**R Programming \_Basic\_Task 1**

1. Vector recycling

Example1: When applying an operation to two vectors that require them to be the same length, R automatically recycles.

> x = c(1,2,3)

> y = c(4,5,6)

> total = x + y

> print(total)

[1] 5 7 9

Example2: When x length is not equal to y, but the length of y is equal to the number of multiplication of x.

> x = c(1,2,3)

> y = c(8,5,6,4,7,9)

> total = x + y

> print(total)

[1] 9 7 9 5 9 12

Example3: Suppose we have two Vectors c(1,2,3), c(8,5,6,4,7), where the first one is shorter with only 3 elements. Now if we sum these two, we will get a warning message as follows.

> x = c(1,2,3)

> y = c(8,5,6,4,7)

> total = x + y

Warning message:

In x + y : longer object length is not a multiple of shorter object length

1. Inner multiplication

Usage: x %\*% y

Vector1 has x digit and vector2 must have the same digit as vector1, if not it will show the error. The result comes from the sum of the multiplication of the same row and the same column.

Example1: using x for vector1 and vector2

> x <- 1:8

> x%\*%x

[,1]

[1,] 204

Example2: using another notation (u) as vector2 and assign a different number (u <- 3:10)

a.

> x <- 1:8

> u <- 3:10

> x%\*%u

[,1]

[1,] 276

b.

> x <- 1:8

> u <- c(1,2,3,4,5,6,5,8)

> x%\*%u

[,1]

[1,] 190

Example3: Error

> x <- 1:8

> u <- 1:3

> x%\*%u

Error in x %\*% u : non-conformable arguments

1. Outer multiplication

X %o% Y

Usage: Multiply array x element with array y element and the result shows in the metric based on the column and row, the first vector is displayed in the row and the second vector is displayed in the column.

Example1:

> x <- 1:7

> y <- 2:5

> x %o% y

[,1] [,2] [,3] [,4]

[1,] 2 3 4 5

[2,] 4 6 8 10

[3,] 6 9 12 15

[4,] 8 12 16 20

[5,] 10 15 20 25

[6,] 12 18 24 30

[7,] 14 21 28 35

Example2:

> x <- 1:7

> y <- 2:5

> y %o% x

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] 2 4 6 8 10 12 14

[2,] 3 6 9 12 15 18 21

[3,] 4 8 12 16 20 24 28

[4,] 5 10 15 20 25 30 35

1. Functions
   1. sample() is used to randomly determined number, if you try this function repeatedly, you’ll get different results every time. In R, you use the set.seed() function to specify your seed starting value. The argument to set.seed() is any integer value.

Example1: use sample() only

> sample(4:9)

[1] 8 6 5 7 4 9

> sample(4:9)

[1] 4 8 9 5 6 7

Example2: use sample() with set.seed()

> set.seed(1)

> sample(1:6)

[1] 1 4 3 6 2 5

> set.seed(1)

> sample(1:6)

[1] 1 4 3 6 2 5

* 1. seq() is used to generate the sequence number.

> seq(8)

[1] 1 2 3 4 5 6 7 8

> seq(1:6)

[1] 1 2 3 4 5 6

> seq(3,7,3)

[1] 3 6

* 1. rep() is used to repeat the value.

> rep(1:3, 3)

[1] 1 2 3 1 2 3 1 2 3

* 1. round() is used to cut off the number that is lower than .5 and will increase if it is more than .5.

> round(4.59)

[1] 5

> round(2.7654,2)

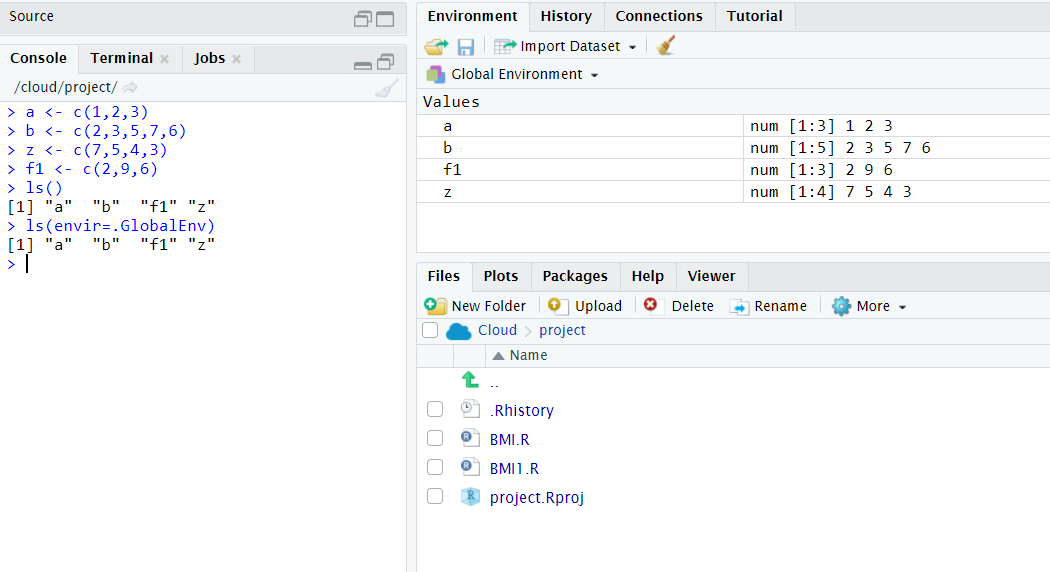
[1] 2.77

* 1. factorial() is used to calculate the factorial of input.

> factorial(3)

[1] 6

* 1. ls() show us the list of objects in global environment



* 1. mean() is used to calculate the average

> mean(2:6)

[1] 4

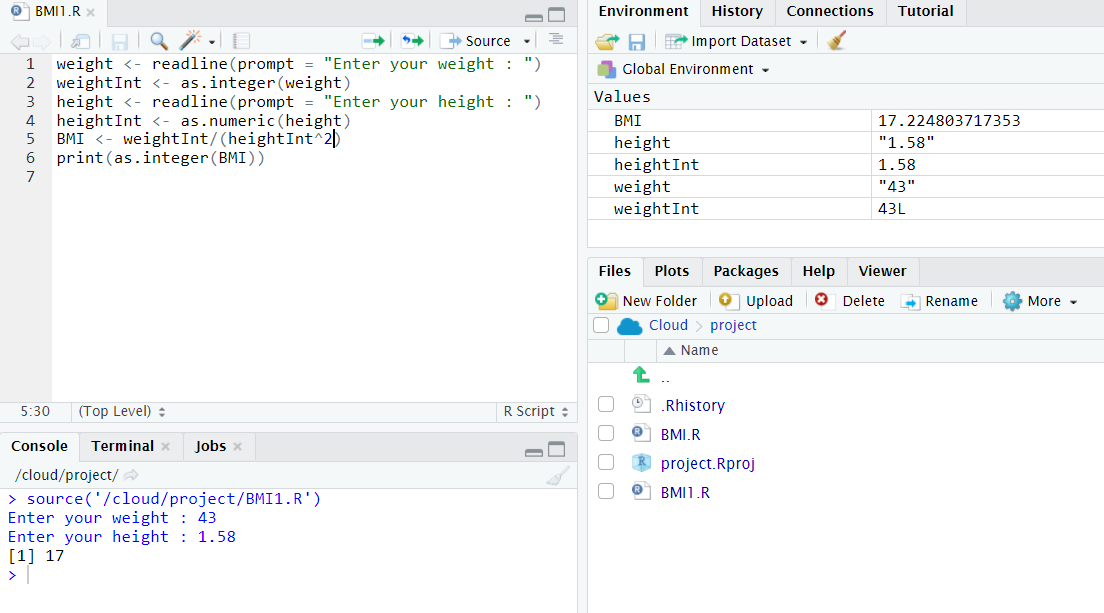
1. Subset is used to find the index and remove the element

> x <- 2:8

> x[2]

[1] 3

1. Write a program to calculate the BMI rate?Get the user input & Result should be in integer:



1. Create a function to calculate the BMI Rating? Result should be in integer:

