Introduction to R



**Anish Mahapatra**

1. Introduction

Hello everyone!

Welcome to the course Introduction to Data Science with R. The very fact that you registered for this course proves that you’re headed a step in the right direction to start your journey and to enhance your position in the Data science domain.

But before we begin let me tell you what you can expect from this course.

I will primarily be speaking about data science, some of the applications, not all of them since it’s a field with a vast scope across different domains. I will also be introducing you to the statistical coding language R which acts as a great tool to actively compute complex statistical analysis in a few simple steps.

The objective that I would like to achieve is to make sure you leave this course a little better versed in R and it’s uses without delving too deep into the technical side of things.

We will be dealing with questions such as What is data? What is R? What are packages? And a whole lot more that will give you a basic understanding of the beautiful field and how to go about it in the right way.

My name is Anish Mahapatra. I’m a Senior Data Scientist with a master’s degree in Data science with more than 5 years of work experience with a fortune 500 company and I also am a consultant that’s worked with multiple clients in the data science domain in India as well as the world.

Without furthor ado, let us now dive into the fundamentals of Data Science and R!

1. What is Data?

It’s a simple question, right? What is Data?

1. What is R?

R is an object-orientated programming language. This means that you create objects, and give them names.

You can then do things to those objects like:

Perform operations and calculations in a simple way.

Statistical Analysis and Tests

Make tables

Draw plots and visualise data for better understanding.

Objects can be single numbers, characters, vectors of numbers, matrices, multi-dimensional arrays, lists containing different objects and so on. We will use a fantastic IDE1 provided by RStudio. This is free to download, provides some neat features, and crucially, looks the same on all operating systems!

1. The applications of R - (Talk about RStudio)
2. What is an R Script?

Importance of scripts I

They keep a systematic record of your analysis, which enables you to reproduce your work at a later date, or can be passed to collaborators or other users to enable them to replicate your work.

This record means that you do not have to rely on your memory to figure out what you did. • R scripts allow you to comment your code, which means that you also won’t forget why you did it. Importance of scripts II • In more advanced settings, R scripts can also be run in batch mode, which means that you can ping a script off to run remotely on a server remotely • Although programs like SPSS allow outputs to be saved, R scripts contain inputs, which are much more useful, since it is easier to generate the outputs from the inputs than it is to reconstruct the likely inputs from the outputs. • In fact, R scripts can be combined with a markup language called ‘markdown’ to generate fully reproducible documents, containing both inputs and outputs. It does this using the fantastic knitr and rmarkdown packages. (In fact this workshop was written using rmarkdown and a package called bookdown.)

Comments

RStudio comes with its own text editor, but if you are not using RStudio, others are available. R is case-sensitive. If something doesn’t work, it’s often because you have failed to capitalise, or capitalised where you shouldn’t have. NEVER, EVER, EVER use Word to edit your R scripts! Word often tries to correct your grammar and is an absolute nightmare to work with when writing code. NEVER use it for writing code.

1. What are packages in R
2. Commonly used packages in R
3. R for statistical analysis
4. Excel versus R versus Python
5. Data Types in R
6. Coding with R – Installation of R / R Studio
7. How to read a dataset in R

As discussed before R is a coding language that primarily deals with Statistical Analysis and Modeling, but in-order for us to do this we first need to have some data, right?

This data can be be of various types like we discussed in the previous section and can be wrangled, cleaned and formatted to our liking based on our requirements and objectives.

Let’s not get ahead of ourselves, let’s first load the data into our IDE which in this case is going to be Rstudio.

This can be done in a few ways but let’s discuss the two most popular and easy ways to go about it.

The first method I would like to discuss is simple, effective and consists of one line of code. It is the (read.csv) function, It’s a relatively simple function that entails the dataframe you would like to assign the data to, the function itself and the name of the file which needs to be suffixed (dot csv)

So if I have a dataset named weather that I would like to read into Rstudio I would like to first assign a Dataframe into which I am reading my data, for example we will call it df but you can name it whatever you like although I recommend you name it something relevant to the data you’re going to read to avoid confusion in the future, anyway first I assign the name of the dataframe, in this case df then I use the assign symbols <- followed by my read.csv function then open paranthesis and quotation marks followed by the name of the data file I am trying to read which in this case is weather and I add the dot csv and close the quotations and paranthesis. And voila you’re data has been sucessfully been read into Rstudio. There are a few variations to this function such as read.csv2 and read\_csv which perform a similar function but with tiny differences although Read.csv is the most commonly used. An important point to remember is to make sure you set the working directory as the same location of the file or folder of data you would like to read as this allows Rstudio to know where to find it. This can be done by simply by typing in the setwd() function.

Another popular way to read data onto Rstudio is to manually choose the import dataset feature found in the file menu. Select the “From text (Readr)” option and browse through the files you would like to import onto the dataset.

1. Data Operations in R

Now that we have sucessfully been able to read our data into Rstudio let us check out some of the basic operations we can do in R. An operation is a computation that is being executed and an operator is the type of computation to be executed. For example when we do some basic addtion of 2+2 the entire process of doing this is an operation and the + symbol is an operator. Lets look at some of the basic types of Operations, in this section we are primarily gonna be discussion Arithmetic, Relational and Logical operators.

Arithmetic

Some basic mathematics with no requirements to for Syntax of R can be simply computed in Rstudio much like that of a calculator.

(Anish can you show some basic operations here in Rstudio doing 2+2 and all that stuff, please use the screenshot below to add all the relvant arithmetic operators)



We can also see the operator associativity in R Associativity means which of the operations will take precedence. Lets say you have both division and multiplication then which operation should take precdence.

Example of (6+3\*4) = 18 because multiplication takes precedence over addition and so on.

Relational Operators

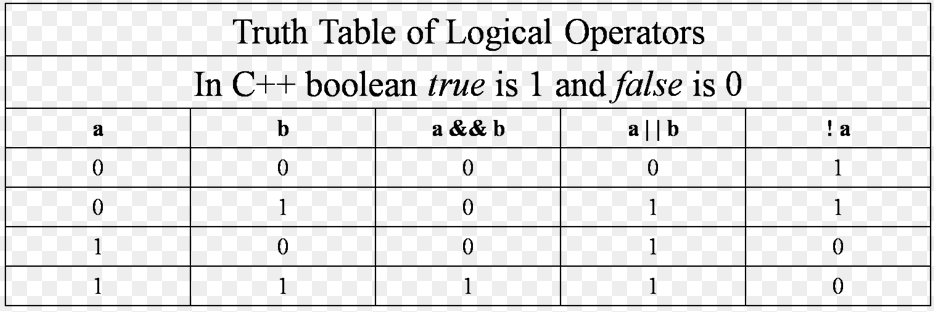
Relational operators are frequently used for conditional filtering of Data in R. We can assign a value to a variable using “<-”. A variable is named unit of data that can be assigned a value. The other main use of Relational operators such as <>=! is to check if the value is greater than, lesser than and/or equal to each other.

So if the statement we put forth is 8>5 then the value that R will show us is TRUE, similarly if we out forth 8<6 then we will get an output of FALSE

Logical operators

A logical operation will return an output of True or False and is frequently used for conditional filtering and other operations in R.

This is based on the AND, OR and NOT operator



(Use the table above to explain Anish)

1. Data Wrangling in R

What is data wrangling? Usually in most cases it isn’t possible to acquire data in a perfect format with absolutely no missing values, or some random variables that just don’t make any sense. The ones that are in the wrong format, the outliers and so on. So we clean the data in different ways based on what we’re looking to do, for example in a dataset with over 10 coloumns we are looking for values in 1 of them we can simply elimainate the rest and continue our process as they are irrelevant but in case we do need them then we must look at the data, clean and format it before we conduct operations on it. R has many functions which are usually part of packages that help us seamlessly explore data and help wrangle it.

We will not delve deep into this but some of the cool things that can be accomplished with the help of R is firstly exploring the data and having a sense of it, we can check the number of missing values in each coloumn, the highest, lowest, mean, median values all with the help of the summary function. We use the head function to see the first 5 values of each of the coloums which is customisable bu adding the number of rows you wish to see in the paranthesis of the head function.

The tidyverse package is commonly used in most projects to help with this aspect of the projects.

Some of the cool things we can do is merge two datasets, aggregate coloumns, impute missing data and whole lot more.

Imputing data is done when there’s missing data which we do not wish to ignore but instead we would like to fill it as accurately as possible, this can be done using the average of the previous and next filled row, or through a variety of different techniques. There are specific packages in R designed for this purpose and only this purpose such as MICE.

We can also reshape data which means change the layout of a dataset, transpose the rows and coloumsn and basically make it suitable to us in any way shape or form. We cn use subsets to perform operations to experiment, We can make new variables by mutating which is compute and append one or more new coloums.

Combining datasets using join function (left, right, inner, full)

Group data using groupby function

The scope for this is truly endless, R acts as tool to not only make it easier to do our wrangling but it also makes it far more efficient.

1. Visualization with R

Visualization of data is arguably one of the most important aspects of data science, this is mainly because most of the data science projects if not all of them will have a practical implication that has an effect on the world, be it business, personal, climate related and so on. Thus step of the process bridges the gap between statistics and business implementation because when you present your results to stakeholders it would be more practical to make them understand with the help of graphs and tables rather than numbers and mumbo jumbo.

Data visualisation is the presentation of data in a graphical format.

It can provide a valuable insight into your data and help in identifying patterns

Numerous methods are available to visualise your data

bar charts

pie charts

scatterplots

histograms

box plots

line plots

maps

and many more!

(Anish see the file ‘Working with Data’ page 6, this has elaboratley detailed out different visualizations and I think it will be a good idea to display it and talk about correlations, change over time etc)

R performs various types of visualizations with the help of many packages such as ggplot2, seaborn and some other packages with specific types of visualisations such as maps. A commonly used one for many different types of visualisations is ggplot2

The ggplot2 package is a powerful graphics package in R. You build a ggplot up piece by piece, combining the pieces with the “+” operator. Graphics using ggplot2 can be tailored to your analysis. You can create histograms for example, but the beauty is you will be able to fully customize this by addig labels to the axes as well as changing the colour, font and also a legend to give you a descriptive graph with all the information you need.

1. Applications of Machine Learning in R

Machine learrning has applications in all domains ( I’m not sure if you’re doing this) it won’t take me long to do if you aren’t let me know cause 15 was repeated twice, anyway it’s like 2:30 am so I’ll crash and worst case scenario I’ll do it tomorrow if you havn’t done it

1. AI in R
2. Thank you!

# Author

**Anish Mahapatra**   
*Senior Data Scientist*   
LinkedIn: [https://www.linkedin.com/in/anishmahapatra](about:blank) ([LinkedIn](about:blank))   
Email: anishmahapatra01@gmail.com