Lab - 4

1 Factors

```
first.name <- c("Priya","Aarav","Darpan","Bandita", "Anika",
                "Chaitanya","Shruti","Chandran", "Darsh","Evanshi")
sex.num <- c(1,0,0,1,1,0,1,0,0,1)
sex.str <- c("female","male","male","female","female","male",
             "female","male","male","female")
first.name[sex.num==0]

## [1] "Aarav"     "Darpan"    "Chaitanya" "Chandran"  "Darsh"

first.name[sex.str=="female"]

## [1] "Priya"   "Bandita" "Anika"   "Shruti"  "Evanshi"

sex.num.fac <- factor(x=sex.num)
sex.num.fac

##  [1] 1 0 0 1 1 0 1 0 0 1
## Levels: 0 1

sex.str.fac <- factor(x=sex.str)
sex.str.fac

##  [1] female male   male   female female male   female male   male   female
## Levels: female male

levels(x=sex.str.fac)

## [1] "female" "male"

mob.str=c("Feb","Jul","Jun","Sep","Dec","Aug","Jul","Mar","Jul","Mar")
mob.str.fac=factor(x=mob.str)
levels(mob.str.fac)

## [1] "Aug" "Dec" "Feb" "Jul" "Jun" "Mar" "Sep"

ms <- c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")

mob.fac <- factor(x=mob.str,levels=ms,ordered=TRUE)
mob.fac

##  [1] Feb Jul Jun Sep Dec Aug Jul Mar Jul Mar
## 12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < ... < Dec

mob.fac[2]<mob.fac[3]

## [1] FALSE
```

```r
Y <- c(0.53,5.4,1.5,3.33,0.45,0.01,2,4.2,1.99,1.01)
br <- c(0,2,4,6)
cut(x=Y,breaks=br)

##  [1] (0,2] (4,6] (0,2] (2,4] (0,2] (0,2] (0,2] (4,6] (0,2] (0,2]
## Levels: (0,2] (2,4] (4,6]

cut(x=Y,breaks=br,right=F)

##  [1] [0,2) [4,6) [0,2) [2,4) [0,2) [0,2) [2,4) [4,6) [0,2) [0,2)
## Levels: [0,2) [2,4) [4,6)

cut(x=Y,breaks=br,right=F,include.lowest=T)

##  [1] [0,2) [4,6] [0,2) [2,4) [0,2) [0,2) [2,4) [4,6] [0,2) [0,2)
## Levels: [0,2) [2,4) [4,6]

lab <- c("Small","Medium","Large")
cut(x=Y,breaks=br,right=F,include.lowest=T,labels=lab)

##  [1] Small  Large  Small  Medium Small  Small  Medium Large  Small  Small
## Levels: Small Medium Large
```

2 List and data frames

Vectors, matrices, and arrays can store only one type of data (Numeric, logical or character). In some cases, in fact, in most cases, we need to store all kinds of data in a tabular form. We will explore two more data structures in this lab session: lists and data frames that allow one to store multiple types of data variables at a time.

2.1 List

```r
my.list <- list(matrix(data=5:8,nrow=2,ncol=2),c(F,T,T,F,T),
                "hi, I am inside a list")
my.list

## [[1]]
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
##
## [[2]]
## [1] FALSE  TRUE  TRUE FALSE  TRUE
##
## [[3]]
## [1] "hi, I am inside a list"

length(x=my.list)
```

```
## [1] 3
```

```
my.list[[3]]
```

```
## [1] "hi, I am inside a list"
```

```
my.list[[3]] <- paste(my.list[[3]],"and you too!")
my.list
```

```
## [[1]]
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
##
## [[2]]
## [1] FALSE  TRUE  TRUE FALSE  TRUE
##
## [[3]]
## [1] "hi, I am inside a list and you too!"
```

```
names(my.list) <- c("my.matrix","my.logical","my.string")
my.list$my.matrix
```

```
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

```
my.list1 <- list(var1=c(T,F,T,T),var2="Do you want to be a part of DRDO project?",
            var3=my.list$my.matrix)
names(my.list1)
```

```
## [1] "var1" "var2" "var3"
```

```
my.list1$inner.list <- list(my.vector=c(T,F,F,T),
                            my.string="Contact/Meet HOD Mathematics")
my.list1
```

```
## $var1
## [1]  TRUE FALSE  TRUE  TRUE
##
## $var2
## [1] "Do you want to be a part of DRDO project?"
##
## $var3
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
##
## $inner.list
## $inner.list$my.vector
## [1]  TRUE FALSE FALSE  TRUE
##
## $inner.list$my.string
## [1] "Contact/Meet HOD Mathematics"
```

```
my.list1$inner.list$my.string

## [1] "Contact/Meet HOD Mathematics"
```

2.2 Data Frames

```
my_data <- data.frame(person=c("Ramesh","Priyanka","Vinodh","Ravi","Palak"),
age=c(22,19,17,19,20),
sex=factor(c("M","F","M","M","F")),
stringsAsFactors=FALSE
)
my_data

##     person age sex
## 1   Ramesh  22   M
## 2 Priyanka  19   F
## 3   Vinodh  17   M
## 4     Ravi  19   M
## 5    Palak  20   F

my_data[1,3]

## [1] M
## Levels: F M

my_data[2:4,1]

## [1] "Priyanka" "Vinodh"   "Ravi"

my_data$age

## [1] 22 19 17 19 20

dim(my_data)

## [1] 5 3

nrow(my_data)

## [1] 5

ncol(my_data)

## [1] 3

my_data$person

## [1] "Ramesh"   "Priyanka" "Vinodh"   "Ravi"     "Palak"
```

```r
newrecord <- data.frame(person="Akansha",age=17,
sex=factor("F",levels=levels(my_data$sex)))
my_data <- rbind(my_data,newrecord)
my_data
```

```
##     person age sex
## 1   Ramesh  22   M
## 2 Priyanka  19   F
## 3   Vinodh  17   M
## 4     Ravi  19   M
## 5    Palak  20   F
## 6  Akansha  17   F
```

```r
stream <- c("CM","CM","ME","ME","EEE","CM")
stream <- factor(x=stream,levels=c("CM","ME","EEE"))
my_data <- cbind(my_data,stream)
my_data
```

```
##     person age sex stream
## 1   Ramesh  22   M     CM
## 2 Priyanka  19   F     CM
## 3   Vinodh  17   M     ME
## 4     Ravi  19   M     ME
## 5    Palak  20   F    EEE
## 6  Akansha  17   F     CM
```

```r
my_data[my_data$stream=="CM",]
```

```
##     person age sex stream
## 1   Ramesh  22   M     CM
## 2 Priyanka  19   F     CM
## 6  Akansha  17   F     CM
```

```r
my_data[my_data$age>18,]
```

```
##     person age sex stream
## 1   Ramesh  22   M     CM
## 2 Priyanka  19   F     CM
## 4     Ravi  19   M     ME
## 5    Palak  20   F    EEE
```

```r
sorted_data <-my_data[order(my_data$person,decreasing = FALSE),]
sorted_data
```

```
##     person age sex stream
## 6  Akansha  17   F     CM
## 5    Palak  20   F    EEE
## 2 Priyanka  19   F     CM
## 1   Ramesh  22   M     CM
## 4     Ravi  19   M     ME
## 3   Vinodh  17   M     ME
```

3 Special values

```
big_num <- 90000^100
big_num

## [1] Inf

-2*Inf

## [1] -Inf

1/Inf

## [1] 0

1+Inf

## [1] Inf

4/0

## [1] Inf

Inf/0

## [1] Inf

Inf+Inf

## [1] Inf

Inf-Inf

## [1] NaN

0/0

## [1] NaN

Inf/Inf

## [1] NaN

is.nan(2+6*(4-4)/0)

## [1] TRUE

spcl <- c(NaN,54.3,-2,NaN,90094.123,-Inf,55)
is.infinite(spcl)

## [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE

1>NaN
```

```
## [1] NA
```

```
a <- c(0,5,NA,9)
a
```

```
## [1]  0  5 NA  9
```

```
a[6]
```

```
## [1] NA
```

```
c(0,5,NULL,9)
```

```
## [1] 0 5 9
```

```
c(NA,NA,NA)
```

```
## [1] NA NA NA
```

```
c(NULL,NULL,NULL)
```

```
## NULL
```

4 is and as dot functions

```
is.numeric(0.1)
```

```
## [1] TRUE
```

```
is.character(0.1)
```

```
## [1] FALSE
```

```
is.character("0.1")
```

```
## [1] TRUE
```

```
as.numeric("0.1")+1
```

```
## [1] 1.1
```

```
is.integer(8.7)
```

```
## [1] FALSE
```

```
as.character(0.1)
```

```
## [1] "0.1"
```

```
as.logical(c("1","0","1","0","0"))
```

```
## [1] NA NA NA NA NA
```

```
as.logical(as.numeric(c("1","0","1","0","0")))
```

```
## [1]  TRUE FALSE  TRUE FALSE FALSE
```