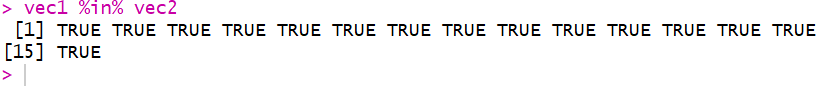
1. Test whether two vectors are exactly equal (element by element).

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

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

ANS.

**vec1 %in% vec2**

****

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

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

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

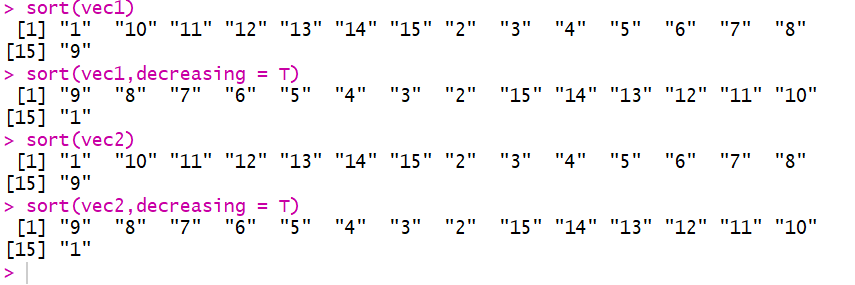
**ANS.**

sort(vec1)

sort(vec1,decreasing = T)

sort(vec2)

sort(vec2,decreasing = T)

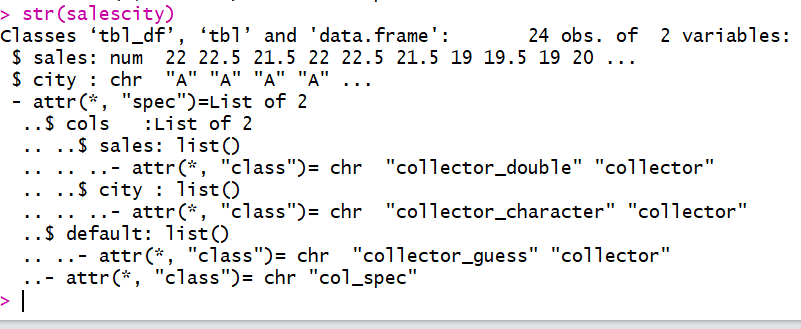


3. What is the major difference between str() and paste() show an example.

ANS.

**str():**

Compactly display the internal structure of an R object, a diagnostic function and an alternative to [summary](http://stat.ethz.ch/R-manual/R-devel/library/base/html/summary.html).



**Paste():**

Function paste is one of the most used function in R. The objective of this function is concatenate a series of strings.

paste("file", "number", "32")

[1] "file number 32"

paste("file", "number", "32", sep = "\_")

[1] "file\_number\_32"

**sep** = The element which separates every term. It should be specified with character string format.

**collapse** = The element which separates every result. It should be specified with character string format and it is optional.

The difference between **paste()**and **paste0()**is that the argument sep by default is ” ” (paste) and “” (paste0).

# Default value of sep with paste function

name\_village <- paste0("Ma", "con", "do")

name\_village

[1] "Macondo"

In conclusion, paste0() is faster than paste() if our objective is concatenate strings without spaces because we don’t have to  specify the **argument sep.**

4. Introduce a separator when concatenating the strings.

ANS. **METHOD 1:**

***temp <- c("Ashwin","Sakharkar")***

***paste(temp,collapse="-")***

******

**METHOD 2:**

***s1<- "Ashwin"***

***s2<-"Sakharkar"***

***paste(s1,s2,sep="-")***

******