### 1.Rank function:

x <- c(2,7,1,-17,NA,Inf,35,21)

Use Rank function in R with NAs as last

rank(x,na.last = TRUE)

## [1] 3 4 2 1 8 7 6 5

Use Rank function in R with NAs as first

rank(x,na.last = FALSE)

## [1] 4 5 3 2 1 8 7 6

### 2. What does below function do?

rowSums function adds all the values present in each rows separately

rowSums(iris[,-5])

## [1] 10.2 9.5 9.4 9.4 10.2 11.4 9.7 10.1 8.9 9.6 10.8 10.0 9.3 8.5  
## [15] 11.2 12.0 11.0 10.3 11.5 10.7 10.7 10.7 9.4 10.6 10.3 9.8 10.4 10.4  
## [29] 10.2 9.7 9.7 10.7 10.9 11.3 9.7 9.6 10.5 10.0 8.9 10.2 10.1 8.4  
## [43] 9.1 10.7 11.2 9.5 10.7 9.4 10.7 9.9 16.3 15.6 16.4 13.1 15.4 14.3  
## [57] 15.9 11.6 15.4 13.2 11.5 14.6 13.2 15.1 13.4 15.6 14.6 13.6 14.4 13.1  
## [71] 15.7 14.2 15.2 14.8 14.9 15.4 15.8 16.4 14.9 12.8 12.8 12.6 13.6 15.4  
## [85] 14.4 15.5 16.0 14.3 14.0 13.3 13.7 15.1 13.6 11.6 13.8 14.1 14.1 14.7  
## [99] 11.7 13.9 18.1 15.5 18.1 16.6 17.5 19.3 13.6 18.3 16.8 19.4 16.8 16.3  
## [113] 17.4 15.2 16.1 17.2 16.8 20.4 19.5 14.7 18.1 15.3 19.2 15.7 17.8 18.2  
## [127] 15.6 15.8 16.9 17.6 18.2 20.1 17.0 15.7 15.7 19.1 17.7 16.8 15.6 17.5  
## [141] 17.8 17.4 15.5 18.2 18.2 17.2 15.7 16.7 17.3 15.8

### 3. What does below function do?

colSums function adds all the values present in each columns separately

colSums(iris[,-5])

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## 876.5 458.6 563.7 179.9

### 4. what does below piece of code do?

z = c("may", "the", "rain", "shower")  
substring(z, 2, 3) <- c("@", "#")

* Substring function will extract or replace the substrings in a character vector.
* First 2 letters of all the elements in the character vector z is replaced with the values {c(“@”,“#”)} successively since the start value in substring is given as 2
* letters more than 3 will not be displayed since the last value in substring is given as 3
* So the output of this code will be: “@y”,“#e”,“@i”,“#o”

### 5. what does below function do?

v1 <- c(2,5,6,3,7)  
v2 <- c(15,16,7,3,2,7,5)  
match(v1,v2)

## [1] 5 7 NA 4 3

* Match returns a vector of the positions of first matches of its first argument(v1) in its second (v2).
* If no match is found then NA will be returned. In this case 2 (v1) is found in 5th position of 2nd argument(v2) so 5 will be returned
* Output of this would be 5 7 NA 4 3

### 6.Sort mtcars by mpg column and sort mtcars by carb and then by cyl

data(mtcars)  
mpg1<-mtcars[order(mtcars$mpg),]  
carb1<-mtcars[order(mpg1$carb),]  
finalData <- mtcars[order(carb1$cyl),]  
print(finalData)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2

### 7. Use identical functions and check if below vectors are equal

a <- c(1,2,4,5)  
b <- c(1,5,4,2)  
identical(a,b)

## [1] FALSE

### 8.Create a function that given one word, return the position of word’s letters on letters vector.

fun <- function(x){  
 match(unlist(strsplit(x, split="")), letters)  
}  
fun("acd")

## [1] 1 3 4

### 9.Create a function that given a numeric vector X returns the digits 0 to 9 that are not in X.

num <- function(y){  
 x<-setdiff(0:9,y)  
 return(x)  
}  
num(c(1,3,5,7))

## [1] 0 2 4 6 8 9

### 10. Create a function that given a numeric vector, sort this in ascending order and duplicate it by two.

dup <- function(AS){  
 AS1<-sort(AS,decreasing = FALSE)  
 final<-rep(AS1,each=2)  
 return (final)  
}  
dup(c(3,5,6,2,8,91))

## [1] 2 2 3 3 5 5 6 6 8 8 91 91

### 11. Consider a data frame df:

###Create a function that, given a data frame and two indexes, exchanges two values ??of the Code variable with each other.

Id=c(1:10)  
Age=c(14,12,15,10,23,21,41,56,78,12)  
Sex=c('F','M','M','F','M','F','M','M','F','M')  
Code=letters[1:10]  
df=data.frame(Id,Age,Sex,Code)  
  
swap <- function(x1,x3){  
 df1<-paste(df[x1,'Id'])  
 df[x1,'Id']<-paste(df[x3,'Id'])  
 df[x3,'Id']<-paste(df1)  
 print(df[x1,'Id'])  
 print(df[x3,'Id'])  
}  
swap(1,3)

## [1] "3"  
## [1] "1"

### 12.Consider a vector…Create a function that given a vector string ST return a matrix:

x<-c('NAME:Maria /COUNTRY:uruguay /EMAIL:mariaUY@gmail.com','NAME:Paul/COUNTRY:UK /EMAIL:PaulUK@gmail.com',  
 'NAME:Jhon /COUNTRY:USA /EMAIL:JhonUSA@gmail.com','NAME:Carlos /COUNTRY:Spain /EMAIL:CarlosSP@gmail.com')  
  
mat <- function(ST){  
 ST1<-gsub(":",",",ST)  
 ST2<-gsub("/",",",ST1)  
 ST3<-strsplit(ST2,split = ",",fixed = TRUE)  
 s<-list(unique(unlist(ST3)))  
 Names<-s[[1]][c(1,2,7,10,13)]  
 Country<- s[[1]][c(3,4,8,11,14)]  
 Email <- s[[1]][c(5,6,9,12,15)]  
 m<-matrix(c(Names,Country,Email),nrow = 3,ncol = 5,byrow = TRUE)  
 return(m)  
}  
mat(x)

## [,1] [,2] [,3] [,4]   
## [1,] "NAME" "Maria " "Paul" "Jhon "   
## [2,] "COUNTRY" "uruguay " "UK " "USA "   
## [3,] "EMAIL" "mariaUY@gmail.com" "PaulUK@gmail.com" "JhonUSA@gmail.com"  
## [,5]   
## [1,] "Carlos "   
## [2,] "Spain "   
## [3,] "CarlosSP@gmail.com"