**CME 4403** **AKİF ÇAKAR**

**Lab 2 Worksheet - Due Date: October 14th 23:59**

*Write down answers the space under the questions. Please replace the file name with your name before submitting to the Classroom.*

1. Write a function **inputControl** that takes a character vector and checks how many times name "apple" occurred in this vector. This function returns the total number of occurrences of "apple" string in the input vector. If there is no occurrence, it returns zero (0).

```
inputControl <- function(vctr){
      return (length(which("apple"==vctr)))
}

inputControl(c("orange","apple","banana","apple"))
[1] 2
```

2. Use built-in data set: **iris**

   a) What is the size (dimension) of iris object?

```
dim(iris)

[1] 150   5
```

   b) Find the mean value of the first four columns (Sepal.Length, Sepal.Width , Petal.Length , Petal.Width) by using **apply** function.

```
apply(iris[1:4], 2, mean)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
5.843333     3.057333     3.758000     1.199333
```

   c) Create a new list (**myList)** by using first 4 columns and first 10 rows.

```
myList <- as.list(as.data.frame(iris[1:10,1:4]))
```

```
myList
$Sepal.Length
[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9

$Sepal.Width
[1] 3.5 3.0 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1

$Petal.Length
[1] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5

$Petal.Width
[1] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1
```

d) For **myList** object, compute sum of each list component and return their summation as a new list (**retList)**. What is the Petal.Length value in the retList?

```
retList <- lapply(myList, sum)
retList$Petal.Length

[1] 14.5
```

e) For **myList** object, compute average value of each list component and return their averages as a new vector (**retVec)**. Print values in the retVec to the screen.

```
retVec <- sapply(myList, mean)
print(retVec)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
4.86         3.31         1.45         0.22
```

3. a) Create an input vector (**inp**) containing several names (including "Ali"). Then find the index of "Ali" name in the **inp** vector (Hint: use match() or which() function).

```
inp <- c("Akif","Emre","Ceren","Ali","Deniz")
which(inp=="Ali")

[1] 4
```

b) Create a vector that contains names and NA values. Write a code to filter NA values in this vector and create a new vector without any NA values (Hint: use is.na() or match()).

```
inp <- c("Akif","Emre",NA,"Ceren",NA,"Ali","Deniz")
inp[!is.na(inp)]
```

```
[1] "Akif"  "Emre"  "Ceren" "Ali"   "Deniz"
```

4. Download “Patient-Subtype.csv” and “Protein.txt” files from the course Classroom page.

```
setwd('C:\\Users\\akifc\\Desktop')
options(max.print=999999)            #additional settings for synchronization
```

a) Read data in the “Protein.txt” file into a data.frame (**proteinData**). Get the dimensions and column names of **proteinData**.

```
proteinData <- read.table("Protein.txt", header = TRUE, sep = "\t", dec = ",")
dim(proteinData)
colnames(proteinData)
```

```
[1] 143  26
```

b) Read data in the “Patient-Subtype.csv” file into a data.frame (**patientData**). Get the dimensions and column names of **patientData.**

```
patientData <- read.csv("Patient-Subtype.csv", header = TRUE, sep = ";")
dim(patientData)
colnames(patientData)
```

```
> dim(patientData)
[1] 25  3
> colnames(patientData)
[1] "Patient.ID" "Sub.type"   "Class"
```

c) Select patients in **patientData** if their “Sub.type” is equal to “basal”. Save “Patient.ID” column of these patients in a new vector **basalPatientID.** How many patients did you get?

```
basalPatientID <- as.vector((patientData[(patientData$Sub.type=="basal"),])[,1])
length(basalPatientID)
```

```
[1] 4
```

d) Select all protein data (from **proteinData** object) of patients saved in **basalPatientID** vector. Save these patients' protein data in a new object with the name of **testData.**

```
testData <- proteinData[!is.na(match(colnames(proteinData),basalPatientID))]
```

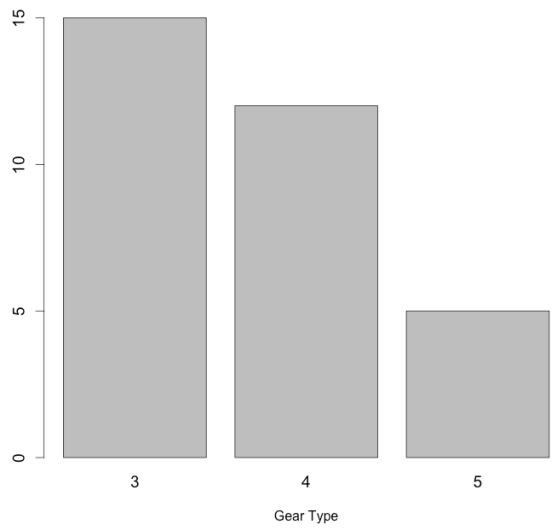
e) What is the mean and standard deviation of each protein sample over all patients saved in **testData** (each row represents one protein; each column represents a patient sample in the testData object).

```
apply(testData, 2, mean)
apply(testData, 2, sd)
```

```
> apply(testData, 2, mean)
TCGA.21.1070 TCGA.21.1081 TCGA.21.5782 TCGA.21.5784
0.2440273    0.0833979    0.4527024    0.2658844
> apply(testData, 2, sd)
TCGA.21.1070 TCGA.21.1081 TCGA.21.5782 TCGA.21.5784
0.6223289    0.7390148    1.0364235    0.9274168
```

5. Use built-in data set: **mtcars**

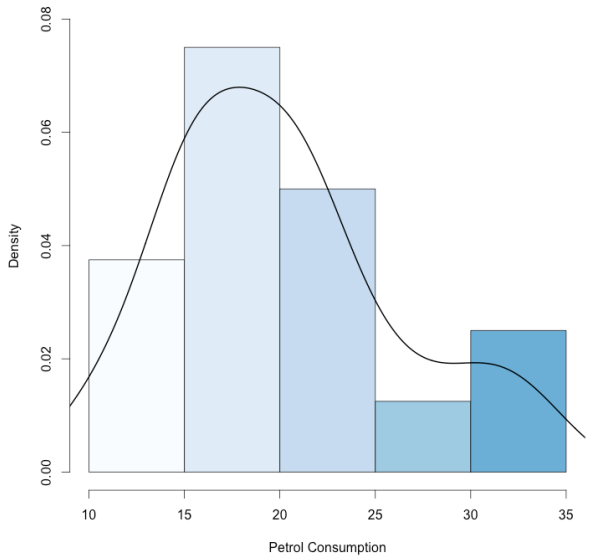
a) Compute distribution of gears (gear) with a barplot as shown below.




```
barplot(table(mtcars$gear), xlab="Gear Type")
```

b) Draw a histogram plot to show the petrol consumption (mpg) of the cars. Then draw their density curve on the same plot (it will look like in the figure below).

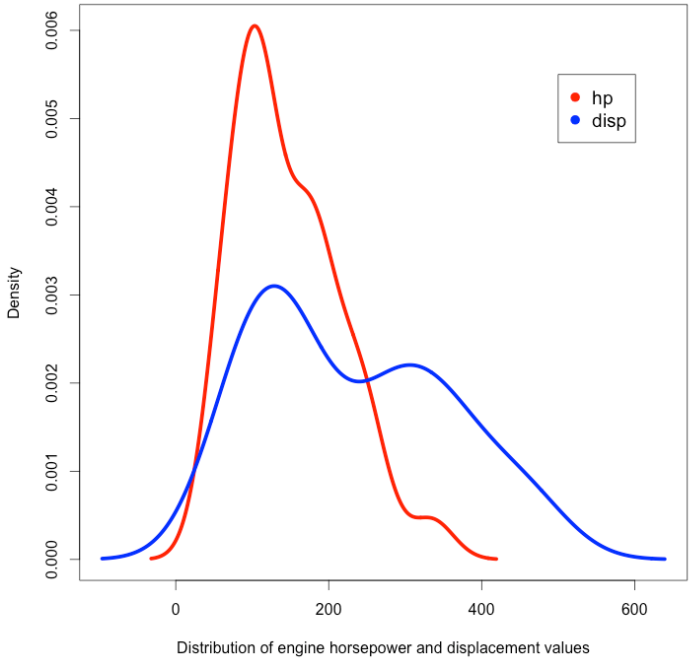
What kind of distribution do you observe in this plot?




```
hist(mtcars$mpg, col="lightblue", freq=FALSE, xlab="Petrol Consumption")
lines(density(mtcars$mpg),col="black")
```

It is a kind of **Unimodal (Single Peaked) distribution** because that has one clear peak.

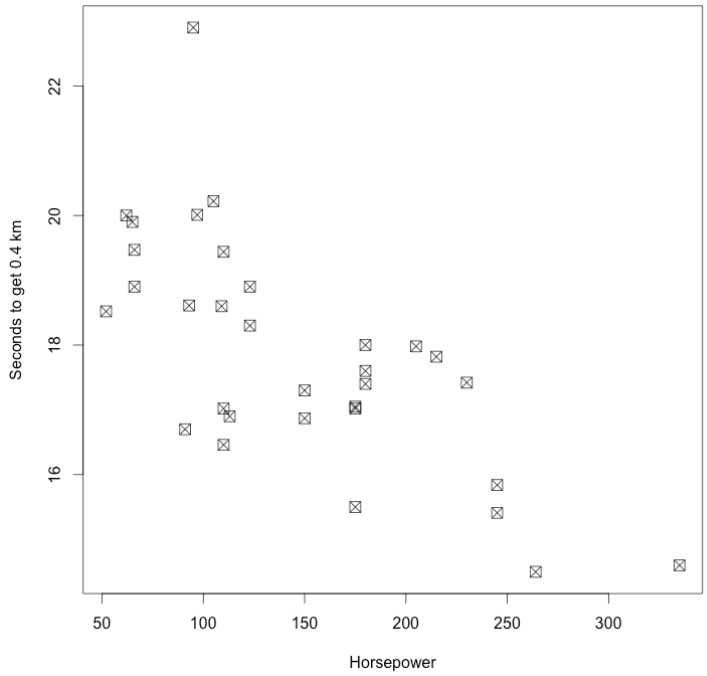
c) Draw density lines for a comparison of cars' engine displacement (disp) and horsepower (hp) on the same plot.



```
d1 <- density(mtcars$disp)
d2 <- density(mtcars$hp)
plot(range(d1$x, d2$x), range(d1$y, d2$y), type = "n",
                xlab = "Distribution of engine horsepower and displacement values",
                ylab = "Density")
lines(d1, col = "blue" , lwd=4)
lines(d2, col = "red" , lwd=4)

legend(x=200, y=0.006, c("hp","disp"), cex=0.5, col=c("red","blue"), lty=1:1)
```

d) Draw a scatterplot to show the relationship between horsepower (hp) and the time a car needs to take 0.4 km (qsec).




```
plot(mtcars$hp,mtcars$qsec,pch = 7,xlab="Horsepower",ylab="Seconds to get 0.4 km")
```