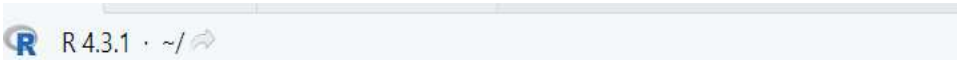
Third: The toolbar

This bar contains the most important and frequently used icons, as shown in the figure below:

- ✧ Open : Open a new worksheet in RStudio.
- ✧ Open Script : Open a file in the program.
- ✧ Save\Save As : Save the workspace.
- ✧ Print .

Fourth - Statusbar

It appears at the bottom of the screen and writes the version type for R and the year.



❖ Special commands in the R programming language

** Mathematical and Logical Operators

Before starting to explain the arithmetic and logical operations, there are some notes about the program:

- R is a case sensitive programming language. This means A and a are different symbols and would refer to different objects.
- Objects in R can be named using letters and numbers (example: A, d, xx, y1, M2); however, we cannot use numbers only to name variables or start names with a number.
- Object names cannot contain symbols like (!, +,-, #) .
- You may use "=" or "<-" to assign a value to your variable (example: a = 5, or a <- 5).
- To add a comment to your code, start your comment with '#'.
- To get information about a specific function, use help(function) (example: help(sum)).

- Arithmetic: R uses the usual symbols for addition +, subtraction-, multiplication *, division /, and exponentiation ^. Parentheses () can be used to specify the order of operations.
- Functions: R has a number of built-in functions, for example exp(x), sqrt(x), and log(x).
- Sequences: We can use R to create objects with certain patterns, for example 1, 2, 3, ..., 100, and 3, 6, 9, ..., 300.
- Vectors and Matrices: R allows us to input our data in vectors and/or matrices. We can recall any element in vectors or matrices using brackets "[]", for example v[3] is the third element in the vector v, and m[4,2] is the element that located in the 4th row and the 2nd column of the matrix m

**Arithmetic Functions and Logical Operations R

The table below shows some basic arithmetic and logical operations:

Mathematical Operators		العمليات الحسابية
2^3 = 8	** or ^	Power or Exponent
5 * 5 = 25 4/2 = 2	* or /	Multiplication and Division
7%%3=1 , 7%/%3=2	%% , %/%	الصحيحة والقسمة القسمة باقي Division Integer and Division Remainder
3+1=4 , 3-1=2	+ , -	Subtraction and Addition
Logical Operators		العمليات المنطقية (عمليات المقارنة)
==		Equality
= !		Inequality
>		Greater
<		Lesser
>=		Greater and equal to
<=		Lesser and equal to
&		Logical "and"
|		Logical "or"

Example :

```
5 + 3 / 2
[1] 6.5
(5 + 3) / 2 # use parentheses to control order of operations.
```

استخدم الأقواس للتحكم في ترتيب العمليات.

```
[1] 4
2^0.5 # raise to power 0.5 to calculate the square root.
[1] 1.414214-7 * 5- 5
[1]-40
x <- 4
x <- x + 1
x
## [1] 5
x^3
## [1] 125
```

Example :

```
> 3==5
[1] FALSE
> 3!=5
[1] TRUE
> 3<5
[1] TRUE
> 3>5
[1] FALSE
> 3<=5
[1] TRUE
> 3>=5
[1] FALSE
> TRUE&FALSE
[1] FALSE
> TRUE|FALSE
[1] TRUE
```

Example :

```
> x=c(4,6,1)
> y=c(3,7,2)
> x>y
[1]  TRUE FALSE FALSE
> x<y
[1] FALSE  TRUE  TRUE
> x==y
[1] FALSE FALSE FALSE
```

Example:

```
> z<- c(TRUE,FALSE,0,7)
> t <- c(FALSE,TRUE,FALSE,TRUE)
> !x
[1] FALSE
> !z
[1] FALSE  TRUE  TRUE FALSE
> z&t
[1] FALSE FALSE FALSE  TRUE
> z|t
[1]  TRUE  TRUE FALSE  TRUE
```

The previous operations have priority in execution as mentioned in the table above in order, but the parentheses in the mathematical operation always have priority in execution.

Example :

```
x=6/2*(1+2)
```

Result (9)

```
Y=6/(2*(1+2))
```

Result (1)

**Mathematical Functions

In the R program, a wide range of pre-built mathematical functions can be used. Th ese functions are important tools for performing mathematical calculations and pro cessing. The most prominent of these are:

abc(x) : Absolute value.

sqrt(x) : Square root.

log(x) : base of the natural logarithm e .

Log10(x) : common logarithm to base 10.

Example :

```
> abs(-7)
[1] 7
> sqrt(16)
[1] 4
> exp(1)
[1] 2.718282
> log(2.718)
[1] 0.9998963
> log10(100)
[1] 2
```

****Trigonometric functions in R**

You can also use trigonometric functions such as (sin, cos, tan) and inverse trigono metric functions (asin, acos, atan). For example:

Example :

```
> sin(pi)
[1] 1.224606e-16
> cos(0)
[1] 1
> asin(1)
[1] 1.570796
```