Practical 1

```
#1.Create Vector
fruits<-c("apple","banana","mango","pineapple");
print(fruits)
#class of vector
print(class(fruits))
num<-c(1,4,5,7,8,9);
print(num)
#class of vetor
print(class(num))

#2.List
list1<-list(c(1:3),fruits)
print(list1)

#3.Matrices
M=matrix(c(1:9),nrow=3,byrow=TRUE)
print(M)
K=matrix(c(1:12),nrow=4,ncol=3,byrow=TRUE)
print(K)

#4.Array
A<-array(c("yes","no","true","false"),dim=c(4))
print(A)
B<-array(c("yes","no","true","false"),dim=c(3,4))
print(B)
C<-array(c("yes","no","true","false"),dim=c(3,4,2))
print(C)

#5.Factor
apple_colors<-c("red","blue","green")
factor_apple<-factor(apple_colors)
#convert in sequence order
print(factor_apple)
print(nlevels(factor_apple))

#6.Data frames
my<-data.frame(
  name=c("veer","nilesh","pooja","habat"),
  gender=c("male","male","female","male"),
  height=c(6.8,6.7,6,5.7),
  weight=c(75,80,50,64),
  age=c(21,19,18,20)
)
print(my)
data.frame(my$name,my$gender)
```

Output:-

```
[1] "apple"     "banana"
"mango"     "pineapple"
[1] "character"
[1] 1 4 5 7 8 9
[1] "numeric"
[[1]]
[1] 1 2 3

[[2]]
[1] "apple"     "banana"
"mango"     "pineapple"

     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
[1] "yes"   "no"    "true"
"false"
     [,1]   [,2]    [,3]    [,4]
[1,] "yes"  "false" "true"  "no"
[2,] "no"   "yes"   "false" "true"
[3,] "true" "no"    "yes"
"false"
, , 1
```

```
     [,1]   [,2]    [,3]    [,4]
[1,] "yes"  "false" "true"  "no"
[2,] "no"   "yes"   "false" "true"
[3,] "true" "no"    "yes"
"false"

, , 2

     [,1]   [,2]    [,3]    [,4]
[1,] "yes"  "false" "true"  "no"
[2,] "no"   "yes"   "false" "true"
[3,] "true" "no"    "yes"
"false"

[1] red   blue  green
Levels: blue green red
[1] 3
    name gender height weight age
1   veer   male    6.8     75  21
2 nilesh   male    6.7     80  19
3  pooja female    6.0     50  18
4  habat   male    5.7     64  20
  my.name my.gender
1    veer      male
2  nilesh      male
3   pooja    female
4   habat      male

[Program exited with exit code 0]
```

Practical 3

#1.find mean of 12,7,3,4,2,1,18,2,54,-21,8,-5
x<-c(12,7,3,4,2,1,18,2,54,-21,8,-5)
result.mean<-mean(x)

```
print(result.mean)

#2.find median of 12,7,3,4,2,1,18,2,54,-21,8,-5
x<-c(12,7,3,4,2,1,18,2,54,-21,8,-5)
result.median<-median(x)
print(result.median)

#3.find mode of 2,1,2,3,12,3,4,1,5,5,3,2,2
getmode=function(v)
{
  uniqv=unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
}
v<-c(2,1,2,3,12,3,4,1,5,5,3,2,2)
result.mode<-getmode(v)
print(result.mode)

#find range of the eruption duration in the data set faithful
duration=faithful$eruption
min(duration)
max(duration)
result.range=max(duration)-min(duration)
print(result.range)

#find standard deviation of the eruption duration in the data set faithful
duration=faithful$eruption
result.sd=sd(duration)
print(result.sd)

#find variance of the eruption duration in the data set faithful
duration=faithful$eruption
result.var=var(duration)
print(result.var)

#find summary of the data set faithful
result.summary=summary(faithful)
print(result.summary)
```

Output:-

```
[1] 7.083333
[1] 3.5
[1] 2
[1] 1.6
[1] 5.1
[1] 3.5
[1] 1.141371
[1] 1.302728
   eruptions        waiting
 Min.   :1.600   Min.   :43.0
 1st Qu.:2.163   1st Qu.:58.0
 Median :4.000   Median :76.0
 Mean   :3.488   Mean   :70.9
 3rd Qu.:4.454   3rd Qu.:82.0
 Max.   :5.100   Max.   :96.0

[Program exited with exit code 0]
```

Practical No: 4

```
salarydata<-data.frame(
salaries_low=c(25000,26000,27000,28000,29000,30000,31000),
salaries_high=c(25999,26999,27999,28999,29999,30999,32999),
Numbers=c(8,10,16,14,10,5,2)
)
print(salarydata)
#salarydata<-salarydata[,cumNumbers:=cumsum(Numbers)]
salarydata

mediangroup<-salarydata
{
  ((cumNumbers-Numbers)<=(max(cumNumbers)/2 &
(cumNumbers)>=(max(cumNumbers)/2))
   {
     l1=mediangroup(,salaries_low)
     l2=mediangroup(,salaries_high)
     f=mediangroup(,Numbers)
     pcf=mediangroup(,cumNumbers-Numbers)
     n=salarydata(,sum(Numbers))
   }
}
median<-l1+(((n/2)-pcf)/f*(l2-l1))

median
```

Output:-

```
    .N

> input<-read.csv(file.choose())
> print(input)
  salaries_low salaries_high Numbers
1        25000         25999       8
2        26000         26999      10
3        27000         27999      16
4        28000         28999      14
5        29000         29999      10
6        30000         30999       5
7        31000         31999       2
> s<-sum(input$Numbers)
> print(s)
[1] 65
> mg<-s/2
> i<-1
> med<-data.table(input)
> med<-med[,cumNumbers := cumsum(Numbers)]
> print(med)
> m<-max(input$Numbers)
> mediangroup<-med[
+   (cumNumbers-Numbers)<=(max(cumNumbers)/2) &
+     cumNumbers>=(max(cumNumbers)/2)]
> l1 = mediangroup[,salaries_low]
> l2 = mediangroup[,salaries_high]
> f = mediangroup[,Numbers]
> pcf = mediangroup[,cumNumbers- Numbers]
> n = med[,sum(Numbers)]
> median <- l1 + (((n/2)-pcf)/f*(l2-l1))
> median
[1] 27905.34
> 
```

Practical No: 5

salarydata<-data.frame(

```
salaries_low=c(25000,26000,27000,28000,29000,30000,31000),
salaries_high=c(25999,26999,27999,28999,29999,30999,32999),
Numbers=c(8,10,16,14,10,5,2)
)
print(salarydata)
salarydata[ , prevNumbers := shift(Numbers,1)]

salarydata[ , nextNumbers := shift(Numbers,-1)]

salarydata

#identifying mode group
modegroup <- salarydata[Numbers == max(Numbers)]
modegroup

#creating the variables needed to calculate mode
l1 = modegroup[,salaries_low]
l2 = modegroup[,salaries_high]
f1 = modegroup[,Numbers]
f0 = modegroup[,prevNumbers]
f2 = modegroup[,nextNumbers]

#calculating mode
groupmode <- l1 + ((f1-f0)/(f1-f0+f1-f2)*(l2-l1))
groupmode
```

Output:-

OUTPUT:

```
~/
> l1<-group[,salaries_low]
> l2<-group[,salaries_high]
> f1<-group[,Numbers]
> f0<-group[,preNumbers]
> f2<-group[,nextNumbers]
> group
   salaries_low salaries_high Numbers preNumbers nextNumbers
1:        27000         27999      16         10          14
> data
   salaries_low salaries_high Numbers preNumbers nextNumbers
1:        25000         25999       8         NA          10
2:        26000         26999      10          8          16
3:        27000         27999      16         10          14
4:        28000         28999      14         16          10
5:        29000         29999      10         14           5
6:        30000         30999       5         10           2
7:        31000         31999       2          5          NA
> l1
[1] 27000
> l2
[1] 27999
> f1
[1] 16
> f0
[1] 10
> f2
[1] 14
> MODE<-l1+((f1-f0)/(f1-f0+f1-f2))*(l2-l1)
> print(MODE)
[1] 27749.25
> 
```

Practical No: 6

```
# a) 12, 6, 7, 3, 15, 10, 18, 5
# b) 9, 3, 8, 8, 9, 8, 9, 18.

input<-data.frame(
  X1=c(12, 6, 7, 3, 15, 10, 18, 5),
  X2=c(9, 3, 8, 8, 9, 8, 9, 18)
)
print("Standard Deviaton of X1 and X2 are : ")
```

```
sd(input$X1)
sd(input$X2)

print("Variance of X1 and X2 are : ")
var(input$X1)
var(input$X2)
```

OUTPUT:-

```
[1] "Standard Deviaton of X1 and
X2 are : "
[1] 5.209881
[1] 4.140393
[1] "Variance of X1 and X2 are : "
[1] 27.14286
[1] 17.14286

[Program exited with exit code 0]
```

Practical No: 7

```
input<-data.frame(
  X=c(0,1,2,3,4),
  F1=c(10,5,2,2,1),
  F2=c(1,2,14,2,1),
  F3=c(1,2,5,5,10)
)
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

mean1<-mean(rep(input$X,input$F1))
median1<-median(rep(input$X,input$F1))
mode1<-getmode(rep(input$X,input$F1))
sd1<-sd(rep(input$X,input$F1))
Pearsonfirst<-((mean1-mode1)/sd1)
Pearsonsecond<-(3*(mean1-median1)/sd1)
print("Pearson's first coefficients of skewness with x and f1 is : ")
print(Pearsonfirst)
print("Pearson's second coefficients of skewness with x and f1 is : ")
print(Pearsonsecond)

mean1<-mean(rep(input$X,input$F2))
median1<-median(rep(input$X,input$F2))
```

```
mode1<-getmode(rep(input$X,input$F2))
sd1<-sd(rep(input$X,input$F2))
Pearsonfirst<-((mean1-mode1)/sd1)
Pearsonsecond<-(3*(mean1-median1)/sd1)
print("Pearson's first coefficients of skewness with x and f2 is : ")
print(Pearsonfirst)
print("Pearson's second coefficients of skewness with x and f2 is : ")
print(Pearsonsecond)

mean1<-mean(rep(input$X,input$F3))
median1<-median(rep(input$X,input$F3))
mode1<-getmode(rep(input$X,input$F3))
sd1<-sd(rep(input$X,input$F3))
Pearsonfirst<-((mean1-mode1)/sd1)
Pearsonsecond<-(3*(mean1-median1)/sd1)
print("Pearson's first coefficients of skewness with x and f3 is : ")
print(Pearsonfirst)
print("Pearson's second coefficients of skewness with x and f3 is : ")
print(Pearsonsecond)
```

OUTPUT:-

```
[1] "Pearson's first coefficients
of skewness with x and f1 is : "
[1] 0.7696196
[1] "Pearson's second coefficients
of skewness with x and f1 is : "
[1] 1.09367
[1] "Pearson's first coefficients
of skewness with x and f2 is : "
[1] 0
[1] "Pearson's second coefficients
of skewness with x and f2 is : "
[1] 0
[1] "Pearson's first coefficients
of skewness with x and f3 is : "
[1] -0.9037158
[1] "Pearson's second coefficients
of skewness with x and f3 is : "
[1] -0.2168918

[Program exited with exit code 0]
```

Practical No: 8

```
Me<-data.frame(
  Spades=402, Diamonds=358,Clubs=273, Hearts=467
)
attach(Me)
result<-chisq.test(Me)
print(result)
if(result$p.value>=0.05){
  print("Null Hypothesis is Accepted")
```

```
}else{
  print("Null Hypothesis is Rejected")
}
```

OUTPUT:-

```
        Chi-squared test for given
probabilities

data:  Me
X-squared = 53.029, df = 3, p-
value = 1.807e-11

[1] "Null Hypothesis is Rejected"

[Program exited with exit code 0]
```

Practical No: 9

```
input<-data.frame(
   Finance = c(12,7),
   Sales = c(38,19),
   HR= c(5,3),Technology= c(8,1))
attach(input)
result<-chisq.test(input)
print(result)
if(result$p.value>=0.05){
  print("Null Hypothesis is Accepted")
}else{
  print("Null Hypothesis is Rejected")
}
```

OUTPUT:-

```
Warning message:
In chisq.test(input) : Chi-squared
approximation may be incorrect

        Pearson's Chi-squared test

data:  input
X-squared = 2.1553, df = 3, p-
value = 0.5408

[1] "Null Hypothesis is Accepted"

[Program exited with exit code 0]
```

Practical No: 10

```
# (a) What percentage play in fewer than 750 games?
# (b) What percentage play in more than 2000 games?
# (c) Find the 90th percentile for the number of games played during a career.

print("What percentage play in fewer than 750 games")
pa<-pnorm(750, mean = 1500, sd = 350)
Percenta <- pa*100
print(Percenta)

print("What percentage play in more than 2000 games")
pb<-pnorm(2000, mean = 1500, sd = 350, lower.tail = FALSE)
Percentb <- pb*100
print(Percentb)

print("the 90th percentile for the number of games played during a career")
p05<-round(qnorm(0.05,mean = 1500, sd = 350),0)
p95<-round(qnorm(0.95,mean = 1500, sd = 350),0)
cat("Range for 90 Percentile is : ",p05,"-",p95)
```

Output:-

```
[1] "What percentage play in fewer
than 750 games"
[1] 1.606229
[1] "What percentage play in more
than 2000 games"
[1] 7.656373
[1] "the 90th percentile for the
number of games played during a
career"
Range for 90 Percentile is :  924
- 2076
[Program exited with exit code 0]
```

Practical No: 11

```
prac<-data.frame(
  H=c(70,63,72,60,66,70,74,65,62,67,65,68),
W=c(155,150,180,135,156,168,178,160,132,145,139,152)
)

print("a) H as the independent variable")
reg<-lm(W ~ H,data = prac)
print(reg)

print("b) H as the dependent variable")
```

```
reg<-lm(H ~ W,data = prac)
print(reg)
```

OUTPUT:-

```
[1] "a) H as the independent
variable"

Call:
lm(formula = W ~ H, data = prac)

Coefficients:
(Intercept)            H
    -60.746        3.216

[1] "b) H as the dependent
variable"

Call:
lm(formula = H ~ W, data = prac)

Coefficients:
(Intercept)            W
    31.1078       0.2317


[Program exited with exit code 0]
```

Practical No: 12

a) Graph the data and show the least-squares regression line.
(b) Find and plot the trend line for the data.
(c) Estimate the value of total agricultural exports in the
year 2006.

```
input<-data.frame(
  YEAR=c(2000,2001,2002,2003,2004,2005),
  TOTAL=c(51246,53659,53115,59364,61383,62958)
  )
reg<-lm(TOTAL ~ YEAR,data = input)
print(reg)

plot(input$YEAR,input$TOTAL,type = "p", col = "blue", pch = 16, cex = 1.3,xlab = "Year",ylab
= "Total Value",main = "total agricultural exports")
abline(reg,col = "red")

print("Estimate the value of total agricultural exports in the year 2006.")
newdata = data.frame(YEAR=2006)
predict(reg, newdata)
```

OUTPUT:-

```
Call:
lm(formula = TOTAL ~ YEAR, data =
input)

Coefficients:
(Intercept)         YEAR
   -4976816         2514

[1] "Estimate the value of total
agricultural exports in the year
2006."
       1
65752.27

[Program exited with exit code 0]
```