ACADGILD

SESSION 5: Data Management Using R

Assignment 2

Problem Statement

1. Test whether two vectors are exactly equal (element by element).

vec1 = c(rownames(mtcars[1:15,]))
vec2 = c(rownames(mtcars[11:25,]))

R-Script

```
vec1 = c(rownames(mtcars[1:15,]))
vec1
vec2 = c(rownames(mtcars[11:25,]))
vec2

a1<- as.numeric(vec1)
a1
a2<- as.numeric(vec2)
a2
#we use this function
identical(a1,a2)
all.equal(a1,a2)
identical(vec1,vec2)
isTRUE(all.equal(vec1,vec2))
setequal(vec1,vec2)
a1 %in% a2
```

Output:

identical(a1,a2)

```
> identical(a1,a2)
[1] TRUE
```

all.equal(a1,a2)

```
> all.equal(a1,a2)
[1] TRUE
```

identical(vec1,vec2)

```
> identical(vec1,vec2)
[1] FALSE
```

isTRUE(all.equal(vec1,vec2))

```
> isTRUE(all.equal(vec1,vec2))
[1] FALSE
```

setequal(vec1,vec2)

```
> setequal(vec1,vec2)
[1] FALSE
```

a1 %in% a2

```
> a1 %in% a2
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
TRUE
```

Problem Statement

2. Sort the character vector in ascending order and descending order.

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

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

Answer 2

```
vec1 = c(rownames(mtcars[1:15,]))
vec1
a1<- as.numeric(vec1)
a1
vec2 = c(rownames(mtcars[11:25,]))
vec2
a2<- as.numeric(vec2)
a2

#sort in ascending order by default
sort(a1)
sort(a2)

#sort in descending order
sort(a1,decreasing = T)
sort(a2,decreasing = T)
```

Problem Statement

3. What is the major difference between str() and paste() show an example.

#str()

#display the structure of an arbitrary object

#ex:

#it compactly display the internal structure of an R object

#a diagnostic function and an alternative to summary

#it displays many useful things

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

#say for cs2m dataset
#str(cs2m)

#ex
#Join multiple strings into a single string.
library(stringr)
str_c("Letter: ", letters)
str_c("Letter", letters, sep = ": ")
str_c(letters, " is for", "...")
str_c(letters[-26], " comes before ", letters[-1])
```

```
str_c(letters, collapse = "")
str_c(letters, collapse = ", ")

#ex
hw <- "heyyyy Vadiv"

str_sub(hw, 1, 6)
str_sub(hw, end = 6)
str_sub(hw, 8, 14)
str_sub(hw, 8)
str_sub(hw, c(1, 8), c(6, 14))
```

#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.Vadivazhagan','learning.Data.Analytics')
x
con_str<-  paste(x[1],x[2],sep = ",")
con_str
```

Problem Statement

4. Introduce a separator when concatenating the strings.

#Answer 4

```
x <- c('My.name.is.Vadivazhagan','learning.Data.Analytics')
x

y<- c(gsub(".","-",x,fixed = TRUE))
y

#use of paste() function to concatenate strings

con_str<-  paste(y[1],y[2],sep = ",")
con_str
```