# Problem Statement

1. What are the different methods to call a function in R?

|  |
| --- |
| > ##call(name, ...) # returns an unevaluated function call  > cl <- call("round", 22.3)  > is.call(cl) #[1] TRUE  [1] TRUE  > eval(cl) #[1] 22  [1] 22  > ##do.call # for calling a function by name and argument list  > a<-list((2.5:9),6.8)  > a  [[1]]  [1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5  [[2]]  [1] 6.8  > cl1 <- do.call("round",a)  > is.call(cl1) # TRUE  [1] FALSE  > cl1  [1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5  > eval(cl1)  [1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5  > # Recursive function to find factorial  > recursive.factorial <- function(x) {  + if (x == 0) return (1)  + else return (x \* recursive.factorial(x-1))  + }  > recursive.factorial(5)  [1] 120 |
|  |
| |  | | --- | | > | |

1. The lazy evaluation of a function means, the argument is evaluated only if it is evaluated only if it is used inside the body of the function. Say True or False.

##TRUE.

#By default, R function arguments are lazy, they’re only evaluated if they’re actually used.

1. **#State True or False:**
2. #Insights driven from descriptive analytics is not meaningful.

##FALSE

1. #The number of values in each Elements of a list, should be equal.

##FALSE

1. #The datasets are not stored in memory of the computer using R.

##FALSE

1. #Data frames and matrices are two dimensional however the array is multidimensional.

##TRUE