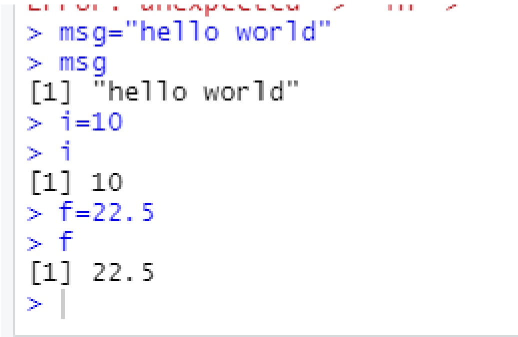
COST PRACTICALS

1. **Using R execute the basic commands ,array,lists and frames.**




```
msg="hello world"
> msg
[1] "hello world"
> i=10
> i
[1] 10
> f=22.5
> f
[1] 22.5
```

VECTORS→ARRAYS

```
> v=c(10,20,30)
> v
[1] 10 20 30>
```

Values with same list and different datatype:

```
 l=list(10,10.5,'A')
> l
[[1]]
[1] 10

[[2]]
[1] 10.5

[[3]]
[1] "A"
```

Assigning vector earlier created

```
 l2=list(10,10.5,'A',v)
> l2
```

```
[[1]]
[1] 10

[[2]]
[1] 10.5

[[3]]
[1] "A"

[[4]]
[1] 10 20 30
```

```
 l2=list(10,10.5,'A',v,l)
> l2
[[1]]
[1] 10

[[2]]
[1] 10.5

[[3]]
[1] "A"

[[4]]
[1] 10 20 30

[[5]]
[[5]][[1]]
[1] 10

[[5]][[2]]
[1] 10.5

[[5]][[3]]
[1] "A"
```

MATRIX

```
> v=c(10,20,30,40)
> v
[1] 10 20 30 40
> m=matrix(v,nrow=2)
> m
```

```
     [,1] [,2]
[1,]   10   30
[2,]   20   40
```

For row wise representation

```
 m=matrix(v,nrow=2,byrow="true")
> m
     [,1] [,2]
[1,]   10   20
[2,]   30   40
>
```

Changing elements in vectors

```
> v[2]=100
> v
[1]  10 100  30  40


> l[4]=20
> l
[[1]]
[1] 10

[[2]]
[1] 10.5

[[3]]
[1] "A"

[[4]]
[1] 20

> m=matrix(l,nrow=2,byrow="true")
> m
     [,1] [,2]
[1,] 10   10.5
[2,] "A"  20
>
```

vector	same	linear
list	different	linear
matrix	Different&same	2D

array	same	2D

Arrays

```
a=array(1:12,dim=c(3,4,2))
> a
, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12


> a=array(c("red","blue"),dim=c(3,4,2))
> a
, , 1

     [,1]   [,2]   [,3]   [,4]  
[1,] "red"  "blue" "red"  "blue"
[2,] "blue" "red"  "blue" "red" 
[3,] "red"  "blue" "red"  "blue"

, , 2

     [,1]   [,2]   [,3]   [,4]  
[1,] "red"  "blue" "red"  "blue"
[2,] "blue" "red"  "blue" "red" 
[3,] "red"  "blue" "red"  "blue"

>
```

Merging 2 vectors and combining it into a table

```
 Name=c("asmita","bharti","aditya")
> age=c(12,34,45)
> details=cbind(Name,age)
> details
```

```
     Name     age
[1,] "asmita" "12"
[2,] "bharti" "34"
[3,] "aditya" "45"
>
```

DATA FRAMES

EXCEL FILE COMES TO R STUDIO IT SEES IT AS DATA FRAMES.

```
students<-data.frame(Name=c('a','b','c'),Age=c(23,45,45),Height=c(140,180,160))
> students
  Name Age Height
1    a  23    140
2    b  45    180
3    c  45    160
>
```

Create a data frame employee having vectors employee name ,emp;loyee id,emp_salary,emp_joining_date,emp_designation.

```
employee<-data.frame(emp_name=c('asmita','aditya','smita','anisha'),emp_id=c(1,4,5,6),emp
_salary=c(12000,40000,56000,89000),emp_joining_date=c('12/6/2022','23/8/2000','23/6/202
1','24/9/2023'),emp_designation=c('hacker','doctor','smoother','cobbler'))
employee
```

Output:

```
> employee
  emp_name emp_id emp_salary emp_joining_date emp_designation
1   asmita      1      12000        12/6/2022          hacker
2   aditya      4      40000        23/8/2000          doctor
3    smita      5      56000        23/6/2021        smoother
4   anisha      6      89000        24/9/2023         cobbler
>
>
>
```

Create vector of fruits and list of colors and merge them.

```
fruits=c('mango','banana','apple')
colors=list('yellow','yellow','red')
druy=cbind(fruits,colors)
druy
```

Output:

```
fruits   colors
```

```
[1,] "mango"  "yellow"
[2,] "banana" "yellow"
[3,] "apple"  "red"
>
```

Create a matrix of name,country and capitalcity of size 4*3

PRACTICAL 2

In the source window,select the code and run.

NEW*

MATRIX

```
 m1=matrix(1:9,nrow=3)
> m2=matrix(10:18,nrow=3)
> m1
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> m2
     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18
```

ADDITION OF A MATRIX

```
m=m1+m2
> m
     [,1] [,2] [,3]
[1,]   11   17   23
[2,]   13   19   25
[3,]   15   21   27
```

```
>
```

SUBTRACTION OF A MATRIX

```
 m=m1-m2
> m
     [,1] [,2] [,3]
[1,]   -9   -9   -9
[2,]   -9   -9   -9
[3,]   -9   -9   -9
>
```

MULTIPLICATION OF A MATRIX.

```
 m=m1*m2
> m
     [,1] [,2] [,3]
[1,]   10   52  112
[2,]   22   70  136
[3,]   36   90  162
```

DIVISION OF A MATRIX

```
m=m1/m2
> m
          [,1]      [,2]      [,3]
[1,] 0.1000000 0.3076923 0.4375000
[2,] 0.1818182 0.3571429 0.4705882
[3,] 0.2500000 0.4000000 0.5000000
```

TRANSPOSE OF A MATRIX -row to column and column to row.

```
 tm=t(m1)
> tm
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
>
```

INVERSE OF A MATRIX.

Determinant

```
 det(m1)
[1] 0
> det(m2)
[1] 5.329071e-15
```

NEW MATRIX.

```
m3=matrix(c(10,2,3,5),nrow=2)
> m3
     [,1] [,2]
[1,]   10    3
[2,]    2    5


> det(m3)
[1] 44
> solve(m3)
            [,1]        [,2]
[1,]  0.11363636 -0.06818182
[2,] -0.04545455  0.22727273
>
```

PRACTICAL 3 & PRACTICAL 4

Using R Execute the statistical functions:mean, median, mode, quartiles, range, inter quartile range histogram

1.CREATE EXCEL FILE.
2.ENTER THE DETAILS INTO THAT
3.SAVE THE FILE WITH .CSV(COMMA DELIMITED).
4.CLICK YES AND OK
5. AFTER THAT GO TO RSTUDIO
6.AND IN ENVIRONMENT SECTION GO TO IMPORT DATASET.AND FIND YOUR FILE AND IMPORT IT.
7.AFTER THAT IMPORT QUERIES
mean([filename]$[columnname])
ex)

MEAN:

NOTE: write code in source

```
mean(student$marks)
median(student$marks).
```

Code:

```
meanval=function(v)
+ {
+   l=length(v)
+   sum=0
+   for(i in v)
+   {
+     sum=sum+i
```

```
+   }
+   avg=sum/l
+   avg
+ }
> meanval(student$MARKS)
[1] 83.8
```

MEDIAN:

```
med=function(v)
+ {
+   l=length(v)
+   v=sort(v)
+   if(l%%2==0)
+   {
+     p=l/2
+     q=p+1
+     md=(v[p]+v[q])/2
+   }
+   else
+   {
+     p=(l+1)/2
+     md=v[p]
+   }
+   md
+ }
>
>
> med(student$MARKS)
[1] 93.5
> med(student$MARKS)
[1] 93.5
>
```

With vectors.

```
v=c(1,2,3,4,5)
med(v)
```

Output;

```
[1] 3
```

MODE:

UNIQUE- ONLY UNIQUE VALUES
MAX= MAXIMUM VALUE
TABLE= RETURN FREQUENCY
MATCH= CHECK X VALUES AND RETURN Y POSITION WITH RESPECT TO X

MATCH(X,Y)

2->1

WHICH PROPRETY- RETURN THE POSITION.

ex)

```
getmod=function(x)
{
  uniqx=unique(x)
  mod=uniqx[which.max(table(match(x,uniqx)))]
  mod
}
getmod(student$MARKS)
```

OUTPUT:

```
> getmod(student$MARKS)
[1] 99
>
```

EXPLAINATION

x=(6,4,6,4,8.6)
uniqx=(6,4,8)

Match
6->1,4->2,6->1,4->2,8->3,6->1

Table
position-> occurrence
1->3
2->2
3->1

max=3 occurrence=1(position)

mod[1]

Therefore the mod is 6.

PRACTICAL 5

VARIANCE.- DIFFERENCE OF AVERAGE WITH RESPECT TO POSITION.

```
> var(student$MARKS)
[1] 573.2889
> var(student$MARKS)
[1] 573.2889
```

VARIANCE CODE:

```
varx=function(x)
{
  n=length(x)
  m=meanval(x)
  sum=0
  for(i in x)
  {
    sqdiff=(i-m)^2
    sum=sum+sqdiff
  }
  variance=sum/n
  variance
}


varx(student$MARKS)
```

OUTPUT:

```
> varx(student$MARKS)
[1] 515.96
> varx(student$MARKS)
[1] 515.96
>
```

PRACTICAL 5

COVARIANCE

```
covar=function(x,y)
{
  mk=meanval(x)
  at=meanval(y)
  n=length(x)
  sum=0
  for(i in seq_along(x))
  {
```

```
    cvmarks=x[i]-mk
    cvattd=y[i]-at
    mul=cvmarks*cvattd
    sum=sum+mul
  }
  cv=sum/(n-1)
  cv
}
```

OUTPUT:

```
> covar(stud45$MARKS,stud45$ATTENDANCE)
[1] 250.6889

> cov(stud45$MARKS,stud45$ATTENDANCE)
[1] 250.6889
```

STANDARD DEVIATION

```
std=function(x)
{
  v=var(x)
  s=sqrt(v)
  s
}
```

OUTPUT:

```
> std(stud45$MARKS)
[1] 28.22782
```

PRACTICAL 6

-0.5 to 0.5= fairly skewed
-0.5 to -1=negatively moderately skewed
-0.5 to 1=positively moderately skewed.

SKEWNESS

```
sk1=function(x)
{
  s=(3*meanval(x)-med(x))/std(x)
  s
}
```

OUTPUT:

```
> sk1(stud45$MARKS)
```

```
[1] 4.718576
> sk(stud45$MARKS)

[1] 4.718576
```

PRACTICAL 7

HYPOTHESIS

QUESTION:

Mean weight=15.4kg
Sample Size=35
Sample mean=14.6kg
Sd=2.5

You can't say that your hypothesis is correct

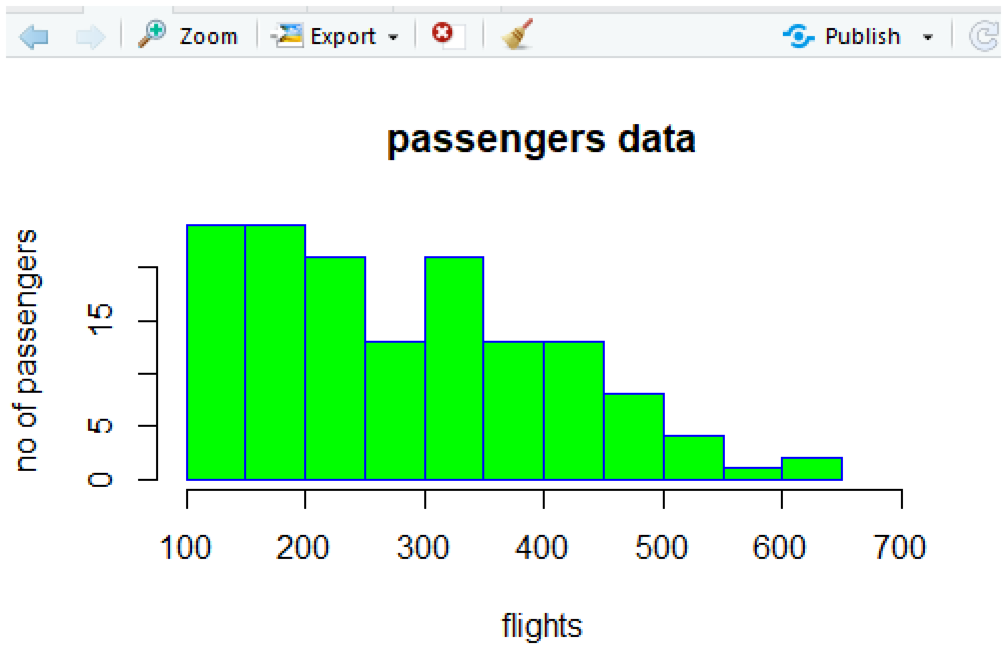
PRACTICAL 3
Using R Execute the statistical functions:mean, median, mode, quartiles, range, inter quartile range histogram

HISTOGRAM

CODE:

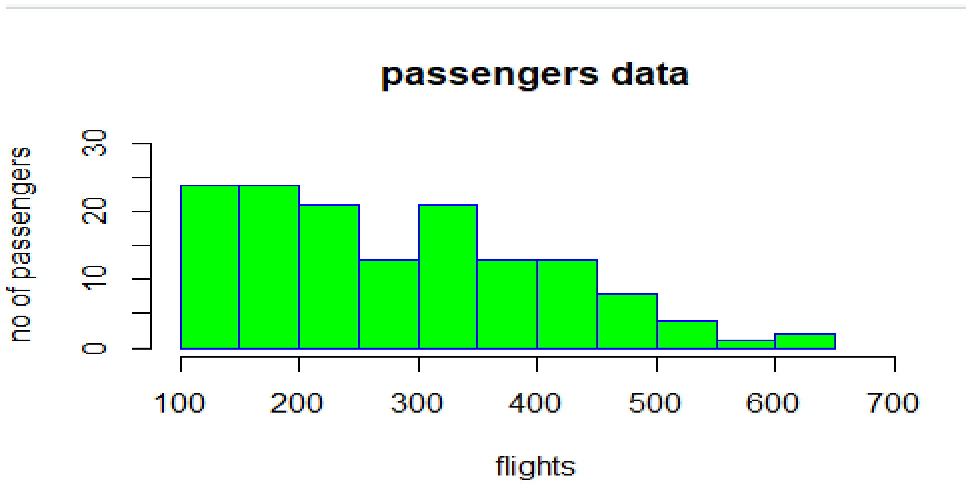
```
hist(AirPassengers,
main="passengers data",
xlab="flights",
ylab="no of passengers",
xlim=c(100,700),
col="green",
border="blue",
breaks=10
)
```

OUTPUT:




```
hist(AirPassengers,
main="passengers data",
xlab="flights",
ylab="no of passengers",
xlim=c(100,700),
ylim=c(0,30),
col="green",
border="blue",
breaks=10
)
```

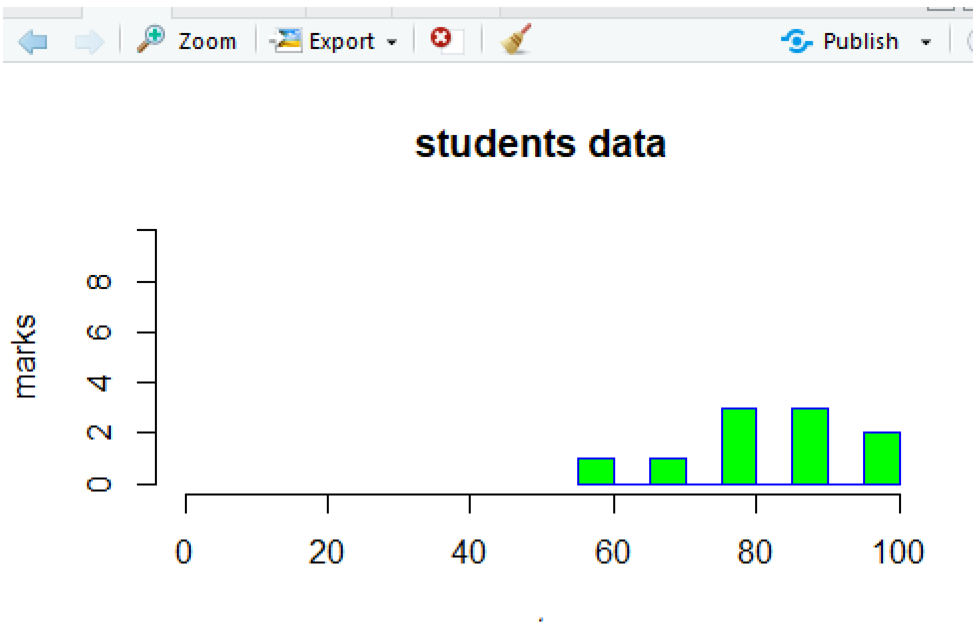
OUTPUT:



WITH STUDENTS DATA

```
hist(student$MARKS,
main="students data",
xlab="stu",
ylab="marks",
xlim=c(0,100),
ylim=c(0,10),
col="green",
border="blue",
breaks=10
)
```

OUTPUT:



QUARTILES

```
quantile(student$MARKS)
```

OUTPUT

```
> quantile(student$MARKS)
   0%    25%    50%    75%   100%
 56.00  77.25  83.50  90.00 100.00
```

InterQuartileRange:(Between 25% and 75%)(Subtraction)

IQR(student$MARKS)

OUTPUT:

```
> IQR(student$MARKS)
[1] 12.75
```

RANGE:

```
> range(student$MARKS)
[1]  56 100
```

PRACTICAL 7

Import the data from Excel / .CSV and perform the hypothetical testing.

Question1: (WEIGHTS OF PENGUINS)
u=15.4kg
n=35
x=14.6kg
sd=2.5
Level of significance=0.05

a) H0=w>=15.4(lower tail test)
b) H0=w<=15.4(Upper tail test)
c) H0=w=15.4(Equality test)-it doesnt matter whether it is grater or lesser but is there is a change or not.

SOLUTION:
a)HO:w>=15.4
H1:w<15.4

```
u=15.4
n=35
x=14.6
sd=2.5
t=(x-u)/(sd/sqrt(n))
t
p=pt(t,df=n-1)
p
```

OUTPUT:

```
> t
[1] -1.893146
> p=pt(t,df=n-1)
> p
[1] 0.03343776  // 3%
>
```

THIS NULL HYPOTHESIS IS REJECTED

b)(STUDENTS SLEEP HOURS)
HO=w<=15.4
H1=w>15.4

```
p=pt(t,df=n-1,lower.tail=FALSE)
P
```

OUTPUT:

```
> p=pt(t,df=n-1,lower.tail=FALSE)
> p
[1] 0.9665622  // 96%
```

THIS NULL HYPOTHESIS IS ACCEPTED

c)
HO=w=15.4
H1=w1=15.4

```
p=2*pt(t,df=n-1)
p
```

OUTPUT:

```
> p=2*pt(t,df=n-1)
> p
[1] 0.06687552
```

Question 2:

x=7.24
u=7
n=22
sd=1.93
Significance level=5%

Solution:

H0:hrs<=7
H1:hrs>7

CODE:

```
u=7
n=22
x=7.24
sd=1.93
t=(x-u)/(sd/sqrt(n))
t

p=pt(t,df=n-1,lower.tail=FALSE)
p
```

OUTPUT:

```
> u=7
> n=22
> x=7.24
> sd=1.93
> t=(x-u)/(sd/sqrt(n))
> t
[1] 0.5832641


> p=pt(t,df=n-1,lower.tail=FALSE)
> p
[1] 0.2829637            // 28%
```

THIS NULL HYPOTHESIS IS ACCEPTED.

Question 3:go for sample sd not for population

x=46500
u=50000
n=28
sd=9800
Significance level =5%

Solution:

HO=L>=50000
H1=L<50000

CODE:

```
u=50000
n=28
x=46500
sd=9800
t=(x-u)/(sd/sqrt(n))
t
p=pt(t,df=n-1)
p
```

OUTPUT:

```
> u=50000
> n=28
> x=46500
> sd=9800
> t=(x-u)/(sd/sqrt(n))
> t
[1] -1.889822
> p=pt(t,df=n-1)
> p
[1] 0.03478162     //3%
>
```

THIS NULL HYPOTHESIS IS REJECTED.

Question 4

x=18.1
u=19
n=40
sd=1.3
Significance level=5%

Solution:
H0: age>=19
H1: age<19

CODE:

```
u=18.1
n=19
x=40
sd=1.3
t=(x-u)/(sd/sqrt(n))
t
p=pt(t,df=n-1)
p
```

OUTPUT:

```
> u=18.1
> n=19
> x=40
> sd=1.3
> t=(x-u)/(sd/sqrt(n))
> t
[1] 73.43068
> p=pt(t,df=n-1)
> p
[1] 1
```

THIS HYPOTHESIS WILL BE REJECTED.

PRACTICAL 8

CHI SQUARED TEST- It is done when columns contain limited data
Test for 2 categorical variables to check whether they are related or bot
Ex male and female(2 categories)

CODE:

```
library(MASS)
print(str(survey))
```

```
> print(str(survey))
'data.frame':   237 obs. of  12 variables:
 $ Sex   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
 $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...
 $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...
 $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...
 $ Fold  : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...
 $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...
 $ Clap  : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...
 $ Exer  : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
 $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...
 $ Height: num  173 178 NA 160 165 ...
 $ M.I   : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...
 $ Age   : num  18.2 17.6 16.9 20.3 23.7 ...
NULL
> stu_data=data.frame(survey$Smoke,survey$Smoke)
```

```
stu_data=data.frame(survey$Smoke,survey$Exer)
stu_data=table(survey$Smoke,survey$Exer)
print(stu_data)
```

OUTPUT:

```
> print(stu_data)

        Freq None Some
  Heavy    7    1    3
  Never   87   18   84
  Occas   12    3    4
  Regul    9    1    7
>
```

print (chisq.test(stu_data,correct = FALSE))

```
	Pearson's Chi-squared test

data:  stu_data
X-squared = 5.4885, df = 6, p-value = 0.4828
```

Q.RELATION BETWEEN THE SURVEY FOLD AND EXERCISES

ho=smoking<>Exercise accept
h1=Smoking~=Exercise

stu_data=data.frame(survey$Fold,survey$Exer)
stu_data=table(survey$Fold,survey$Exer)
print(stu_data)
print (chisq.test(stu_data,correct = FALSE))

```
> print (chisq.test(stu_data,correct = FALSE))

	Pearson's Chi-squared test

data:  stu_data
X-squared = 1.2213, df = 4, p-value = 0.8746
```

print(str(oats))
print(str(Melanoma))

```
  Chi-squared approximation may be incorrect
> print(str(oats))
'data.frame':   72 obs. of  4 variables:
 $ B: Factor w/ 6 levels "I","II","III",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ V: Factor w/ 3 levels "Golden.rain",..: 3 3 3 3 1 1 1 1 2 2 ...
 $ N: Factor w/ 4 levels "0.0cwt","0.2cwt",..: 1 2 3 4 1 2 3 4 1 2 ...
 $ Y: int  111 130 157 174 117 114 161 141 105 140 ...
NULL
> print(str(Melanoma))
'data.frame':   205 obs. of  7 variables:
 $ time     : int  10 30 35 99 185 204 210 232 232 279 ...
 $ status   : int  3 3 2 3 1 1 1 3 1 1 ...
 $ sex      : int  1 1 1 0 1 1 1 0 1 0 ...
 $ age      : int  76 56 41 71 52 28 77 60 49 68 ...
 $ year     File Explorer 72 1968 1977 1968 1965 1971 1972 1974 1968 1971 ...
```

Mel_data=data.frame(Melanoma$status,Melanoma$sex)
Mel_data=table(Melanoma$status,Melanoma$sex)
print(Mel_data)
print(chisq.test(Mel_data))

```
> Mel_data=table(Melanoma$status,Melanoma$sex)
> print(Mel_data)

     0  1
  1 28 29
  2 91 43
  3  7  7
> print(chisq.test(Mel_data))

        Pearson's Chi-squared test

data:  Mel_data
X-squared = 6.793, df = 2, p-value = 0.03349

> 
```

REJECT THE NULL HYPOTHESIS
Q.RELATION BETWEEN CARS TYPE AND ENGINESIZE
print(str(Cars93))
car_data=data.frame(Cars93$Type,Cars93$EngineSize)
car_data=table(Cars93$Type,Cars93$EngineSize)
print(car_data)
print(chisq.test(car_data))

```
> car_data=data.frame(Cars93$Type,Cars93$EngineSize)
> car_data=table(Cars93$Type,Cars93$EngineSize)
> print(car_data)

          1 1.2 1.3 1.5 1.6 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.8 3 3.2 3.3 3.4 3.5 3.8 4.3 4.5 4.6
  Compact 0   0   0   0   0   0   0 2   1   4   4   1   2   1 1   0   0   0   0   0   0   0   0
  Large   0   0   0   0   0   0   0 0   0   0   0   0   0   0 0   0   2   0   1   3   0   0   2
  Midsize 0   0   0   0   0   0   0 1   0   4   0   1   1   1 5   2   0   1   1   3   0   1   1
  Small   1   1   2   6   3   6   1 0   0   1   0   0   0   0 0   0   0   0   0   0   0   0   0
  Sporty  0   0   1   1   2   1   0 1   0   1   2   0   0   1 1   0   0   2   0   0   0   0   0
  Van     0   0   0   0   0   0   0 0   0   0   0   1   1   0 4   0   0   0   0   2   1   0   0

          4.9 5 5.7
  Compact   0 0   0
  Large     1 1   1
  Midsize   0 0   0
  Small     0 0   0
  Sporty    0 0   1
  Van       0 0   0
> print(chisq.test(car_data))

        Pearson's Chi-squared test

data:  car_data
X-squared = 201.26, df = 125, p-value = 1.814e-05

Warning message:
```

REJECT THE NULL HYPOTHESIS

PRACTICAL 9

BINOMIAL DISTRIBUTION

Pdimom -cumalative
dbinom=specific

dbinom(4,size=12,prob=0.2)
[1] 0.1328756
> pbinom(4,size=12,prob=0.2)
[1] 0.9274445
> dbinom(4,size=12,prob=0.8)
[1] 0.0005190451
> pbinom(4,size=12,prob=0.8)
[1] 0.0005812429
> dbinom(4,size=12,prob=0.8)+
+ pbinom(3,size=12,prob=0.8)+
+ pbinom(2,size=12,prob=0.2)+
+ pbinom(1,size=12,prob=0.2)+
+ pbinom(0,size=12,prob=0.2)
[1] 0.9025244
> dbinom(4,size=12,prob=0.8)+
+ pbinom(3,size=12,prob=0.8)+
+ pbinom(2,size=12,prob=0.8)+
+ pbinom(1,size=12,prob=0.8)+
+ pbinom(0,size=12,prob=0.8)

```
[1] 0.0005859738
> dbinom(4,size=12,prob=0.8)+
+ dbinom(3,size=12,prob=0.8)+
+ dbinom(2,size=12,prob=0.8)+
+ dbinom(1,size=12,prob=0.8)+
+ dbinom(0,size=12,prob=0.8)
[1] 0.0005812429
>
```

Question 1

```
dbinom(6,size=10,prob=0.5)
pbinom(6,size=10,prob=0.5)
```

```
> dbinom(6,size=10,prob=0.5)
[1] 0.2050781
> pbinom(6,size=10,prob=0.5)
[1] 0.828125
>
```

```
dbinom(0,size=9,prob=0.8)
dbinom(9,size=9,prob=0.2)
```

Mens question

NORMAL DISTRIBUTION

```
pno,m
pnorm(20.08,mean=20.05,sd=0.02)-
  pnorm(20.03,mean=20.05,sd=0.02)
```

```
[1] 0.4331928
> pnorm(20.08,mean=20.05,sd=0.02)-
+   pnorm(20.03,mean=20.05,sd=0.02)
[1] 0.7745375
>
```

```
pnorm(20.00,20.05,0.02)
1-pnorm(20.09,20.05,0.02)
1-pnorm(84,72,15.2)
```

```
> pnorm(20.00,20.05,0.02)
[1] 0.006209665
> 1-pnorm(20.09,20.05,0.02)
[1] 0.02275013
> 1-pnorm(84,72,15.2)
[1] 0.2149176
>
```

PRACTICAL 10

LINEAR REGRESSION - PREDICTION

```
sumxy=function(x,y)
{
  n=length(x)
  sum=0
  for(i in seq_along(x))
  {
    sum=sum + x[i]*y[i]
  }
  sum
}

v=c(1,2,3,4,5)
v
x=c(2,4,6,8,10)
x
sumxy(v,x)
```

```
sum=function(v)
{
  s=0
  for (i in v)
  {
    s=s + i
  }
  s
}

sum(v)

sumsq=function(v)
{
  s=0
  for(i in v)
```

```
 {
   s=s+i^2
 }
 s
}
sumsq(v)
 }
}
```

```
R 3.3.1 · ~/
[1] 1 2 3 4 5
> x=c(2,4,6,8,10)
> x
[1]  2  4  6  8 10
> sumxy(v,x)
[1] 110
> sum=function(v)
+ {
+   s=0
+   for (i in v)
+   {
+     sum=sum + i
+   }
+   s
+ }
>
> sum(v)
Error in sum + i : non-numeric ar
> sum=function(v)
+ {
+   s=0
+   for (i in v)
+   {
+     s=s + i
+   }
+   s
+ }
>
> sum(v)
[1] 15
> sumsq=function(v)
+ {
+   s=0
+   for(i in v)
+   {
+     s=s+i^2
+   }
+   s
+ }
> sumsq(v)
[1] 55
>
File Explorer
```

```
n=length(v)
n
txy=sumxy(v,x)
txy
psxy=sum(v)*sum(x)
psxy
```

tsqx=sumsq(v)
tsqx
sqsx=sum(v)*sum(v)
sqsx

```
> n
[1] 5
> txy=sumxy(v,x)
> txy
[1] 110
> psxy=sum(v)*sum(x)
> psxy
[1] 450
> tsqx=sumsq(v)
> tsqx
[1] 55
> sqsx=sum(v)*sum(v)
> sqsx
[1] 225
>
```

tx=sum(v)
tx
ty=sum(x)
ty

b=((n*txy)-psxy)/((n*tsqx)-sqsx)
b

a=((ty*tsqx)-tx*txy)/((n*tsqx)-sqsx)
a

```
> tx=sum(v)
> tx
[1] 15
> ty=sum(x)
> ty
[1] 30
> b=((n*txy)-psxy)/((n
> b
[1] 2
> a=((ty*tsqx)-tx*txy),
> a
[1] 0
>
```

y=a+b*6

y

```
> y=a+b*6
> y
[1] 12
>
```

PRACTICAL 11 & 12

Least square means-bls

Less error nice prediction

bls=(ty-(b*tx))/n
bls

x=50
y=b*50+0
y

```
[1] 12
> bls=(ty-(b*tx))/n
> bls
[1] 0
> y=b*50+0
> y
[1] 100
>
```

```
error=function(x,y)
{
  yn=0
  for(i in seq_along(x))
  {
    yn=b*i+bls
    cat(sprintf("error for %d is %f:\n",i,yn-y[i]))
  }
}
error(v,x)
```

```
+     cat(sprintf("error for %d i
+   }
+ }
> error(v,x)
error for 1 is -5.800000:
error for 2 is 2.300000:
error for 3 is 6.400000:
error for 4 is 3.500000:
error for 5 is -6.400000:
>
```