**Session 1 - Introduction**

**Assignment -2**

Problem Statement:

1. How many ways are there to call a function in R?

***Answer***:

Different ways to call a function in R

Method 1:

##call(name, ...)

# returns an unevaluated function call

> cl <- call("round", 55.7)

> is.call(cl)

[1] TRUE

> cl

round(55.7)

> eval(cl)

[1] 56

Method 2:

##do.call

# for calling a function by name and argument list

> a<-list((1:8),10)

> cl1 <- do.call("round",a)

> cl1

[1] 1 2 3 4 5 6 7 8

> eval(cl1)

[1] 1 2 3 4 5 6 7 8

1. Is the below statement true?

* The lazy evaluation of a function means, the argument is evaluated only if it is used inside the body of the function.

***Answer***: true

Arguments to functions are evaluated lazily, so they are evaluated only as needed in the body of the function.

In this example, the function f() has two arguments: a and b .

> f <- function(a,b){ a^2 }

> f(3)

[1] 9

This function never actually uses the argument b , so calling f(2) will not produce an error because the 2 gets positionally matched to a .

1. Mention true or false for below statements:
2. Insights driven from descriptive analytics is not meaningful.

***Answer***: false

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

***Answer***: false

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

***Answer***: false

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

***Answer***: true

[Explanation - Arrays can have any number of dimensions, but every entry has to have the same type.

Data frames are two-dimensional, but each column is allowed to have its own type (while every item within a particular column has to be of the same type). A matrix is a two-dimensional data structure. All the elements of a matrix must be of the same type (numeric, logical, character, complex).]