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

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

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

identical(vec1, vec2)

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

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

# sort (decreasing order)

sort(vec1)

# sort (increasing order)

sort(vec1, decreasing = TRUE)

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

# sort (decreasing order)

sort(vec2)

# sort (increasing order)

sort(vec2, decreasing = TRUE)

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

We can check the structure of the data with the function str():

# structure of the dataset

str(biomed, vec.len = 1)

## data.frame: 336 obs. of 7 variables:

## $ Publisher : chr "BioMed Central Ltd" ...

## $ Journal.name : chr "AIDS Research and Therapy" ...

## $ Abbreviation : chr "AIDS Res Ther" ...

## $ ISSN : chr "1742-6405" ...

## $ URL : chr "http://www.aidsrestherapy.com" ...

## $ Start.Date : int 2004 2011 ...

## $ Citation.Style: chr "BIOMEDCENTRAL" ...

we can use to create and build strings. paste() takes one or more R objects, converts them to "character", and then it concatenates (pastes) them to form one or several character strings.

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

# paste

PI = paste("The life of", pi)

PI

## [1] "The life of 3.14159265358979"

4. Introduce a separator when concatenating the strings.

# paste

IloveR = paste("I", "love", "R", sep = "-")

IloveR

## [1] "I-love-R"