**Introduction to R**



Anish Mahapatra

Table of Contents

[1. Introduction 3](#_Toc120382011)

[2. What is Data? 4](#_Toc120382012)

[3. What is R? 4](#_Toc120382013)

[4. The applications of R - (Talk about RStudio) 5](#_Toc120382014)

[5. What is an R Script? 7](#_Toc120382015)

[6. What are packages in R 8](#_Toc120382016)

[7. Commonly used packages in R 8](#_Toc120382017)

[8. R for statistical analysis 8](#_Toc120382018)

[9. Excel versus R versus Python 8](#_Toc120382019)

[10. Data Types in R 8](#_Toc120382020)

[11. Coding with R – Installation of R / R Studio 8](#_Toc120382021)

[12. How to read a dataset in R 8](#_Toc120382022)

[13. Data Operations in R 9](#_Toc120382023)

[14. Data Wrangling in R 10](#_Toc120382024)

[15. Data Wrangling in R 10](#_Toc120382025)

[16. Visualization with R 10](#_Toc120382026)

[17. Applications of Machine Learning in R 10](#_Toc120382027)

[18. AI in R 10](#_Toc120382028)

[19. Thank you! 10](#_Toc120382029)

[20. Author 12](#_Toc120382030)

[21. Archive 12](#_Toc120382031)

# Introduction

Hello everyone!

Welcome to the course on the 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.

We 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. We will try to keep this course light on code and more about understanding the nuances of R, it is an introductory course focused on learning and getting rid of your fear of programming.

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 MSc degree in Data science with over than 5 years of work experience. I currently work at a Fortune 200 Retail Giant and I also am a consultant that’s worked with multiple Fortune 500 clients in the data science domain. A great way to know more about me would be to simple Google my name – Anish Mahapatra.

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

# What is Data?

It’s a simple question, right? What is Data? Data is a collection of discrete values that convey information. This can be in the form of quantity, quality, fact, statistics, and other basic units of meaning.

The world is driven by data – all the way from your mobile phones, e-commerce, banking and even your cars!

When looking at it from an uber perspective, it is important to understand what are companies spending their money on. The reason Data Science has such high holding for the next couple of decades is that companies are in the process of leveraging their data.

Data can be seen as rows and columns, where we are able to get contextualized information in a concise manner. It can be somethingas simple as your shopping list.

Now, how does this work in the digital world? We can have data in the form of csv files, excel files, a database and on the cloud platform. All of this data flows from different sources and there is great value in understanding what all of it means.

It might be easy to understand data that comprises of 10-20 rows. But, what if we increase that number to a 100 million rows? It becomes difficult to “understand” the data then, because it does not even open in Excel.

This is where a language like R can help. It can help you understand, infer and visualize large volumes of data.

Let’s have a look at What is R in our next session. This is probably a great place to tell you to not fear code. You have gotten this far and it is an achievement, and we will make this as easy to understand as possible. Stick with me.

# What is R?

R is a great language for everyone to learn as it’s sole purpose is to collect, analyse, interpret and present data. In fact, this is the very definition of statistical analysis.

R is a programming language and environment for statistical computing and graphics to analyze and visualize data.

Every language has associated rules, called syntax that govern it. There are three items that are a part of the syntax of R are:

1. **Variables**: Objects that can store data
2. **Comments**: To make the code easier to understand and more readable
3. **Keywords**: Words that are reserved for the compiler of the language. What is a compiler? It converts instructions into machine-level language.

So, why do people actually use R?

As one of the top five programming languages currently. R is used by large companies in disciplines such as Fintech, Retail, Social Media, Healthcare etc. to do the following:

* Data Analytics
* Statstical Inference
* Machine Learning

There is a good reason why a language becomes one of the most widely used in the world. Some of the reasons for R programming to be one of the op languages are:

* **Open-source free language**: R and it’s suite of tools is completely free to download and use
* **Platform Independent**: Whether you have a mac, windows, linux or even a cloud server, R can be run on all platforms with ease
* **Leading tool for machine learning**: R programming is one of the leading tools for machine learning, statstics and data analysis. It is one of the most requested language in the Data Science job market.
* **Flexibility**: R can be integrated with many other languages, and this makes it even more useful, irrespective of the language that is used
* **Community Support**: Any questions that you have with R can easily be answered with a Google search. This is because there are a lot of people that support the community.

Alright Anish, you seem to have sold me on the potential of R, not just in the job market, but also in the search for great Data Science jobs. So, how do we actually use it?

I’m glad you asked. In the next session, we will talk more about some of the applications of R, before we deep-dive and look at how you can get started in under ten minutes!

# The applications of R - (Talk about RStudio)

Hi everyone, welcome back to this session, where we will discuss the applications of R. At any point if the complexity of data increases, we need ways that we can interpret, interface and analyze the data. There are use-cases in multiple domains such as:

* **Research & Academics**: Since R Programming is used primarily for statistical computing, it is a great resource for students & teachers to leverage
* **Information Technology (IT) Sector**: The IT sector uses R to enable businesses to build statistical, computational tools, data handling and business intelligence.   
    
  There are use-cases such as the analysis of web activity, recommendation systems, marketing campaign analysis and even matchmaking systems
* **Banking & Finance**: Risk management of assets, providing loans to potential customers, upselling financial & banking products are some use-cases where R can be tremendously helpful.   
     
  It can also be used for fraud detection, loan stress modelling, loan stress test simulation and a lot of other forms of risk analytics.
* **E-Commerce**: E-commerce uses the potential of R to work on use-cases such as marketing mix modelling, cross-selling, targeted advertising, sales and financial data modelling.
* **Healthcare**: R is greatly used in healthcare for a plethora of operations. It can be used to do analysis in the fiels of Genertics, bioinformatics, epidemiology, and, drug discovery. It can also be used to understand the statstical understanding of the success of clinical trials.

Alright, great, it’s good to know that it is used in a lot of verticals across multiple industries. Now, the question is are we getting closer to understanding how can I get started with R? Yes, we are getting close, but we are not just there yet.

An IDE or integrated development environment is a software to build code, which combines developer tools into a single interface. Luckily for us, the IDE for R, called RStudio is a all-in-one solution to quickly get started with R!

In the next section, we will learn more about the core concept of how we will set out to build fantastic things with R – using R scripts!

# What is an R Script?

Welcome to this session, where we will understand more about R Scripts! Put Simply, R Scripts are just text files that contain commands to run in R. They form the playground with which we can let R work it’s magic.

Let’s discuss the importance of scripts in R, here are some of the reasons why scripts are important in R:

* **Helps reproduce your work**: R scripts enable you to keep a systematic record of your analysis, so that you can reproduce your work later or collaborate with other users
* **Do not need to remember what you did**: Keeping a record of your R code on R Scripts means that you do not need to rely on your memory to figure out what you did
* **Allows you to add comments**: R Scripts enables you to add comments to your code, sothat you not only remember what you did, but also **why** you did it
* **Can be run in batch mode**: R scripts can be run in batches, which means a series of scripts can be run parallelly or in a series
* **Can add inputs in R scripts**: R programming allows you to give inputs to your program, so that it will be easier to generate the outputs from the inputs so that it is easier to reconstruct your code while showcasing it
* **Can be combined with markdown**: One of the biggest advantages of R scripts is that it lets you combine R with a language called ‘markdown’, which let’s you generate fully reproducible documents that have both the inputs and the outputs.   
    
  This is done using the knitr and rmarkdown packages!

“Packages” New Terminology Alert! We will learn more about packages in our upcoming session on “What are packages in R?”, and we will also get to see examples of packages and their super powers.

Meanwhile, I’d like to quickly touch upon RStudio. RStudio has it’s own text editor, which you can use to write R Scripts. Later in this course, we will teach you how to download RStudio and write your first R Script with it! See you in the next session!

# What are packages in R

Welcome to this session, where we we will learn more about packages in R. Let’s say we would like to calculate the sum of a set of numbers in R. We can either write the R code from scratch or we can use a function.

For a lot of simple or complex tasks, the community has already written the code. Wouldn’t it be great if we could all leverage code that has already been written?

Great news! This already exists. Packages in R are a collection of R functions, data and compiled data in a well-defined format, that is created to add specific functionality.

One of the reasons that R is one of the top 5 programming langauges in the world is because it has over 10,000+ user contributed packages and this number is growing constantly! This is a great test of how supportive the open-source community is.

This enables anyone working in R to simply install a package from the package repository and use it per their use-case. CRAN, or Comprehensive R Archive Network contains an archive of the latest and previous version of R, the documentation and all the versions of the R packages.

Now that we have a clear understanding on packages in R, in the next session we will understand more about the commonly used packages in R.

# Commonly used packages in R

Welcome back to this session, where we will learn about some of the popular packages in R. Some of the more popular packages that are used in R are:

* **dplyr**: The dplyr package is used to perform data analysis and data transformations. Where there are several functions to work with dataframes
* **ggplot2**: ggplot is the mostpopular package to create graphics. It enables the user to create elegant plots and graphs from the data
* **tidyr**: tidyr is a package that is used to tidy the data. In R, a data is considered tidy when each row represents an observation, and each variable represents a column
* **shiny**: R goes beyond simple visualizations. R Shiny is an interactive web application that lets the user to embed visualization in a web platform and also has in-built support for animation, graphs, plots and charts
* **caret**: Caret stands for Classification and Regression training. The Caret package helps model complex regression and classification problems. CaretEnsemble can also be used to combine (ensemble) different models

Some of the more popular packages you can look up are tidyverse, plotly, knitr and mlr3. Now that we know what packages in R are let’s move on to on of R’s primary purpose – Statistical Analysis. Let’s learn more in the next session.

# R for statistical analysis

There is imeense scope in R for statistical analysis. In this session, we will briefly cover some ways that we can use R for statistical analysis. Of course, to understand the topic in depth, we would need more time.

We will touch upon the main pointers through which we can use R for statistical analysis:

* **Descriptive analysis using R**
  + Descriptive analysis is the process of using current and historical data to identify trends and relationships. It is the simplest form of analysis.
  + We can understand how to measure central tendency of the data through Mean, Median and Mode with R
  + The variation in the data can also be understood through Range, IQR, which stands for Inter-Quartile Range and Deviation.   
      
    It’s fine if you are not completely aware of the terminology. This is 10th Grade mathematics and a simple revision through the course will help you understand the possibilities of R can help understand the underlying trends in the data.
* **Data visualization with R** 
  + R enables visualization by giving the user the ability to plot the data in terms of bar charts, scatter plots, pie charts, histograms and along with many other visuals
  + We can aslo take subsets of the data, and modify detaisl in the plot such as the axis name, the colors, the markings and denotions, text size and color and even plot multiple charts in a single space
  + R also enables us to plot time series plots, where we are able to see trends over a period of time. We will have a chance to go through some of the visualizations in R in a later section.
* **Probability with R**: We can deep-dive in probability with the help of R Programming as well, where concepts such as union and interesection, factorial, and permutation and combination are covered.
* **Probability Distributions** 
  + This is a large segment to cover. Essentially, R is one of the top programming langauges used to cover the probability distributions that can be derived from the data.
  + It covers concepts such as Central Limit Theorem. Normal Probability Distribution, Binomial Probability Distribution, Poisson Distribution and many more.
* **Inferential Statistics**: This is a bit of an advanced topic, but a fundamental understanding of Inferential statistics can help us understand the potential power.   
    
  Inferential Statistics is a field of statistics that uses analytical tools to draw conclusions about a poplation by analysing random samples.It can cover the following concepts:
  + **Hypothesis testing**: This covers concepts such as Z-test, F-test, T-test, ANOVA test etc.
  + **Regression Analysis**: Linear Regression, Nominal Regression, Logistic Regression, Ordinal Regression etc.

We have now successfully gone over some of the key statistical abilities of R.

# Excel versus R versus Python

Welcome to this session, where we will get to learn more about R versus Python. We will also touch upon Excel and discuss it as all you may have used Excel at some point.

* Excel
  + Excel is a great starting point to understand some fundamentals of Data Analysis. Being able to see the data in rows and columns and visualize it showcases the power data can have.
  + However, there are some known limitations of Excel such as limitations in the amount of data excel can handle, complex modelling falls short in Excel, collaboration is hindered when using Excel, and there are limitations when handling multiple spreadsheets simultaneously.
* R versus Python
  + **Language**
    - Python is a general purpose language that is used for development as well as deployment of projects
    - R is a statistical language that is used for the analysis and visual representation of data
  + **Use-cases**
    - Python is suitable for machine learning, deep learning and large scale web applications
    - R is suitable for statistical learning and has powerful libraries for data exploration and experimentation
  + **Statistical package support** 
    - Python’s statistical packages are less powerful
    - R’s statistical packages are highly powerful and provide a lot of detailed analysis
  + **Syntax**
    - Python has a simple syntax and is easy to learn
    - R is relatively more complex and the learning curve can take some users time to fully utilize the true potential of R
  + **Application Integration**
    - Python is useful when data analysis needs to be integrated with web applications
    - R is generally used when the data analysis required standalone analysis and processing
  + **Popularity**
    - Python is arguably the most popular language and has a vast user based of developers, programmers and data scientists
    - R is not as popular as Python and it’s user base comprises of scientists and the research and development terms who rely on data analysis. However, it is still one of the main requested tools in Data Scientist interviews and other data-based roles.

Fantastic! You have thoroughly understood the real-world impact of learning R, and where it stands among some of the other popular data analysis tools. Congratualtions on making it so far. We are almost at the most exciting part. Let’s deep-dive into the Data Types in R.

# Data Types and Data Structures in R

Welcome back to this session on Data Types in R. While programming in any lanuage, we use variables to store information. When we have variables, a certain portion of the computer’s memory is reserved for it, based on the variable that we decide on.

If you have learnt any programming language before, most of them declare the type of the variable. This is not the case in R. Rather, the variables are assigned as R Objects and the data type of the R object becomes the data type of the variable. The simplest data type is a vector object. There are six data types of the smallest unit of these vectors, called atomic vectors. They are:

* **Logical**: True and False comprise of the logical data types
* **Numerical**: Numbers such as 1, 2, 99, 34.5 comprise of the numeric data type in R
* **Integer**: The latter “L”, when appended to a number indicates that the number is stored as an interger, instead of a double data type, which takes more space
* **Complex**: Complex numbers such as 3 + 2i can also be stored in R as a complex data type
* **Character**: Text stored within double quotes indicateds that the data is of character type
* **Raw**: The raw data type in R holds raw bytes, and there are very specific use cases for it

Now that we have learnt about the data types, let’s understand more about the data structures in R, which is essentially a comabination of these data types. We will go over the understanding instead of the code in this course. More can be found in the courses later on.

* **Vectors**: Vectors are used when more than once elements are combined into an object, which is denoted with c() function
* **Lists**: Similar to how we can make a shopping list with an assortment of letters and numbers, there is a list() function which does the same thing in R, where different data types can be combined
* **Matrices**: Denoted by a two dimensional rectagular dataset, there is a matrix() function, where vectors can be input into the matrix function. Matrices can be n-dimensional
* **Arrays**: The array is a subset of a matrix. When matrices are confined to two dimensions, they are known as an array. Note, this does not mean two rows or two columns, it simply means that there will only be a n x n matrix
* **Data** **Frames**: A data frame is the tables that we are used to seeing in excel. Data frames are tabular objects. Unlike a matrix, in a data frame, each column can have different data types. It is a list of vectors of equal length. It is created by using the data.frame() function.

Great, you now have a fundamental understanding on the data types and data structures in R. In the next session, we will go over two methods how we can work with R on your laptop.

# Coding with R – Installation of R / R Studio

Alright, welcome to this session where we will learn more about how we can get started with the installation of R on yoru laptop - whether you are running a flavour of Linux, macOS or Windows.

[[Give a demo for R & R Studio]]

* Link to R: <https://cran.rstudio.com/>
* Link to RStudio: <https://posit.co/download/rstudio-desktop/>

# 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.

# 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

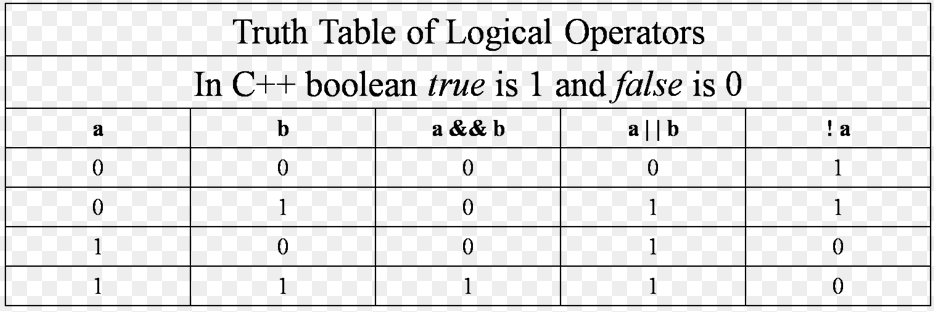
Relational operators are frequently used for conditional filtering of Data in R. We can assign a value to a variable using “<-”. A variable is named unit of data that can be assigned a value. The other main use of Relational operators such as <>=! is to check if the value is greater than, lesser than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



[[Anish will explain the above table]]

# 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.

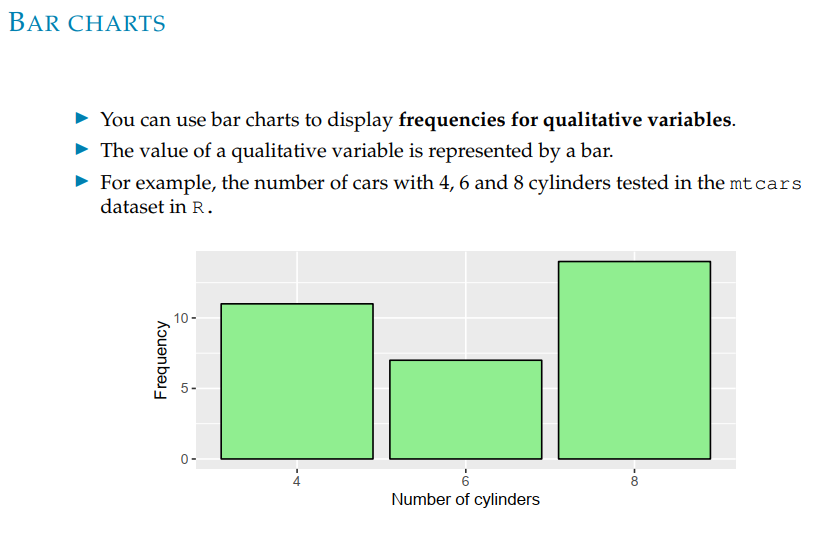
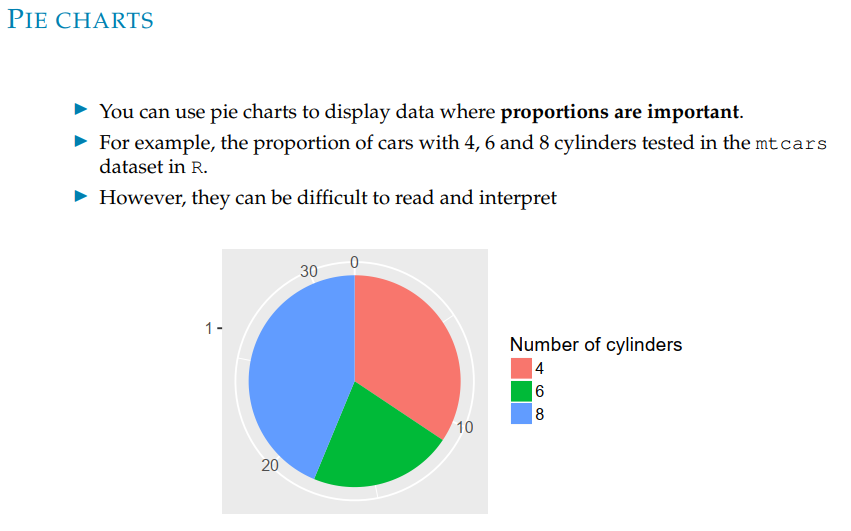
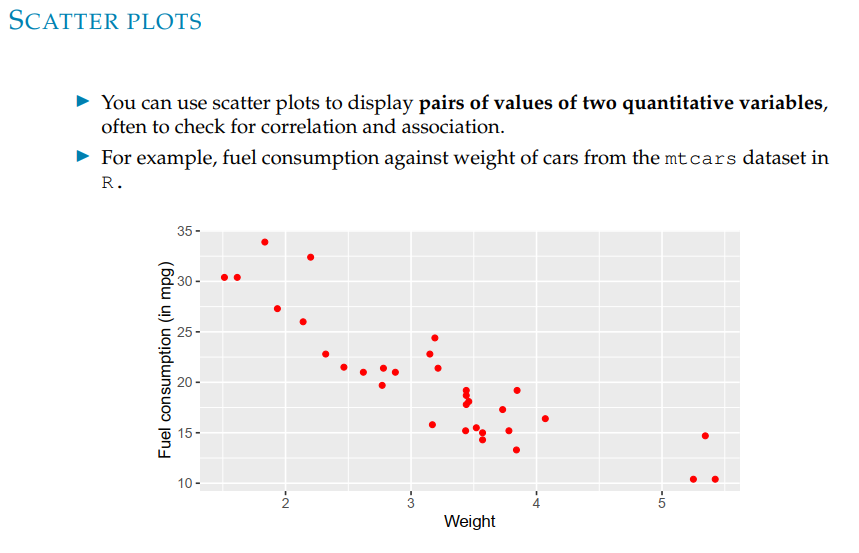
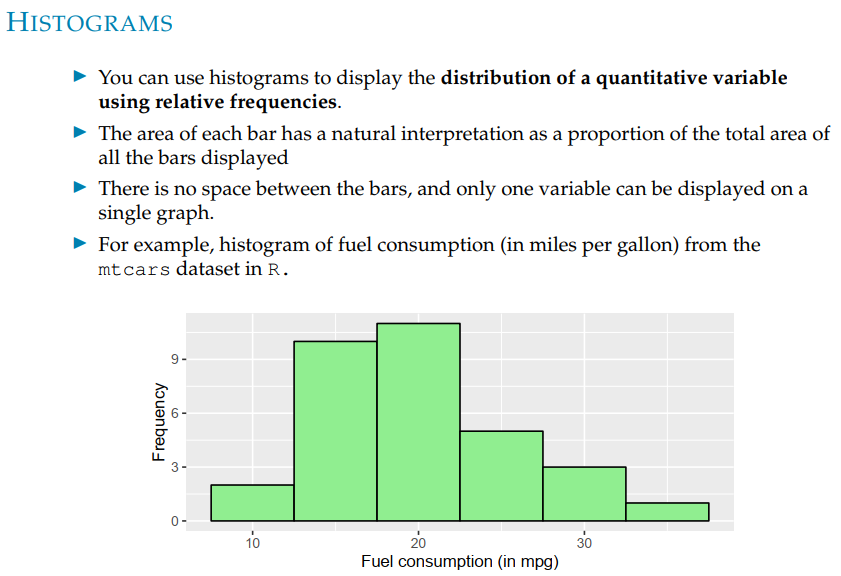
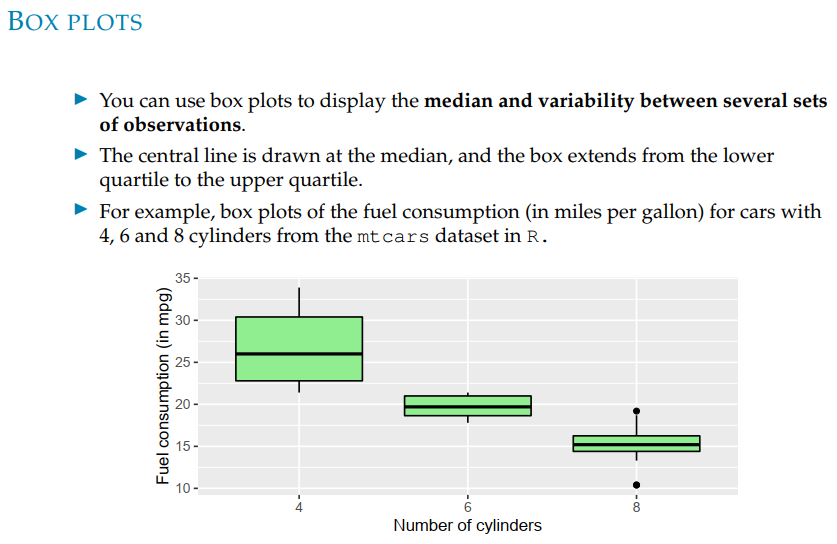
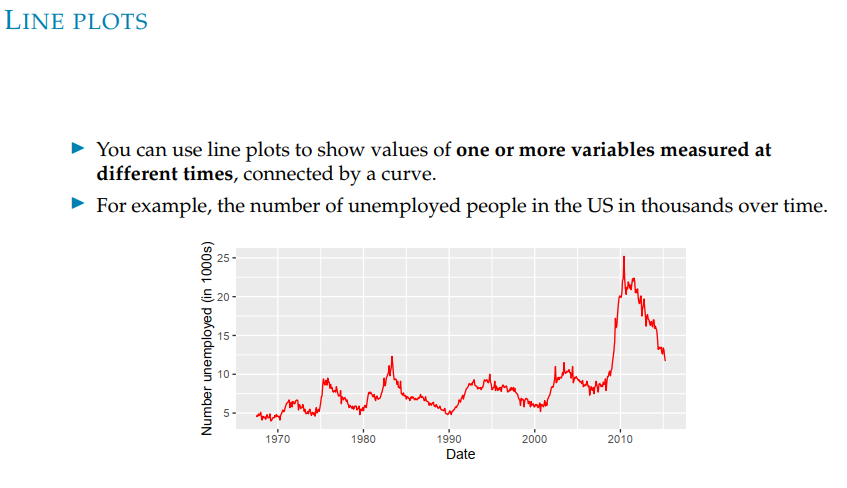
# Visualization with R

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. This 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!

*[[Here, we will need to build custom visualizations to showcase to the learner, or else, I can showcase it in a ppt in case the team does not have time to build it]]*

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.

# Applications of Machine Learning in R

# AI in R

# Thank you!



# Author

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

# Archive

1. Introduction
2. What is Data?
3. What is R?
4. The applications of R - (Talk about RStudio)
5. What is an R Script?
6. What are packages in R
7. Commonly used packages in R
8. R for statistical analysis
9. Excel versus R versus Python
10. Data Types in R
11. Coding with R – Installation of R / R Studio
12. How to read a dataset in R
13. Data Operations in R
14. Data Wrangling in R
15. Visualization with R
16. Applications of Machine Learning in R
17. AI in R
18. Thank you!