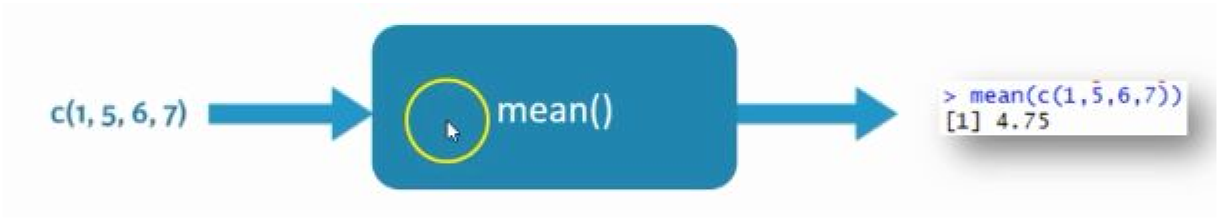
Function in R:

What is function?

The function is exactly the same concept as any other language it’s like a Black Box, you give input and based on the function it tries to solve and it would give you the output.

gular Snip
INPUT
BLACK BOX
OUTPUT

Like, you have a function called mean(), it gets as input and whatever it's given,it provides you the output. Similarly, in R a function called mean() and if you give the input as 1,5,6,7 and you combining all of them by combine function and you would get the output as 4.75.

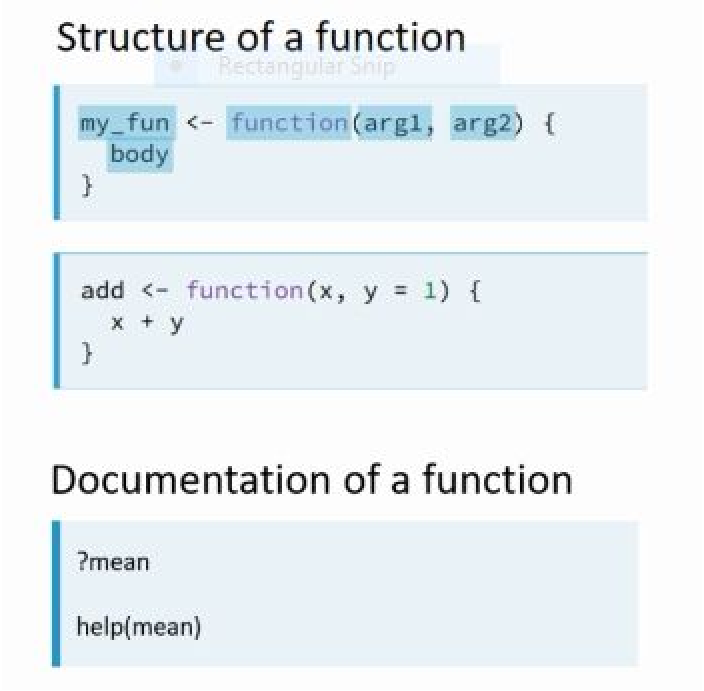


```
> mean(c(1,5,6,7))
[1] 4.75
```

By using the keyword ‘function’ the R function is created.

Function Structure & Documentation:

Now, R has a lot of inbuilt function like Numeric function, Statistical Function and Character functions and lots of other function as well. Before we actually deep drive in those function let’s understand what is the structure of a function.



Structure of a function

```
my_fun <- function(arg1, arg2) {
  body
}
```

```
add <- function(x, y = 1) {
  x + y
}
```

Documentation of a function

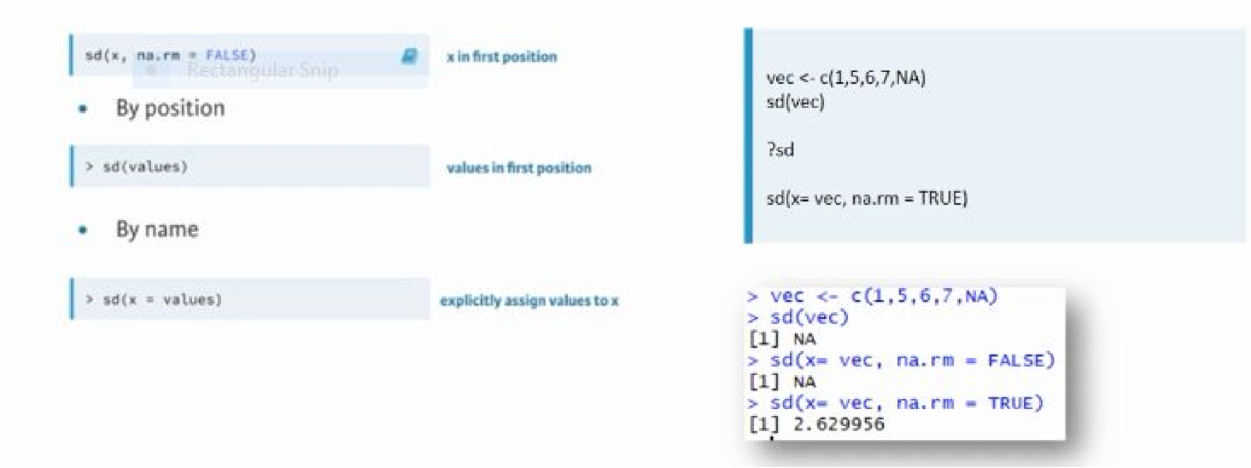
```
?mean

help(mean)
```

Therefore, the function has a body where you have written all those things. There is another function we are trying to create all those things which are about added, so if you pass two argument X and Y then we will add them and return the value. Now, by default you can set some values and if you set X, Y then there is so no default value but if you set X, Y=1 then by default you are saying that even you don't pass anything in Y value, and suppose you pass 2 in this function then it would give the result 2+1=3.

So, we will be learning all those things, like, How to write a function on your own but before we learn all those things, we will actually learn a couple of inbuilt function in R, which are very popular and we are actually using all those functions in our subsequent courses.

Function Arguments Matching:

In the fifth or sixth position argument is very difficult to remember that's why, whenever you are passing any value to a function, you can pass it two way either By Position where you say the first position is value, the second one is na.rm. But if you forget it by any chance then you just remember it By Name. so, if you have two arguments and suppose in Standard Deviation.



```
sd(x, na.rm = FALSE)
```

x in first position

- By position

```
> sd(values)
```

values in first position

- By name

```
> sd(x = values)
```

explicitly assign values to x

```
vec <- c(1,5,6,7,NA)
sd(vec)

?sd

sd(x= vec, na.rm = TRUE)
```

```
> vec <- c(1,5,6,7,NA)
> sd(vec)
[1] NA
> sd(x= vec, na.rm = FALSE)
[1] NA
> sd(x= vec, na.rm = TRUE)
[1] 2.629956
```

Then in the fifth position, there is an argument which you want to change. Instead of position, you can choose By name so you can say na.rm equal to False or True and it would understand.

Introduction to Function:

Now, we are talking about lots of Inbuilt Function. There is a couple of Numeric Function, a couple of Statistic Function and couple of Character Function as well.

At first, we are discussing Numeric Function.

Numeric Function:

We are learning the Numeric Function from R Studio. We are talking about its various function. The first thing is a couple of easier one **sqrt(x).**

Sqrt(x): It is a numeric value or a valid numeric expression for which you want to see square root. If the Numeric Expression is positive value then sqrt() function will go back to the square root of a given value.

If the numeric expression is negative value then sqrt() function will return NaN.

If the numeric expression is not a number(NaN), or negative infinity then sqrt in R will return NaN.

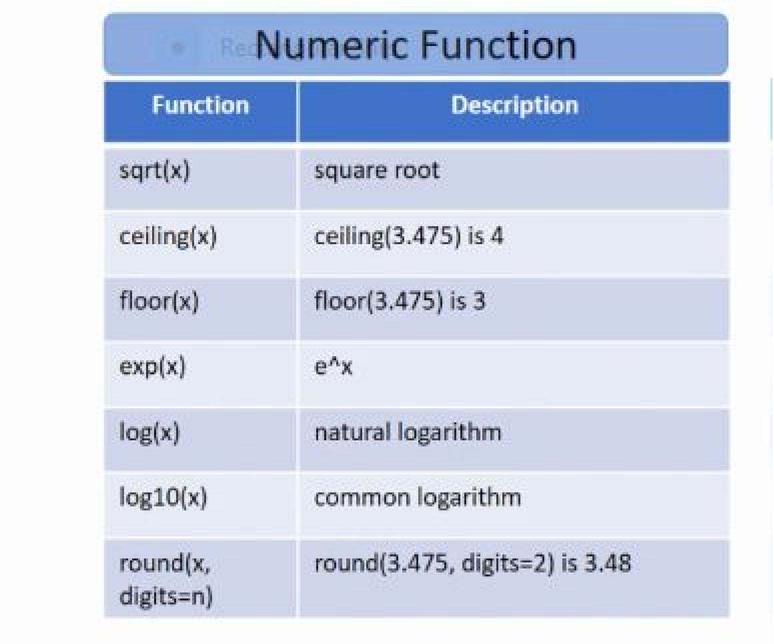
If the numeric expression is positive infinity then sqrt function will return the result as positive infinity.

Ceiling(x): It is one of the R Math Function which is used to return the smallest integer value that is greater than or equal to an individual number, or an expression. It can be a numeric value for which you want to find the square root. If the numeric Expression is positive or negative numeric value, then ceiling() function will return the ceiling value.

If the numeric expression is positive or negative zero then ceiling() function will return zero.

If the numeric expression is not a number(NaN) therefore ceiling will go back to NaN.

If the numeric number is positive or negative infinity then the function will return the same.

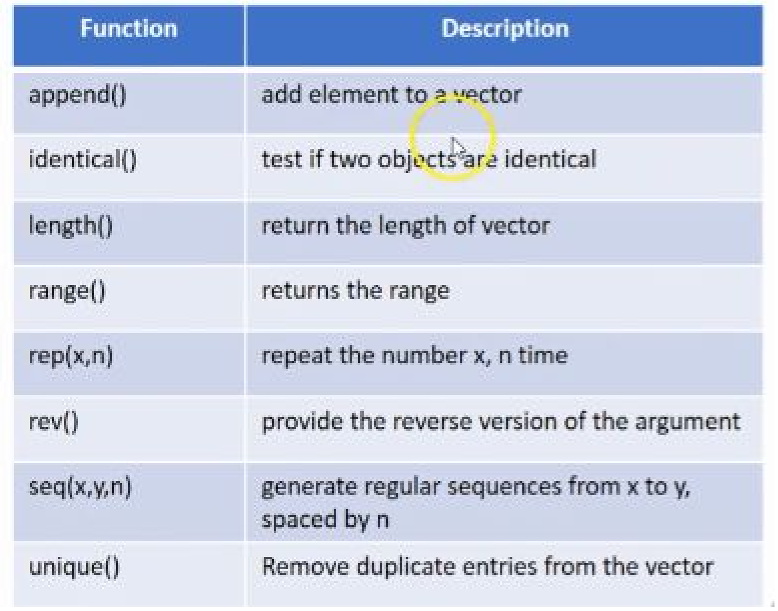


Numeric Function

Function	Description
sqrt(x)	square root
ceiling(x)	ceiling(3.475) is 4
floor(x)	floor(3.475) is 3
exp(x)	e^x
log(x)	natural logarithm
log10(x)	common logarithm
round(x, digits=n)	round(3.475, digits=2) is 3.48

Floor(x): The R Floor method is one of the R Math Function that is used to return the largest integer value which is not greater than(less than) or equal to an individual number or an individual expression.
It can be a numeric value for which you want to find the square root. If the numeric expression is positive or negative numeric value then floor function will return the floor value.
If the numeric expression is positive or negative zero then the function will return zero.
If the numeric expression is NaN(not a number), therefore floor function will return NaN.
If the numeric expression is positive or negative infinity, therefore the function will return the same.
Exp(x): The function Exp() defines exponential distribution,a one-parameter distribution for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The mu parameter represents the mean of a distribution. The functions are dEXP,pEXP,qEXP and rEXP define the density, distribution function, quantile function and random generation for the specific parameterization of the exponential distribution defined by Exp() function. “Keywords” is “distribution” “regression”
Log(x): log computes logarithms, by default natural logarithms, log10 computes common (that is base 10) logarithms, and log2 computes binary (that is base 2) logarithms. The general form log(x, base) computes logarithms with the base.
Round(x,digits=n): Round is classified in some steps like,
Ceiling: It takes a single numeric argument x and returns a numeric vector containing the smallest integers not less than the corresponding elements of x.
Floor: It takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x.
Trunc: It takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

Round: rounds the values in its first argument to the specified number of decimal places (default 0)
Signif: rounds the values in its first argument to the specified number of significant digits.
Append(): Append values to x, probably inserted into the middle of x. This function is important since its trains to perform a little faster than using the concatenation (c) function.
Identical(): The safe and reliable way to tests two objects for being equal True in this case and False in every other case.
Length(): Get the length of vectors and factors, and any other R object for which a method has been defined.
Range(): Range returns a vector containing the minimum and maximum of all the given arguments. The range is a generic function, its methods can be defined for it directly or via the summary group generic. Its arguments should be unnamed and dispatch is on the first argument. The keyword is “arith” , “Univar”.



Function	Description
append()	add element to a vector
identical()	test if two objects are identical
length()	return the length of vector
range()	returns the range
rep(x,n)	repeat the number x, n time
rev()	provide the reverse version of the argument
seq(x,y,n)	generate regular sequences from x to y, spaced by n
unique()	Remove duplicate entries from the vector

Rep(x,n): Rep reproduces the values in X. It is a generic function and the default method is described here. The keyword is “manip”, “chron”.
Rev(): Rev provides a reverse version argument. It is a generic function with a default method for vectors and one for dendrograms.
Seq(x,y,n): Generated regular sequences,seq is a standard generic with a default method. Seq.int is a primitive that can be much faster but has a few restrictions. Seq.along and seq.len are very fast primitives for two common cases. The keyword is “manip”.
Unique(): Unique goes back to a vector, data frame or array-like x but with duplicate elements removed. The keyword is “manip”, “logic”.