1. **The following table gives the size of the floor area (ha) and the price ($A000), for 15 houses sold in the Canberra (Australia) suburb of Aranda in 1999. Type these data into a data frame with column names area and sale.price.**

	area	sale.price
1.	694	192.0
2.	905	215.0
3.	802	215.0
4.	1366	274.0
5.	716	112.7
6.	963	185.0
7.	821	212.0
8.	714	220.0
9.	1018	276.0
10.	887	260.0
11.	790	221.5
12.	696	255.0
13.	771	260.0
14.	1006	293.0
15.	1191	375.0

(a) Plot sale.price versus area.

```
area <- c(694, 905, 802, 1366, 716, 963, 821, 714, 1018, 887, 790, 696, 771, 1006, 1191)
sale.price ← c(192.0, 215.0,  215.0, 274.0, 112.7, 185.0, 212.0, 220.0, 276.0, 260.0, 221.5, 255.0, 260.0, 293.0, 375.0)
house <- data.frame(area, sale.price, stringsAsFactors = FALSE)
plot(sale.price~area, data=house)
```

(b) Use the hist() command to plot a histogram of the sale prices.

```
hist(house$sale.price, main="house prices", xlab="price", col="blue")
```

(c) Repeat (a) and (b) after taking logarithms of sale prices.

```
house[2]<- log(house[2])
plot(sale.price~area, data=house)
hist(house$sale.price)
```

(d) The two histograms emphasize different parts of the range of sale prices. Describe the differences.

#Your Answer

2. **Create a list of vectors of varying length:**
   a) **Create a vector "vector1" of 100 random numbers from 10 to 20.**

```
vector1 <- runif(100,10,20)
vector1
```

   b) **Create a list with 100 vectors containing random numbers from uniform distribution of size given by "vector1".**

```
l <- lapply(1:100, function(x) runif(length(vector1)))
```

   c) **Use a for loop to find the lengths of the vectors in the list. First make a vector for storing the lengths.**

```
len <- vector()
for(i in 1:100){
  len[i] <- length(l[[i]])
}
```

   d) **Repeat c) using sapply**

```
len_vec <- sapply(1:100,function(x) length(l[[i]]))
```

   e) **Repeat c) using lapply**

```
Alen_list <- lapply(1:100,function(x) length(l[[i]]))
```

3. **Examine the built in ChickWeight data (the help gives background about the data).**
   (a) Construct a plot of weight against time for chick number 34.

```
data <- ChickWeight
chick34 <- data[data$Chick==34,]
plot(chick34$weight~chick34$Time,
    xlab="Time",ylab="Weight",
    main="Weight vs Time",
    type="l",col="red")
```

   (b) For chicks in diet group 4, display box plots for each time point.

```
diet4 <- data[data$Diet==4,]
boxplot(diet4$weight~diet4$Time,
      xlab="Weight",ylab="Time",
      main="Distribution of Wt vs Time")
```

(c) Compute the mean weight for chicks in group 4, for each time point. Plot this mean value against time.

```
timesteps <- unique(diet4$Time)
mean_diet4 <- sapply(timesteps, function(x) mean(diet4$weight[diet4$Time==x]))
plot(mean_diet4~timesteps,
    xlab="Time",ylab="Mean Wt",
    main="Mean Weight vs Time",
    type="l",col="red")
```

(d) Repeat the previous computation for group 2. Add the mean for group 2 to the existingplot.

```
diet2 <- data[data$Diet==2,]
mean_diet2 <- sapply(timesteps, function(x) mean(diet2$weight[diet2$Time==x]))
lines(mean_diet2~timesteps,col="blue")
```

(e) Add a legend and a title.

```
legend("topleft",lty=1, c("Diet-4","Diet-2") ,col=c("red","blue"))
```

4. **From the library MASS, use "cats" data and perform the following.**
   a) Extract Male cats data set separately.

```
library('MASS')
male <- cats[cats$Sex=='M',]
```

b) Display a scatterplot for male cats. Interpret the dependent and independent variables of the data.

```
plot(Hwt~Bwt,data=cats,
    xlab="Heart Weight",
    ylab="Birth Weight",
    main="Plot of Heart Wt vs Birth Wt")
```

c) Write a function to implement simple linear regression and fit a linear regression model for male cats. Add fitted regression line to scatterplot of male cats data.

```
model <- lm(Hwt ~ Bwt, data=cats)
abline((model),col="red")
```

5. **Consider the data of Pneumoconiosis among coalface workers as shown in table and perform the following to examine the relationship between exposure time (years) and risk of disease. Store the data in coalworker1.csv**

Exposure time	Normal	Diseased
5.8	98	0
15	51	3
21.5	34	9
27.5	35	13
33.5	32	19
39.5	23	15
46	12	16
51.5	4	7

a) Create a data frame with three columns "exposure time", "normal" and "diseased".

```
Exposure <- c(5.8,15,21.5,27.5,33.5,39.5,46,51.5)
Normal <- c(98,51,34,35,32,23,12,4)
Diseased <- c(0,3,9,13,19,15,16,7)
data <- data.frame(Exposure, Normal, Diseased)
write.csv(data, '/home/ashrafalis/Downloads/data.csv', row.names = FALSE)
data1 <- read.csv('/home/ashrafalis/Downloads/data.csv')
```

b) Plot the data to get an overview of data. Interpret the dependent and independent variables of the data.

```
total <- data$Normal + data$Dieased
risk <- data$Diseased / total
plot(data$Exposure,risk)
```

c) Fit the linear regression model for the data. Does the model seem to fit the data reasonably well?

```
fit <- lm(risk~data$Exposure)
abline(fit)
print ("Risk doubles on doubling exposure time (Linear MOdel)")
```

d) Predict the danger values if the exposure time doubled.

```
danger_val = -fit$coefficients[1]/fit$coefficients[2] # solve for y = 0
print (paste("Danger Value : ", danger_val))
```

6. **Data from 20 chemical dissolutions are collected to analyze the association between the toxicity of dissolution on one side and 3 explanatory variables on the other side. Store the data in lser.csv with variables**
   tox: toxicity of dissolution
   base:ability to accept hydrogen ions
   acid: ability to liberate hydrogen ions
   colour: ability to change colour
   Create dataset called lser with data. Plot to get overview of dataset. Fit a linear regression models with
   tox as response to base,acid and colour as explanatory variables(tox vs base, tox vs acid, tox vs colour). Give the interpretation of the parameter estimates. Are all explanatory variables significant? Measure expected toxicity for a specific solvent with specific base.

7. **Create dataset in R to store students details(NAME,Gender,Gatescore,college,City). Gendershould be entered as female or male. Write and execute commands for the following using the above dataset created:**

```
name <- c("ajay","ABC","qwerty","hello","pkjh","lkg")
gender <- c("m","f","f","m","f","m")
score <- c(29,45,12,32,60,60)
college <- c("msr","rk","rk","sit","hbit","hbit")
city <- c("blore","mlore","blore","blore","mlore","mlore")
stu.data <- data.frame(name,gender,score,college,city)
```

a. Plot the proportions of female and male students performance

```
m <- stu.data[which(stu.data$gender == "m"),]$score
f <- stu.data[which(stu.data$gender == "f"),]$score
hist(m,col="red")
hist(f,col="blue", add=T)
```

b. Using ANOVA, check Is there any significant association between Gate score and college where the student studied?

```
avo.way <- aov(score~college, data=stu.data)
avo.way
```

c. Plot female students performance and male students performance using line graph

```
plot(f, type = "o", col = "red", xlab = "Feq", ylab = "Score", main = "Performance")
lines(m, type = "o", col = "blue")
```

8. **Create dataset in R to store students details(NAME,Gender,Gatescore,college,City). Gender should be entered as female or male. Write and execute commands for the following using the above dataset created:**

```
Name <- c("Amrutha", "Architha", "Ashraf", "Bhagya", "Chaitra", "Chandana",
"Chetana", "Dhanush")
Gender <- as.factor(c("female", "female", "male", "female", "female", "female",
"female", "male"))
Gatescore <- c(95, 72, 72, 90, 79, 96, 63, 86)
College <- c("msrit", "rv", "pes", "msrit", "rv", "pes", "msrit", "rv")
City <- c("aaa", "aab", "aac", "aaa", "aab", "aac", "aaa", "aab")
Student <- data.frame(Name, Gender, Gatescore, College, City)
```

a. Add new column which mention the average Gate score of each college. Attach that to each student who belongs to that college.

```
Avg <- with(Student, ave(Gatescore, College, FUN = function(x) mean(x)))
Student <- data.frame(Student, Avg)
```

b. Display the top scorers in each college

```
#Top Score in Each College
Agg <- aggregate(Gatescore~College, Student, max)
a <- (match(Agg$Gatescore, Student$Gatescore))
Student$Name[a]
Student$College[a]
```

c. Plot slopes for each college Gate performance

```
plot(Avg~College, data=Student)
```

9. **Load the in-built dataset mtcars() and perform the following.**
   a) Dot plot of mpg for each car model

```
dotchart(mtcars$mpg , labels = row.names(mtcars),
      main = "Mileage for Car Models", cex = 0.6,
      xlab = "Miles per Gallon")
```

b) Create a colored histogram of 12 bins with x-axis as “Miles per gallon“ and y-axis as “frequency“.

```
hist(mtcars$mpg, col = "blue", breaks = 12,
    xlab = "Miles per Gallon", xlim = c(10,40),
    ylab = "Frequency")
```

c) Create kernel density plots of mpg by number of cylinders with legends as 4 cylinders, 6 cylinders and 8 cylinders. Interpret the results obtained in (a) & (b).

```
cyl4 <- mtcars$mpg[mtcars$cyl==4]
cyl6 <- mtcars$mpg[mtcars$cyl==6]
cyl8 <- mtcars$mpg[mtcars$cyl==8]
d1 <- density(cyl4)
d2 <- density(cyl6)
d3 <- density(cyl8)
plot(d1,col="red",ylim=c(0,0.3))
lines(d2,col="green")
lines(d3,col="blue")
legend("topright",c("4","6","8"),lty = 1,col=c("red","green","blue"))
```

d) Generate a box plot of car mileage versus transmission type and number of cylinders.

```
boxplot(mpg ~ am + cyl,data=mtcars)
```

10. From the library MASS, use birthwt data and perform the following.

a) Use with() along with the tapply() function to produce a table showing the % of babies born weighing under 2500g within each combination of mother‟s race and smoking status.

b) Use the tapply() function to produce a table showing the proportion of babies born weighing under 2500g, broken down by race, smoking status, and hypertension.

c) Repeat part (b) using the aggregate() function.

11. **Implement k- nearest neighbor algorithm in R. Use the data below to find the *k*-nearest neighbor for record #10, using *k* = 3.**

Record	Age	Marital	Income	Risk
1	22	Single	$46,156.98	Bad loss
2	33	Married	$24,188.10	Bad loss
3	28	Other	$28,787.34	Bad loss
4	51	Other	$23,886.72	Bad loss
5	25	Single	$47,281.44	Bad loss
6	39	Single	$33,994.90	Good risk
7	54	Single	$28,716.50	Good risk
8	55	Married	$49,186.75	Good risk
9	50	Married	$46,726.50	Good risk
10	66	Married	$36,120.34	Good risk

```
Record <- c(1,2,3,4,5,6,7,8,9,10)
Age <- c(22,33,28,51,25,39,54,55,50,66)
Marital <- as.factor(c("Single", "Married", "Other", "Other", "Single", "Single", "Single", "Married", "Married", "Married"))
Income <- c(46156.98, 24188.10, 28787.34, 23886.72, 47281.44, 33994.90, 28716.50, 49186.75, 46726.50, 36120.34)
Risk <- c("Bad Loss", "Bad Loss", "Bad Loss", "Bad Loss", "Bad Loss", "Good Risk", "Good Risk", "Good Risk", "Good Risk", "Good Risk")
Data <- data.frame(Record, Age, Marital, Income, Risk)

#75% data to train
random <- sample(1:nrow(Data), 0.75 * nrow(Data))
nomalization <-function(x) { (x -min(x))/(max(x)-min(x))   }
norm <- as.data.frame(lapply(Data[,c(1,2,4)], nomalization))
summary(norm)
train <- norm[random,]
test <- norm[-random,]
target_category <- Data[random,5]
test_category <- Data[-random,5]
library(class)
pr <- knn(train,test,cl=target_category,k=3)
tab <- table(pr,test_category)
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)
k=3
km = kmeans(Data[,c(1,2,4)], k, iter.max=10, algorithm=c("Forgy"))
dist = dist(Data[,c(1,2,4)])
mds = cmdscale(dist)
plot(mds, col=km$cluster, pch = 20, cex = 3)
km1 = kmeans(mds, k, iter.max=1000, algorithm=c("Forgy"))
points(km1$centers, col = 10, pch = 8, cex = 2)
```

12. **Implement the Naïve Baye's algorithm in R.**

```
data(iris)
#if u get error uncomment this
#colnames(iris)[1:5]=c("sepal_length","sepal_width","petal_length","petal_width","class")
#str(iris)
iris$class=factor(iris$class)
table(iris$class)
sample_iris=sample(150,110,replace = FALSE)
iris_training=iris[sample_iris,]
iris_test=iris[-sample_iris,]
iris_training_labels=iris[sample_iris,]$class
iris_test_labels=iris[-sample_iris,]$class
table(iris_training$class)
table(iris_test$class)
iris_classifier=naiveBayes(iris_training,iris_training_labels)
#performance
iris_test_pred=predict(iris_classifier,iris_test)
iris_test_pred
```

13. **Implement k-means clustering algorithm in R. Use a suitable dataset (iris dataset) for demonstrating the algorithm.**

```
library(ggplot2)
data(iris)
```

```
k = 3
km = kmeans(iris[1:4], k, iter.max=1000, algorithm=c("Forgy"))
dist = dist(iris[1:4])
mds = cmdscale(dist)
palette(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3", "#FF7F00", "#FFFF33",
"#A65628", "#F781BF", "#999999", "#000000"))
plot(mds, col=km$cluster, pch = 20, cex = 3)
# plot centroids
km1 = kmeans(mds, k, iter.max=1000, algorithm=c("Forgy"))
points(km1$centers, col = 10, pch = 8, cex = 2)
```

14. Load the in-built dataset mtcars() and perform the following.

a) Dot plot of mpg for each car model

```
library(ggplot2)
data(mtcars)
dotchart(mtcars$mpg,labels=row.names(mtcars),cex=.7)
```

b) Create a colored histogram of 12 bins with x-axis as “Miles per gallon“ and y-axis as “frequency“.

```
hist(mtcars$mpg, xlab="Miles per gallon", breaks=12, col = c("blue", "red", "gray",
"green"))
```

c) Generate a box plot of car mileage versus transmission type and number of cylinders

```
boxplot(mpg~am, data = mtcars, col = c("blue", "blue"), xlab = "Transmission", ylab =
"Miles per Gallon", main = "MPG by Transmission Type")
boxplot(mpg~cyl, data = mtcars, col = c("red"), xlab = "Cylinders", ylab = "Miles per
Gallon", main = "MPG by Number of Cylinders")
```

d) Fit a regression model for car mileage versus transmission type and number of cylinders.

```
fit <- lm(mpg~am, data = mtcars)
summary(fit)

fit <- lm(mpg~cyl, data = mtcars)
summary(fit)
```