# <u>WEEK 1: TUT SHEET 1</u>
# INTRO TO R SYNTAX

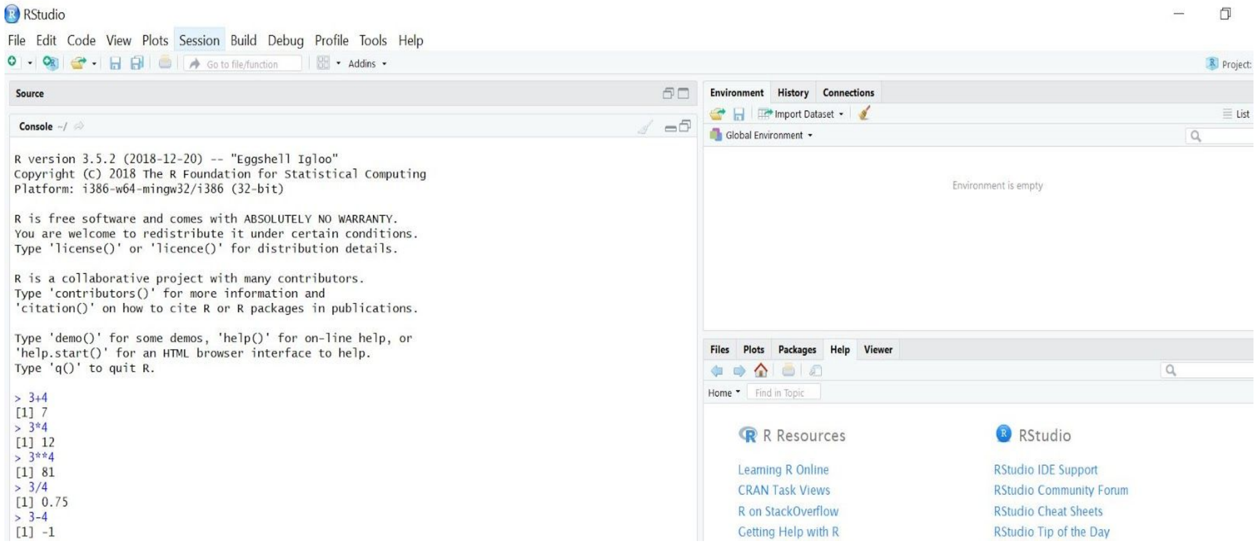
In this tutorial we'll start with learning basic syntax of the language.

**Please note that R does not require semicolon to end a statement.**

## <u>BASIC CALCULATIONS:</u>

In the console window of your Rstudio, you can perform basic mathematical calculations. Press **ctrl + L** to clear the console screen.

Exercise: Try out basic operations. Press Enter to see the output in the console.
NOTE: The operations follow the standard order like bodmas.



```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 3+4
[1] 7
> 3*4
[1] 12
> 3**4
[1] 81
> 3/4
[1] 0.75
> 3-4
[1] -1
```

## <u>COMMENTS</u>:

Just like any other programming language, you can add comments to make your code more readable. Use **#** to add a comment.

## <u>VARIABLES:</u>

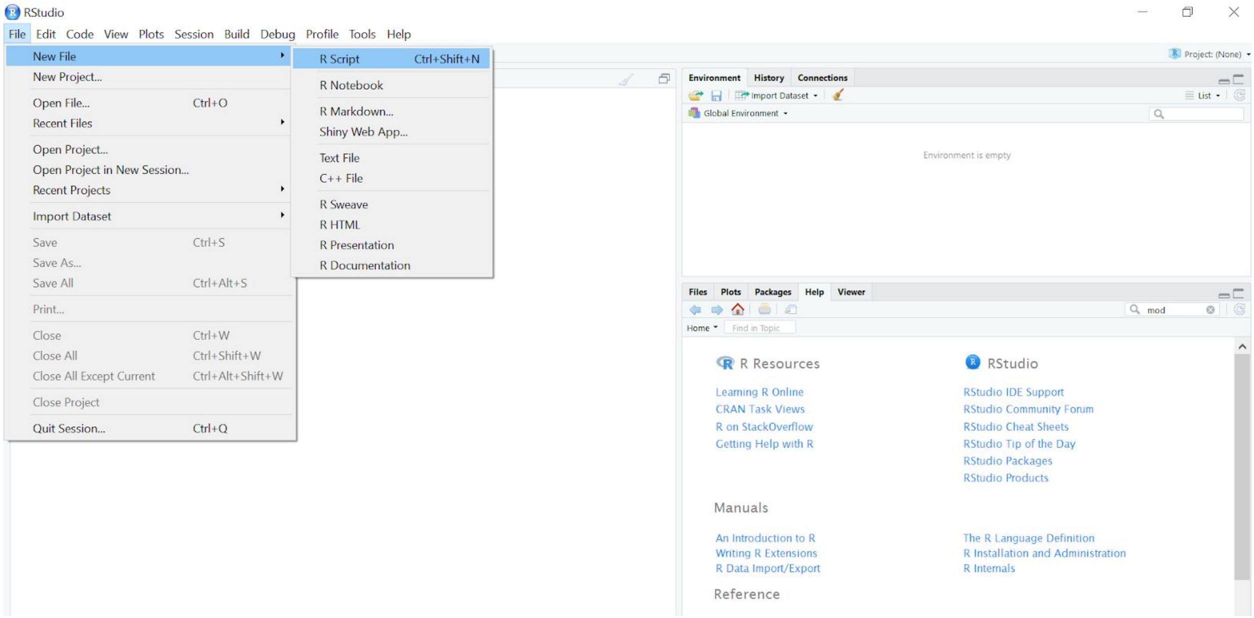
In R, we assign variables using assignment operator, <- (arrow made using lesser than and hyphen) or  = (equality).

Exercise: Assign a variable with your name as string. Print it to see the output.
Eg: print(name) where name<-'abcd'

# R STUDIO GUIDE:

To save your work in RStudio **create a new R script** , type and save the file. The file will be saved as .R extension. Use **Open File** option to open the file. **Recent files** option display the files you opened recently.

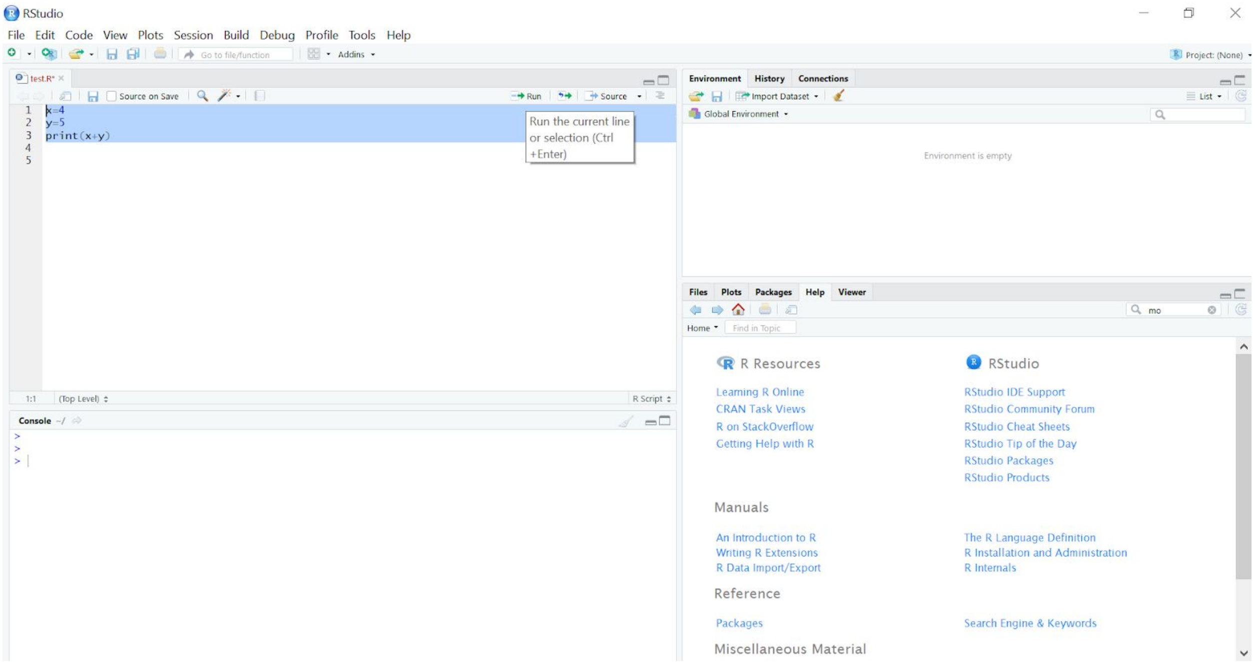
You can combine your work consisting of many files in a project. Create project using **New Project** option and select the appropriate directly. You can access your project by **Open Project** option. **Use Recent Projects** option to get list of recently opened projects.
**R Markdown** allows you to document your work about which we will talk later.



NOTE: Try exploring other options of RStudio, you will learn more options as we will start doing projects.

There are three other tabs: Environment, History and Connections.
**Environment** just stores the variables. **History** gives you the commands that you typed on the console. **Connection** allows you to connect to various data sources.

To **run** code from R file, select the part of code you want to run and press the **run** button. Output will be displayed in the console window.



```
x=4
y=5
print(x+y)
```

Run the current line or selection (Ctrl +Enter)

Environment is empty

## **DATA TYPES:**

They are used to classify information. It's major types are:

1. Numeric: Any number with or without a decimal point. Eg: 2, 3.14 etc.
2. Character: Strings are formed using single or double quotes.
3. Vectors: A list of data that is all of same type.
4. Logical: This data type has only two possible values TRUE or FALSE.

NOTE:Use print() with appropriate argument to print the output in the console.

## **VECTORS:**

Vectors are a list-like structure that contain items of the same data type.

For instance: colours<- c("red", "blue", "green", "yellowl")

The c() is used to sort of wrap the data items in the vector.

typeof(vector_name): For checking the type of elements present in the vector.

length(vector_name): For finding out the length, i.e. the number of elements present on the vector.

Vector_name[i]: For accessing the individual data item present at the ith location. Note that i here is 1-indexed.

Exercise: Create a vector with months of the year as its elements. Print the length of the vector and the 7th month.

## CONDITIONAL STATEMENT:

This is similar to other programming languages.

```
If(true-condition){
 #perform some action
} else{
 #perform some action
}
```

## COMPARISON OPERATORS:

Less than: <
Greater than: >
Less than or equal to: <=
Greater than or equal to: >=
Is equal to: ==
Is NOT equal to: !=

The console returns logical true or false depending upon the comparison.

Exercise: Use comparison operators to find that 123 is greater than 17.

## LOGICAL OPERATORS:

The and operator: &
The or operator: |
The not operator: !