#Assignment\_3.3 - Session - 3

#Q1: Test whether two vectors are exactly equal (element by element).

#vec1 = c(rownames(mtcars[1:15,]))

#vec2 = c(rownames(mtcars[11:25,]))

#Solution 1:

vec1 = c(rownames(mtcars[1:15,]))

vec1

vec2 = c(rownames(mtcars[11:25,]))

vec2

#we use the functions identical(), all.equal(), isTRUE(), setequal() to test wheather two vectors are exactly equal

all.equal(vec1,vec2)

identical(vec1,vec2)

isTRUE(all.equal(vec1,vec2))

setequal(vec1,vec2)

vec1 %in% vec2

#Q2: Sort the character vector in ascending order and descending order.

#vec1 = c(rownames(mtcars[1:15,]))

#vec2 = c(rownames(mtcars[11:25,]))

#Solution 2:

vec1 = c(rownames(mtcars[1:15,]))

vec1

vec2 = c(rownames(mtcars[11:25,]))

vec2

#use sort() function to sort in ascending and descending order

#Ascending:

sort(vec1)

sort(vec2)

#Descending:

sort(vec1,decreasing = T)

sort(vec2,decreasing = T)

#Q3: What is the major difference between str() and paste() show an example.

#Solution 3:

#str()

#display the structure of an arbitrary object

#it comactly display the internal structure of an R object

#a diagnostic function and an alternative to summary

#it displays many useful things

#ex:

a<- c("1","2","3","hey")

a

str(a)

#str function does not return anything, for efficiency reasons. The obvious side effect is output to the terminal.

#paste()

#used for Concatenate Strings

#paste (., sep = " ", collapse = NULL)

#ex:

x <- c('My.name.is.PRANOY.RAY','learning.Data.Analytics')

x

con <- paste(x[1],x[2],sep = ",")

con

#Q4: Introduce a separator when concatenating the strings.

#Solution 4:

x <- c('Machine.Learning.in.R','Data.Analytics.in.R')

x

#use paste() function to concatenate strings and seperator = -

con<- paste(x[1],x[2],sep = "-")

con