PoweR of Programming

Product development & Innovation team, Nielsen, Bangalore

2017-08-17

Introduction to R

What?

R is a programming language and an environment for interactive data analysis. The first version of R[1] was released in 1993, growing out of the S[2] language developed at Bell Labs in the late 70's.

The main strength of [R Core Team, 2017] is its rich ecosystem, which includes commercial vendors, thousands of add-on packages, mechanisms for getting help and many venues for scientific publishing.

[1] To know more about R: 20 years of CRAN
[2] History of Advanced Statistical Computing with S: Forty years of S

Why?

Simply because:

- R[3] 3.4 comes with a JIT and Bit wise compilation.
- Easy to integrate with Python, Scala, C, C++
- Support for SQL database
- Scaleable and Open Source
- 10000+ packages in CRAN and community support
- Great visualization support and JavaScript visual library integration
- Reproducible research & version control from IDE
- Mathematical notation, HTML, PDF, presentation, authoring books and journals using IDE

[3]

Still confused? Check out this playlist to know more:
R Programming for Excel Users

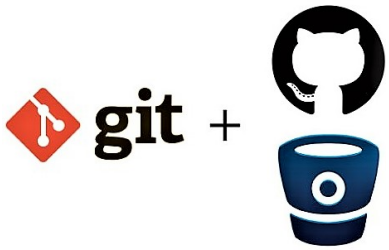
How?

It is impossible to cover R in a single session. So the objective of this knowledge sharing session is to share important resources and showcase some of the useful examples so that we can leverage the open source community support of R according to our need.

- In Nielsen we have collaboration with a company called Xurmo[4]. They are building a data manipulation and cleaning product for us. We can use an R script to customize that data cleaning and transformation process.
- R scripts are reusable. It can save a lot of time and resource when it comes to data cleaning and visualization.

[4] Click here to visit Xurmo

- We can leverage other people's code to solve out problem. There are many open source community support such as stack overflow, cross validate and GitHub[5]. From my personal experience I can say people are really helpful out there and one can seek other people's help and learn things on their own.
- We can read and write large scale data with R which is not possible in excel.



[5]

Up and running:

It is quite easy to install R in general and here is a video tutorial which demonstrates how to install R in a windows computer. BUT if you are using office laptop then contact IT-man.

Motivation:

Visualization:

R has one of most beautiful visualization available for data analysis. It is based on a rich theoretical foundation. There are many packages available in R for plotting, but base R is the fastest among all. When it comes to elegant and beautiful looking graphics, other packages such as ggplot2, performs better than the base graphic engine of the R.

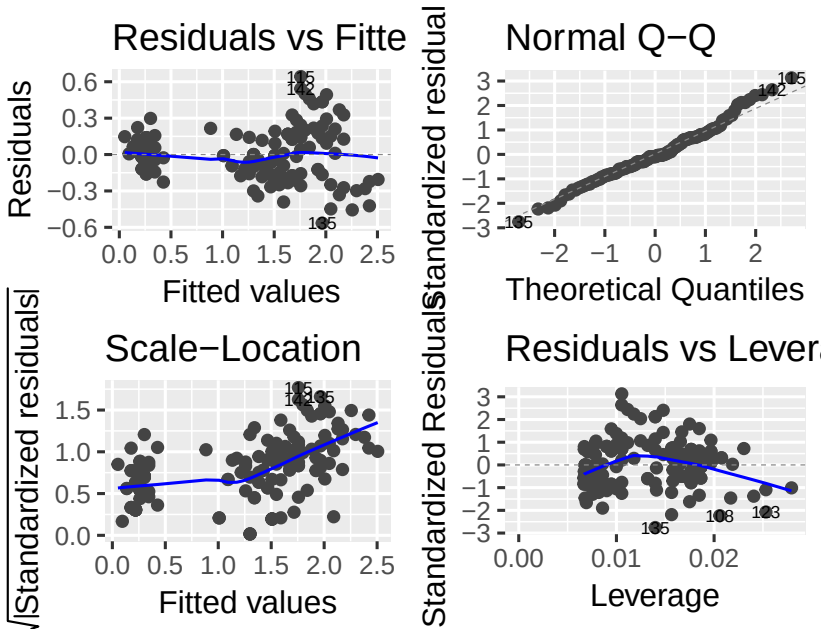


Figure 1: Diagnostic Plots for Linear Regression Analysis

This is a residual diagnostic plot generated using package `ggfortily`.

It takes only one line of code[6] to generate this beautiful diagnostic plot. This package is developed on top of the most popular graphics library `ggplot2`.

When it comes to visualization, or is the best available data visualization, open source programming language available at present. Here are some of the useful visualization libraries in R for interactive visualization[7]:

- htmlwidgets : Implementation of JavaScript based plotting libraries.
- googlevis : Google's data visualization package for R.
- ggplot2: It is a package based on the grammar of graphics.
- ggvis: It is a data visualization package for R.
- Plotly: Graphing library makes interactive, publication-quality graphs online

Here are some useful example[8] from these useful plotting libraries:

- htmlwidgets
- googlevis
- ggplot2
- ggvis
- Plotly

Some of these projects do not have support currently but I will recommend to go through them. Some of them have some truly awesome visualization examples[9].

Modeling:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.3631	0.0398	-9.13	0.0000
Petal.Length	0.4158	0.0096	43.39	0.0000

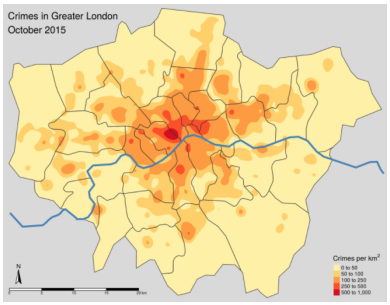
Here is the model result generated using the `summary()` function in R. To check the model result one can use the the following code[10].

There are multiple packages avaiable for model buildling in R but `caret` is my personal favourite.

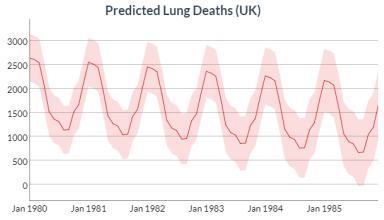
Reproducibility and Version Control[11]*:*

> "If you're serious about software development, you need to learn about Git. Git is a version control system, a tool that tracks changes to your code and shares those changes with others. Git is most useful when combined with GitHub, a website that allows you to share your code with the world, solicit improvements via pull requests and track issues."

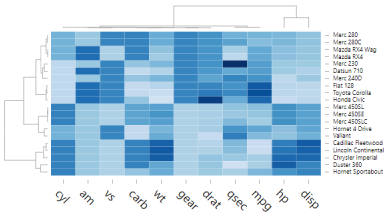
[6] `ggfortify::autoplot(lm(Petal.Width ~ Petal.Length, data = iris), label.size = 3)`



7



8



9

[10] `summary(lm(Petal.Width ~ Petal.Length, data = iris))`

11

— Hadley Wickham

One of the major aspects related to data analysis is to create the same result multiple times with same data. This becomes a major concern because many people collaborate on a same project in different time. Also reproducibility is important in terms of scientific experiment. Reproducible is one of the major aspects of any research project. When it comes to R or any other programming language, many people depend on other people's work for their own work. In order make their program robust so that their code does not change along with the change in authors' code this is important to make it reproducible and here comes 'Packrat'. Packrat is mainly useful for package authors. R packages are nothing but a set of function written with standardized documentation. It is east to create packages in R. For details go through the following materials:

- How to create packages in R
- Setting seed in R
- Monte Carlo Simulation in R
- Meet Packrat

We all can be agreed in a point that it is absolutely important to collaborate in a project for better outcome. Git or any other version control is the best way to track your code and work with the other people. Here are some more reasons why one should use Version Control System:

- Collaboration[12]
- Storing Versions (Properly)
- Restoring Previous Versions
- Understanding What Happened
- Backup

12

Here are some useful resources which can be helpful for installing various version control system such as git, GitHub, bit bucket and using them from Rstudio :

- Install git
- RStudio & GitHub Integration
- Collaborate with bit bucket Rstudio and R

Data manupulation[13]*:*

When it comes to data manipulation, cleaning and transformation, R provides an easy to use interface called dplyr. It has a specific grammar which makes it easy to use. It has a lot of similarity with SQL and Python.

13

This makes data manipulation in r quite easy and intuitive. This makes the language easy to pick up for those who are coming from different background. This is an attempt to standardize the data manipulation process standardized across the landscape. One huge advantage of using dplyr is the it provides a common interface for big data infrastructure in R. It is better to pick up this data manipulation tool because if someone is comfortable with the syntax of this package, they can leverage this for big-data manipulation with some little change during the setup process.

Here are some useful examples and use case using dplyr:

- Data Wrangling & Feature Engineering with dplyr
- R Package :: {dplyr}

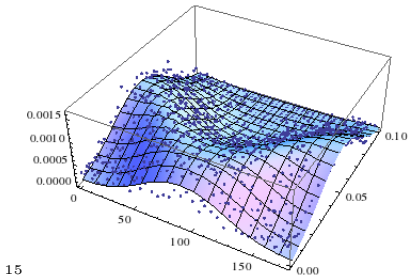
Model building:

There are multiple packages available in R but `caret`[14] is my personal favorite. This package is written by Max Kuhn. This library has been developed through past 10 years. It is a collecting of almost all model building methodologies available in R. The theory associated with the package can be found in the book named Applied Predictive Modeling and in this website. Along with extensive methods it comes with some of the best pre-processing and advanced setting with common syntax. Here is a list of some important libraries for model building in R:

14

- caret
- elasticnet
- rpart
- e1071
- RandomForest

Here are some videos which can be useful in this context[15]:



15

- Introduction to Machine Learning with R and caret
- Caret package webinar - Max Kuhn
- Applied Predictive Modeling - Max Kuhn
- R: Applied Predictive Modeling

Webscrapping:

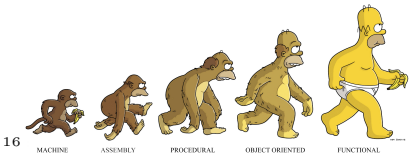
For a data scientist accruing and cleaning the data is one of the most important part of the journey. One of the main resources for data today is the web. It may be through API or it can be simply from an html page. Here in R we have an awesome package called “rvest”. This one is quite similar to the beautiful soup in Python. It helps us to fetch large scale data from web and convert them into a structured

form such as dataframe. Here in this session we will go through some of the examples of similar problem.

- Download data from websites in easy and structured manner

Functional programmming:

Functional programming[16] is branch of programming which can help us to pass functions as argument to a function. R has some pretty awesome packages which can deal with multiple functional arguments, unstructured data and many more. Such packages are useful to deal with more than one model in R. purrr, broom are the most useful packages in this direction.



16

- Hadley Wickham: Managing many models with R
- Functional Programming In R

Simulation:

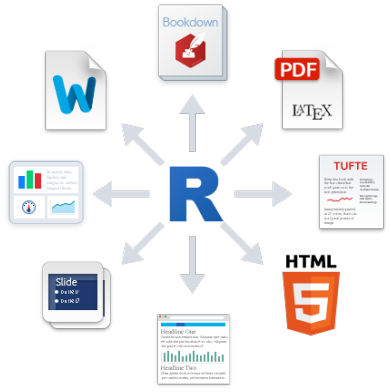
Simulation is one of the most interesting concepts of statistical programming. It became popular during the Second World War. Those who are interested can go through the article called "The Manhattan Project, the first electronic computer and the Monte Carlo method"[17]. In simple terms replicating a real life experiment in a virtual setup is simulation. There are numerous example and applications of these methods in different field. In this session we will go through some of the basic examples of simulation in R.


17

Easy to share:

This is perhaps the most important part of an experiment. An experiment is of no use if it cannot be easily shared. R makes analysis easy to share in different platform. Rmarkdown[18] and shiny are packages developed by Rstudio. It helps to share reports generated in R and also helps to create easy and beautiful web applications. With a single click one can generate webpage, presentation, word file, web applications using [Allaire et al., 2017]. There are multiple formats approved by the journals as template in R which makes life easier even in academia.

Here are some resources which can be useful in this context:



18

- Publish in Rpubs from R
- Using Rmarkdown
- In depth tutorial using R markdown
- RMarkdown & RNotebooks for Research

Imputation:

Imputation is the most important concept when it comes to real life data. Real life data are much more different than what we see in classroom set-up. Most of the real life data come with missing values. One of the easiest ways to deal with this form of problem is imputation. There are a number of packages available for solving this type of problems in R but here are some of the most useful packages I have come across.

- simputation: Simple Imputation
- MICE

These two provides the most unique strecture to run imputation using different methods. Due to the commonality in the structure this is easy to remember the syntax.

Tidy Data:

One of the core philosophies of the R community is structuring unstructured data in a manner so that it is easy to deal with big-datasets. Tidy data is a dataset which says that related things should stay to gather. Based on this philosophy many packages have been developed in R. In this session we will go through some of the examples of tidy datasets and an example related to this concept has been demonstrated in Hadley Wickham: Managing many models with R.

Optimization and efficient programming:

> "The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming."
>
> — Donald Knuth

It is important to write efficient code. But more than writing efficient code, it is important to write efficient code when it is required. R has many options to understand efficiency of a code and optimize them. Here are some usful links related to code optimization in R.

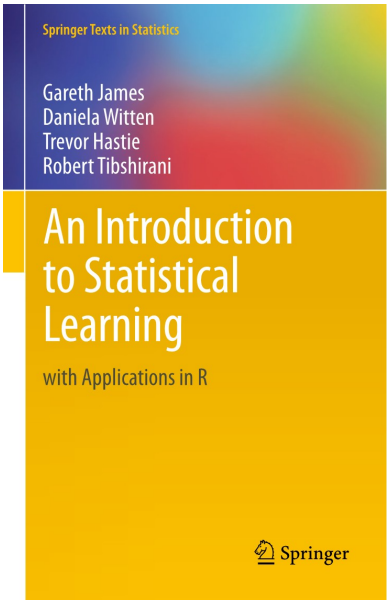
- Efficient R Programming
- Efficient R Programming II
- Writing Efficient R Code

Resources:

Although there are number of available resources[19] available online for

learning R but here I'm shortlisting some of my favorite courses and online resources which I found very useful. Most of these are free and helpful but a few of these courses are paid but according to me these courses will definitely add some value to your skillset.

- Introduction to Statistical Learning[20]
- Elements of statistical learing
- Data science for R
- Caret tutorial
- ISLR video couse
- Datacamp
- UseR!
- Rstudio::conf 2017
- Rstudio Webinars
- Rstudio Cheatsheets



[20]

I shall be more than happy anybody is interest to collaborate in a learning initiative.

References

JJ Allaire, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, Rob Hyndman, and Ruben Arslan. *rmarkdown: Dynamic Documents for R*, 2017. URL `http://rmarkdown.rstudio.com`. R package version 1.6.0.9001.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.