```
#Question 1:
# Consider the following weights 60, 72, 34, 56, 87, 80, 89, 55, 93,
28, 48, 59. Use the R script to finish the following questions

#assign data
> weight<-c( 60,72,34,56,87,80,89,55,93,28,48,59)
 [1] 60 72 34 56 87 80 55 93 28 48 59
#mean of weight
> mean(weight)
 [1] 63.41667
#standard deviation of weight
> sd(weight)
 [1] 21.21088
#length of weight
> length(weight)
 [1] 12
#length of weight which is larger than 55
> length(weight[weight>55])
 [1] 8
# compare each element if it is larger than 55 & less then 85
> weight>55&weight<85
 [1]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
TRUE


#Question 2:
# Use the following script, we can generate a 3X4 matrix tmp<-
matrix(rnorm(12), 3, 4).

# create a matrix
> tmp<-matrix(rnorm(12),3,4)
> tmp
           [,1]       [,2]        [,3]       [,4]
[1,] 0.61210923 -1.1160224 1.835798330 -0.1216463
[2,] 0.14520974 -2.0374560 1.756647663  0.3807652
[3,] 0.04482355 -0.2772726 0.009757602 -1.9218685

# find mean of each row
>  apply(tmp,1,mean)
[1]  0.30255971  0.06129165 -0.53613998
> mean(tmp[1,])
[1] 0.3025597
> mean(tmp[2,])
[1] 0.06129165
> mean(tmp[3,])
[1] -0.53614

#find dimension of the matrix
> dim(tmp)
[1] 3 4
```

```
# Use ‘cat’ function to output elements in the first row that are
larger than 0.5
> x<-tmp[1,]
> cat(x[x>0.5],"\n")
0.6121092 1.835798


#Question 3
#Write the logical expression to extract blood.glucose greater than 10
and short.velocity greater than 1.5 in the thuesen data set.
> install.packages("ISwR")
> library(ISwR)
> data(thuesen)
> thuesen[(thuesen$blood.glucose>10)&(thuesen$short.velocity>1.5),]
   blood.glucose short.velocity
1           15.3           1.76
13          19.0           1.95


#Question 4
#Generate 10 random integers that are uniformly distributed between 1
and 50 (1 and 50 included).
> sample(1:50,10,replace=T)
 [1] 35  1 37  4  4 11 43 47  8 34
> sample.int(50,10,replace=T)
 [1] 41 18 30 45 27  9  4 22 26 25


#Question 5
# (1) Use ‘sample’ function to generate a random vector that follows a
multinomial distribution with probability (0.2, 0.3, 0.5).
>sample(c(1,2,3),10,replace=T,prob=c(0.2,0.3,0.5))
 [1] 1 3 3 3 2 2 2 3 2 2

#(2) Without use the ‘sample’ function, generate a random vector that
follows a multinomial distribution with probability (0.2, 0.3, 0.5).
>x<-rep(0,100)
for(i in 1:100)
{
m<-runif(1,0,1)
{if(m<=0.2) {x[i]="low"} else if (m>0.5) {x[i]="high"} else
{x[i]="middle"}}
}
# it is a function generate run a value between 0 to 1 and  is
assigned to different categories.  X should be different each time!


#Question 6
#(a) a normally distributed variable with mean 36 and standard
```

```
deviation 6 is larger than 43
> 1-pnorm(43,36,6)
[1] 0.1216725

#(b) X>6.7 in a chi-square distribution with 3 degrees of freedom
> 1-pchisq(6.7,3)
[1] 0.08210006

#(c) getting 10 out of 10 successes in a binomial distribution with
probability 0.8
> dbinom(10,10,0.8)
[1] 0.1073742


#Question 7
#Construct the following table that summarizes the number of people
who have car accidents in a school.
> car_accident<-matrix(c(30,15,17,4,123,98,139,60),nrow=4)
> colnames(car_accident)<-c("Accidents","No")
> rownames(car_accident)<-c("18-20","21-23","24-25",">25")
> names(dimnames(car_accident))<-c("Age","")
> car_accident

Age     Accidents  No
  18-20        30 123
  21-23        15  98
  24-25        17 139
  >25           4  60


#Question 8
#Generate 100 exponentially distributed random variables with rate 2,
and plot their empirical distribution function.
> x<-rexp(100,rate=2)
> n=100;
> plot(sort(x),(1:n)/n,type="s",ylim=c(0,1))


#Question 9
#(a) Plot a histogram for the “react” data set in the ISwR package.
> library(ISwR)
> data(react)
> hist(react)

#(b) Try “truehist” function from the MASS package as a replacement of
“hist” function.
> #install.packages("MASS")
> library(MASS)
> truehist(react)
```