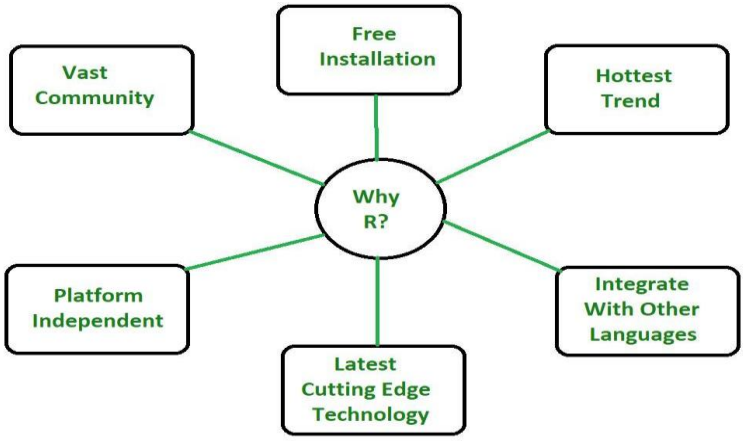
R Programming Language – Introduction

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool.

It was designed by **Ross Ihaka and Robert Gentleman** at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

Why R Programming Language?



- R programming is used as a leading tool for machine learning, statistics, and data analysis. Objects, functions, and packages can easily be created by R.
- It's a platform-independent language. This means it can be applied to all operating system.
- It's an open-source free language. That means anyone can install it in any organization without purchasing a license.
- R programming language is not only a statistic package but also allows us to integrate with other languages (C, C++). Thus, you can easily interact with many data sources and statistical packages.
- The R programming language has a vast community of users and it's growing day by day.
- R is currently one of the most requested programming languages in the Data Science job market that makes it the hottest trend nowadays.

Features of R Programming Language

Statistical Features of R:

- **Basic Statistics:** The most common basic statistics terms are the mean, mode, and median. These are all known as "Measures of Central Tendency." So using the R language we can measure central tendency very easily.
- **Static graphics:** R is rich with facilities for creating and developing interesting static graphics. R contains functionality for many plot types including graphic maps, mosaic plots, biplots, and the list goes on.

- **Probability distributions:** Probability distributions play a vital role in statistics and by using R we can easily handle various types of probability distribution such as Binomial Distribution, Normal Distribution, Chi-squared Distribution and many more.
- **Data analysis:** It provides a large, coherent and integrated collection of tools for data analysis.

Programming Features of R:

- **R Packages:** One of the major features of R is it has a wide availability of libraries. R has CRAN(Comprehensive R Archive Network), which is a repository holding more than 10, 0000 packages.
- **Distributed Computing:** Distributed computing is a model in which components of a software system are shared among multiple computers to improve efficiency and performance. Two new packages **ddR and multidplyr** used for distributed programming in R were released in November 2015.

Programming in R:

Since R is much similar to other widely used languages syntactically, it is easier to code and learn in R. Programs can be written in R in any of the widely used IDE like **R Studio, Rattle, Tinn-R**, etc. After writing the program save the file with the extension **.r**. To run the program use the following command on the command line:
`R file_name.r`

Example:

- R

```
# R program to print Welcome to
GFG!



# Below line will print
"Welcome to GFG!"

cat("Welcome to GFG!")
```

Output:

```
Welcome to GFG!
```

Advantages of R:

- R is the most comprehensive statistical analysis package. As new technology and concepts often appear first in R.
- As R programming language is an open source. Thus, you can run R anywhere and at any time.
- R programming language is suitable for GNU/Linux and Windows operating system.
- R programming is cross-platform which runs on any operating system.
- In R, everyone is welcome to provide new packages, bug fixes, and code enhancements.

Disadvantages of R:

- In the R programming language, the standard of some packages is less than perfect.
- Although, R commands give little pressure to memory management. So R programming language may consume all available memory.
- In R basically, nobody to complain if something doesn't work.
- R programming language is much slower than other programming languages such as Python and MATLAB.

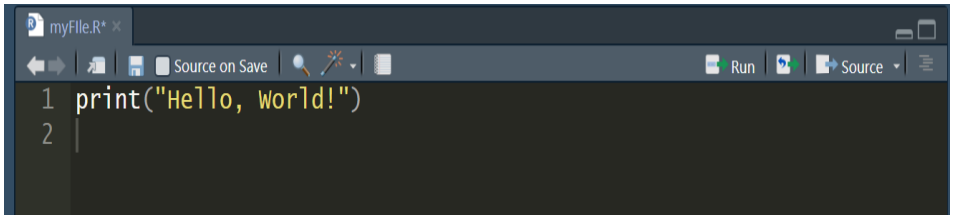
Applications of R:

- We use R for Data Science. It gives us a broad variety of libraries related to statistics. It also provides the environment for statistical computing and design.
- R is used by many quantitative analysts as its programming tool. Thus, it helps in data importing and cleaning.
- R is the most prevalent language. So many data analysts and research programmers use it. Hence, it is used as a fundamental tool for finance.
- Tech giants like Google, Facebook, bing, Twitter, Accenture, Wipro and many more using R nowadays.

Basic Sysntax

Now we can do the same thing using `print()` which prints to the console. Usually, we will write our code inside scripts which are called **RScripts** in R. To create one, write the below given code in a file and save it as **myFile.R** and then run it in console by writing:

```
Rscript myFile.R
```



Output:

```
[1] "Hello, World!"
```

Syntax of R program

A program in R is made up of three things: Variables, Comments, and Keywords. Variables are used to store the data, Comments are used to improve code readability, and Keywords are reserved words that hold a specific meaning to the compiler.

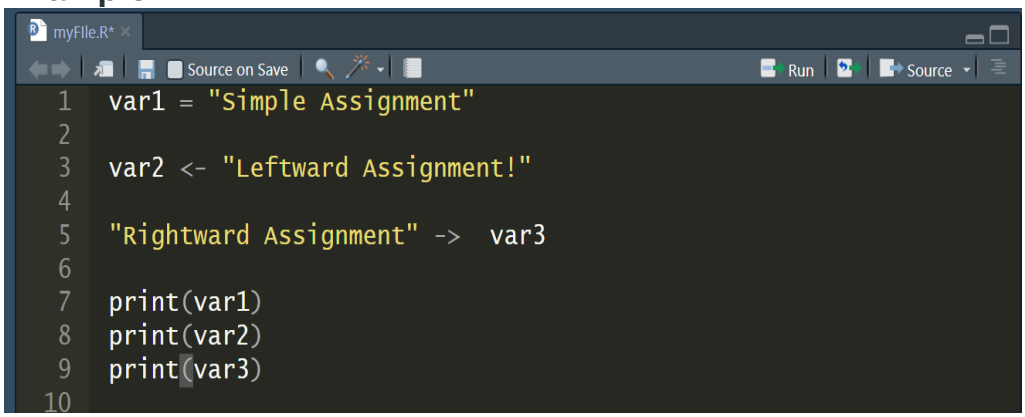
Variables in R

Previously, we wrote all our code in a print() but we don't have a way to address them as to perform further operations. This problem can be solved by using **variables** which like any other programming language are the name given to reserved memory locations that can store any type of data.
In R, the assignment can be denoted in three ways:

1. **= (Simple Assignment)**
2. **<- (Leftward Assignment)**

3. **-> (Rightward Assignment)**

Example:



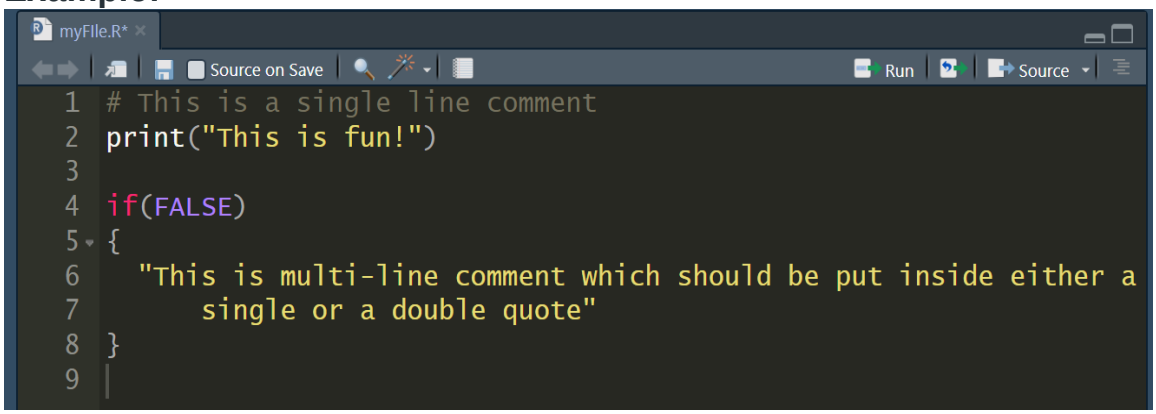
Output:

```
"Simple Assignment"

"Leftward Assignment!"

"Rightward Assignment"
```

Comments in R

Comments are a way to improve your code's readability and are only meant for the user so the interpreter ignores it. Only single-line comments are available in R but we can also use multiline comments by using a simple trick which is shown below. Single line comments can be written by using # at the beginning of the statement.

Example:



Output:

```
[1] "This is fun!"
```

From the above output, we can see that both comments were ignored by the interpreter.

Keywords in R

Keywords are the words reserved by a program because they have a special meaning thus a keyword can't be used as a variable name, function name, etc.

We can view these keywords by using either `help(reserved)` or `?reserved`.

Reserved words in R

if	else	while	repeat	for
function	in	next	break	TRUE
FALSE	NULL	Inf	NaN	NA
NA_integer_	NA_real_	NA_complex_	NA_character_	...

- if, else, repeat, while, function, for, in, next and break are used for control-flow statements and declaring user-defined functions.
- The ones left are used as constants like `TRUE/FALSE` are used as boolean constants.
- `NaN` defines Not a Number value and `NULL` are used to define an Undefined value.
- `Inf` is used for Infinity values.

Note: *R is a case sensitive language so* `TRUE` *is not same as True.*