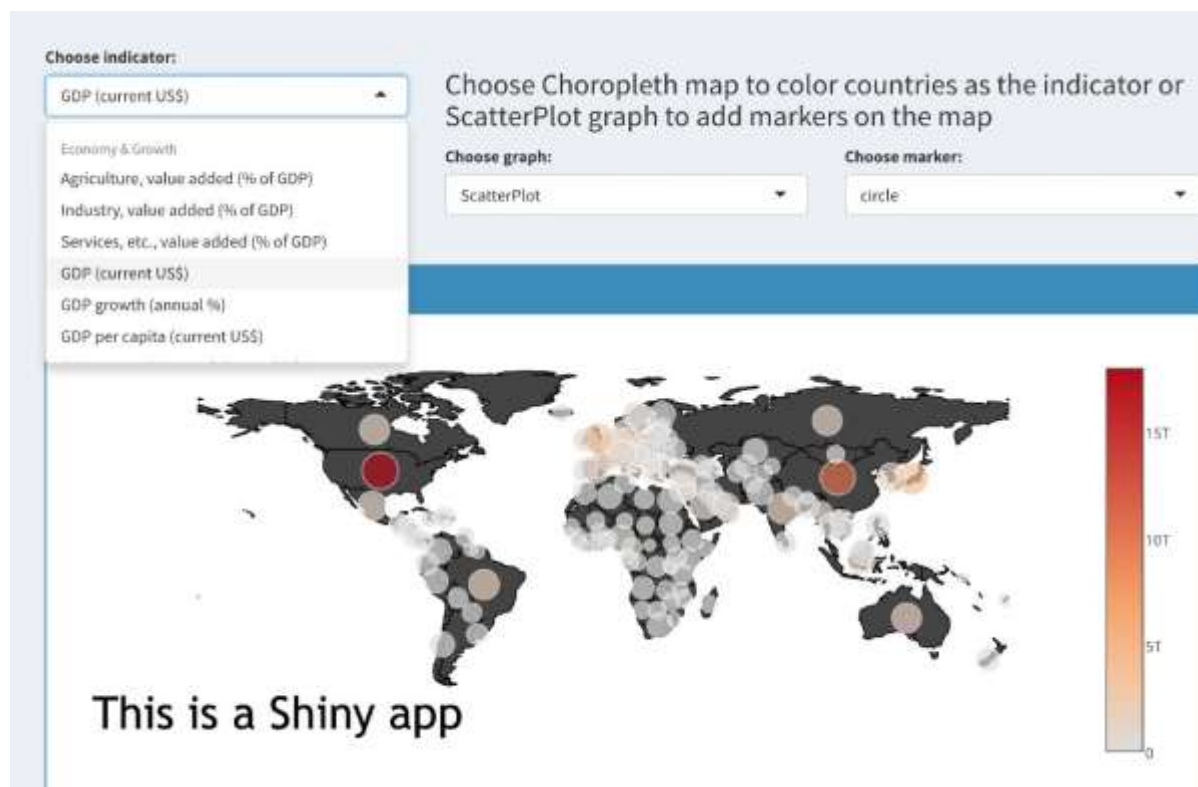
papers, book and presentation

What can we do with R

<http://rmarkdown.rstudio.com>.



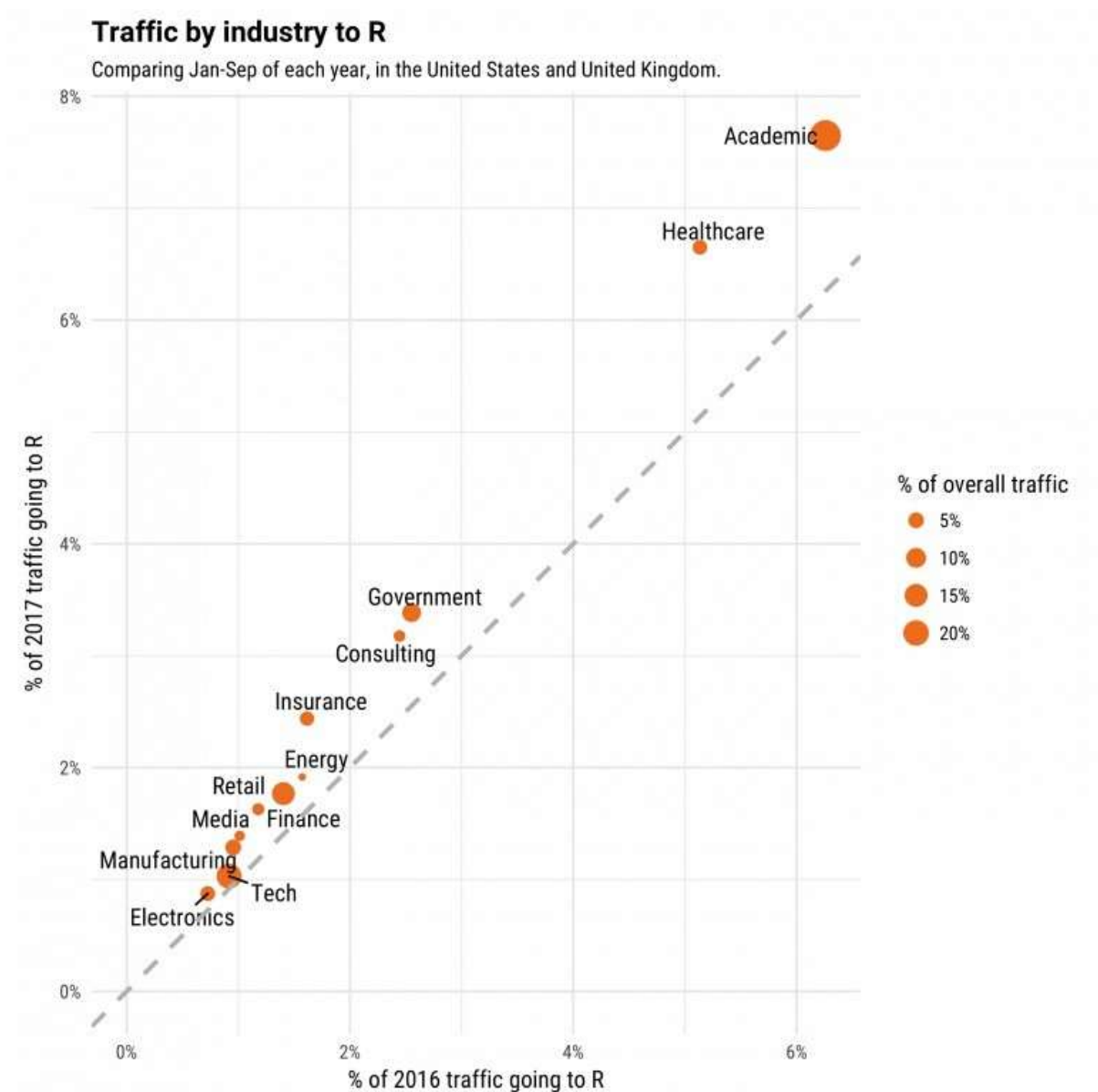
Rstudio has a great tool to create an App easily. Below is an example of app with the World Bank data.



Why use R?

Data science is shaping the way companies run their businesses. Without a doubt, staying away from Artificial Intelligence and Machine will lead the company to fail. The big question is which tool/language should you use?

They are plenty of tools available in the market to perform data analysis. Learning a new language requires some time investment. The picture below depicts the learning curve compared to the business capability a language offers. The negative relationship implies that there is no free lunch. If you want to give the best insight from the data, then you need to spend some time learning the appropriate tool, which is R.

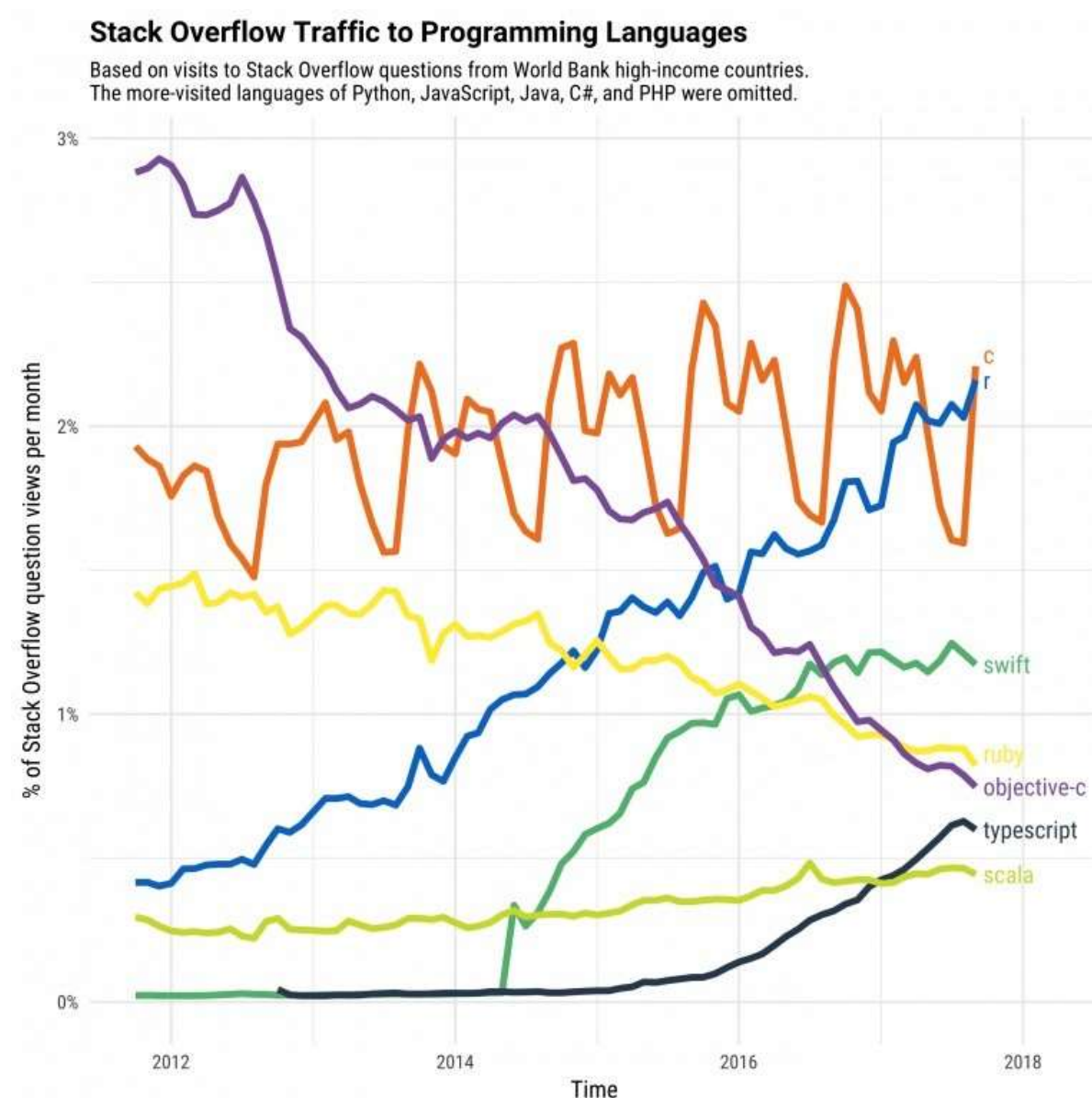


On the top left of the graph, you can see Excel and PowerBI. These two tools are simple to learn but don't offer outstanding business capability, especially in term of modeling. In the middle, you can see Python and SAS. SAS is a dedicated tool to run a statistical analysis for business, but it is not free. SAS is a click and run software. Python, however, is a language with a monotonous learning curve. Python is a fantastic tool to deploy Machine Learning and AI but lacks communication features. With an identical learning curve, R is a good trade-off between implementation and data analysis.

When it comes to data visualization (DataViz), you'd probably heard about Tableau. Tableau is, without a doubt, a great tool to discover

patterns through graphs and charts. Besides, learning Tableau is not time-consuming. One big problem with data visualization is you might end up never finding a pattern or just create plenty of useless charts. Tableau is a good tool for quick visualization of the data or Business Intelligence. When it comes to statistics and decision-making tool, R is more appropriate.

Stack Overflow is a big community for programming languages. If you have a coding issue or need to understand a model, Stack Overflow is here to help. Over the year, the percentage of question-views has increased sharply for R compared to the other languages. This trend is of course highly correlated with the booming age of data science but, it reflects the demand of R language for data science.



n data science, there are two tools competing with each other. R and Python

are probably the programming language that defines data science.

Should you choose R?

Data scientist can use two excellent tools: R and Python. You may not have time to learn them both, especially if you get started to learn data science.

Learning statistical modeling and algorithm is far more important than to learn a programming language. A programming language is a tool to compute and communicate your discovery. The most important task in data science is the way you deal with the data:

import, clean, prep, feature engineering, feature selection. This should be your primary focus. If you are trying to learn R and Python at the same time without a solid background in statistics, its plain stupid. Data scientist are not programmers. Their job is to understand the data, manipulate it and expose the best approach. If you are thinking about which language to learn, let's see which language is the most appropriate for you.

The principal audience for data science is business professional. In the business, one big implication is communication. There are many ways to communicate: report, web app, dashboard. You need a tool that does all this together.

Is R difficult?

Years ago, R was a difficult language to master. The language was confusing and not as structured as the other programming tools. To overcome this major issue, Hadley Wickham developed a collection of packages called tidyverse. The rule of the game changed for the best. Data manipulation become trivial and intuitive. Creating a graph was not so difficult anymore.

The best algorithms for machine learning can be implemented with R. Packages like Keras and TensorFlow allow to create high-end machine learning technique. R also has a package to perform Xgboost, one the best algorithm for Kaggle competition.

R can communicate with the other language. It is possible to call Python, Java, C++ in R. The world of big data is also accessible to R. You can connect R with different databases like Spark or Hadoop.

Finally, R has evolved and allowed parallelizing operation to speed up the computation. In fact, R was criticized for using only one CPU at a time. The parallel package lets you to perform tasks in different cores of the machine.

Summary

In a nutshell, R is a great tool to explore and investigate the data. Elaborate analysis like clustering, correlation, and data reduction are done with R. This is the most crucial part, without a good feature engineering and model, the deployment of the machine learning will not give meaningful results.

Buy Now \$9.99