Roll No : 54

Practical No. 01

(1)

- Using R execute the basic commands, array, list and frames

Variable declaration

```
> x = 5
> x
[1]    5

> x = c(1,2,3)
> y = c(1,3,4)
> Sum = x+y
> Print (sum)
[1]    2 5 7
```

List -

```
>Studentlist <- list ("Sara", "Tanaya", "Soham", "Parth")
>Print (Studentlist)
 [1]   Sara
 [2]   Tanaya
 [3]   Soham
 [4]   Parth
```

Adding new name -

```
> Studentlist [5] <- "Rohan"
> Print (studentlist [5])
[1]    Rohan
```

Frame -

```
> Srno <- c (1, 2, 3)
> examno <- c (4, 5, 6)
> name <- ("Tanaya", "Rutwik", "Sara")
> df <- data.frame (Srno, examno, name)
> Print (df)

[1]      Srno     examno     name
          1         4        Tanaya
          2         5        Rutwik
          3         6        Sara

> df $marks <- c (44, 46, 48)
> Print (df)

[1]      Srno     examno     name      marks
          1         4        Tanaya     44
          2         5        Rutwik     46
          3         6        Sara       48
```

Array -

```
> a <- (1, 2, 3)
> b <- (4, 6, 5)
> c <- (6, 7, 8)
> ans <- array (c (a, b, c), dim = c (3, 3))
> Print (ans)

[1]    1    4    6
[2]    2    6    7
[3]    3    5    8
```

Roll No : 54

Practical No. 02

- Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operation.

Matrix

```
> a <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
> A <- matrix (a, nrow = 3 , ncol = 3 , byrow = "True")
> B <- matrix (a, nrow = 3 , ncol = 3 , byrow = "False")
> Print (A)
[1]      1    2    3
[2]      4    5    6
[3]      7    8    9
> Print (B)
[1]      1    4    7
[2]      2    5    8
[3]      3    6    9
```

Addition

```
> a <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
> A <- matrix (a, nrow = 3 , ncol = 3 , byrow = "True")
> B <- matrix (a, nrow = 3 , ncol = 3 , byrow = "False")
> Sum <- A + B
> Print (Sum)

[1]      2    6    10
[2]      6    10   14
[3]      10   14   18
```

Transpose

```
> a <- c (1, 2, 3, 4, 5, 6, 7, 8, 9)
> A <- matrix ( a, nrow = 3, ncol = 3 , byrow = "True")
> t (A)

[1]    1    4    7
[2]    2    5    8
[3]    3    6    9
```

Inverse

```
> a <- c (1, 2, 3, 4)
> A < - matrix ( a, nrow = 2 , ncol = 2, byrow = "True")
> Print (A)
[1]    1    2
[2]    3    4

> a <- c (1, 2, 3, 4)
> A < - matrix (a, nrow = 2 , ncol = 2 , byrow = "True")
> IA <- solve (A)
> Print (IA)

[1]    -2.0    1.0
[2]     1.5   -0.5
```

Roll No: 54

Practical No. 03

- Using R Execute the statistical functions : Mean, median, mode, quartiles, range, inter quartile range histogram.

Mean :

1) For ungrouped data

Q. Find mean for data (1,2,3,4,5)

```
umean = function(x)
{
mean = sum(x)/length(x);
Print(mean)
}
x <- c (1,2,3,4,5)
umean(x)
>> 3
```

2) For grouped data

Q. Find the mean of group data.

C.I	10-20	20-30	30-40
f	5	4	7

```
gmean = function (ll + ul, f)
{
x = (ll+ul)/2
mean = sum (f*x)/sum (f)
Print (mean)
}
ll <- c (10, 20, 30)
ul <- c (20, 30, 40)
f <- c (5, 4, 7)
```

```
gmean (ll, ul, f)
```

[1] 26.25

Median

1) For Ungrouped data

Q. Find the median for data (3, 4, 2, 5, 1, 6)

```
umedian = function (x)
{
xi <- sort (x)
 n = length (xi)
 c = n/2
 d = c+1
 p = xi [c]
 q = xi [d]
median = (c+d)/2
> Print (median)
> x <- c (3, 4, 2, 5, 1, 6)
> umedian (x)
```

[1] 3.5

2) For grouped data

Q. Find the median for data

CI	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
f	4	9	19	20	18	6	2	1	1	0

```
> gmedian = function (ll, ul, fi)
  {
  cf = cumsum (fi);
  n = sum (fi);
  m = n/2 ;
  ccf = min ( cf [which ( cf >=m)]) ;
  a = which ( cf= = ccf);
  b = a - 1 ;
  f = fi [a];
  pcf = cf [b];
  l1 = ll [which ( cf = = ccf)]
  l2 = ul [which ( cf = = ccf)]
  i = l2 - l1
  median = l1 + ( ( (m-pcf) / f ) *i)
  Print (median)
  }
> ll = c (0, 10, 20, 30, 40, 50, 60, 70, 80, 90)
> ul = c ( 10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
> fi = c ( 4, 9, 19, 20, 18, 6, 2, 1, 1, 0 )
> gmedian ( ll, ul, fi)

[1] 34
```

Mode

1) For Ungrouped data

Q. Find the mode for data

x	3	2	8
f	4	3	5

```
umode = function (x, f)
{
mode = x [which (f == max (f))]
Print (mode)
}
x = c (3, 2, 8)
f = c (4, 3, 5)
umode (x, f)

[1] 8
```

2) For grouped data

Q. Find the mode for data

CI	0-25	25-50	50-75	75-100
f	500	700	800	400

```
gmode = function (ll, ul, f)
{
f1 = max (f);
l1 = ll [which (f == f1)]
l2 = ul [which (f == f1)]
```

```
a = which (f==f1)
b = a-1;
c = a+1;
f0 = f[b];
f2 = f[c];
i = l2-l1;
p = f1-f0;
q = f1-f2;
mode = l1 + ((p/(p+q))*i);
disp(mode)
}
ll = c(0, 25, 50, 75)
ul = c(25, 50, 75, 100)
f = c(500, 700, 800, 400)
gmode(ll, ul, f)
```

[1] 55

Quartile

Q. Data

CI	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
f	4	9	19	20	18	6	2	1	1	0

```
gquartile = function (ll, ul, fi)
{
cf = cumsum(fi);
n = sum(fi);
for (k in 1:3)
{
```

```
m = K*n/4;
ccf = min (cf[which (cf >= m)]);
a = which (cf == ccf);
b = a-1;
f = fi[a];
pcf = cf[b];
l1 = ll[which (cf == ccf)];
l2 = ul[which (cf == ccf)];
i = l2-l1
quartile = l1 + ((m-pcf)/(f) * i);
print (quartile)
}
}
ll = c(0,10,20,30,40,50,60,70,80,90)
ul = c(10,20,30,40,50,60,70,80,90,100)
f = c(4,9,19,20,18,6,2,1,1,0)
gquartile (ll, ul, fi)

[1]    23.68421
[1]    34
[1]    44.4444
```

Range

1) For ungrouped data

Q. Find the range of data (1, 2, 3, 4, 5)

```
urange = function (x)
{
s = min (x)
m = max(x)
range = m - s ;
Print (range)
}
x = c (1,2,3,4,5)
urange (x)
```

[1] 4

2) For grouped data

Q. Find the range for data

x	10-20	20-30	30-40	40-50
f	2	6	9	3

```
grange = function (ul, ll, f)
{
m = max (ul)
s = min (ll)
range = m-s ;
Print (range)
}
ul = c (20, 30, 40, 50)
ll = c (10, 20, 30, 40)
```

grange (ul, ll, f)

[1] <u>40</u>

Roll No. 54

Practical No. 04

D D M M Y Y Y Y

- Using R import the data from Excel/.csv file and Perform the above functions.

1) Mean

Q. Find mean for following data:

x	2	3	6	7	9
f	2	1	4	2	1

```
table <- read.csv(file.choose(), header = T)
table
   x   f
   2   2
   3   1
   6   4
   7   2
   9   1
mean = sum(table$f * table$x)/sum(table$f)
mean
[1]  5.4
```

Q. Find mean for the data:

x	10-20	20-30	30-40	40-50
f	5	6	7	8

```
table <- read.csv(file.choose(), header =T)
table
  ll     ul    f
  10     20    5
```

```
20      30      6
30      40      7
40      50      8
gmean = function (ll, ul, f)
{
x = (ll + ul) /2
mean = sum (f * x) / sum(f);
Print (mean);
}
gmean (table$ll, table $ul, table $ f)
[1]   31.92508
```

2) Median

For Grouped data

```
table <- read.csv(file.choose(), header = T)
table
ll    ul    fi
10    90    50
20    80    60
30    70    30
40    60    20
50    50    10
60    40    40
70    30    90
80    20    80
90    10    70
gmedian = function (ll, ul, fi)
{
n = sum(fi)
m = n/2
```

```
cs = cumsum(fi)
cf = min (cs[which (cs>m)])
a = which (cs==b)
b = cs[a]
l1 = ll [which (cs==b)]
l2 = ul [which (cs==b)]
f = fi[which (cs==b)]
c = a-1
d = cs[c]
median = l1+((m-d)/f)*(l2-l1)
print (median)
}
gmedian (table$ ll, table$ ul ,table$ fi)
[1]   63.33333
```

3) Mode

For Ungrouped data

```
> table <- read.csv (file.choose(), header =T)
table
xi   fi
1    3
2    6
3    5
4    8
5    7
mode = function (xi,fi)
{
mode = xi [which (fi >= max(fi))]
print (mode)
}
```

mode (table $ xi , table $ fi)

[1] 4

4) Quartile

table <- read.csv (file.choose (), header = T)
table

ll	ul	fi
0	10	4
10	20	9
20	30	19
30	40	20
40	50	18
50	60	6
60	70	2
70	80	1
80	90	1
90	100	0

```
gquartile <= function (table$ll, table$ul, table$fi)
{
n = sum (fi)
cs = cumsum (fi)
for (k in 1:3)
{
m = (k*N)/4
cf = min (cs [which (cs > m)])
a = which (cs == cf)
b = a-1
l1 = ll [which (cs == cf)]
l2 = ul [which (cs == cf)]
```

```
f = fi [which (cs == cf)]
pcf = cs [b]
i = l2 - l1 ;
Quartile = l1 + (( m - pcf)/(f) * i);
Print (quartile)
}
gquartile (table$ul, table$ul, table$fi)
```

```
[1]  23.68421
[1]  34
[1]  44.4444
```

Roll No. 54

Practical No. 05

- Using R import data from excel/.csv file and calculate standard deviation, variance and co-variance.

1) Find s.d of 2, 3, 6, 7, 8.

```
table <- read.csv(file.choose(), header = T)
table
x
2
3
6
7
8
usd = function(x)
{
n = length(x);
a = sum(x * x);
b = sum(x);
sd = sqrt((a/n) - (b/n)^2);
print(sd)
}
usd(table$x)
[1]   2.315167
```

2) standard deviation for grouped data

x	2	3	4	5	6	9
f	1	2	3	4	5	6

```
table <- read.csv(file.choose(), header = T)
table
```

x	f
2	1
3	2
4	3
5	4
6	5
9	6

```
gsd = function (x,f)
{
n = sum (table$x);
a = sum (table$f * table$x * table$x);
b = sum (table$f * table$x);
sd = sqrt ((a/n) - ((b/n)^2));
print (sd);
}
gsd (table$x, table$f)
[1]   3.247243
```

Variance

```
table <- read.csv (file.choose(), header = T)
table
```

x	f
2	1
3	2
4	3
5	4
6	5
9	6

```
gsd = function (x.f)
{
n= sum (table$x)
a = sum (table$f * table $x * table $x);
b = sum (table$f * table$x);
sd = sqrt ((a/n) - ((b/n) ^2));
print (Paste ("sd =", sd));
var = sd^2;
Print ( paste ("Variance - ", Var))
}
gsd (table$x, table$f)
[1] "sd = 3.247243
[1] "Variance - 10.54458
```

Roll No. 54

Practical No. 06

DD MM YYYY

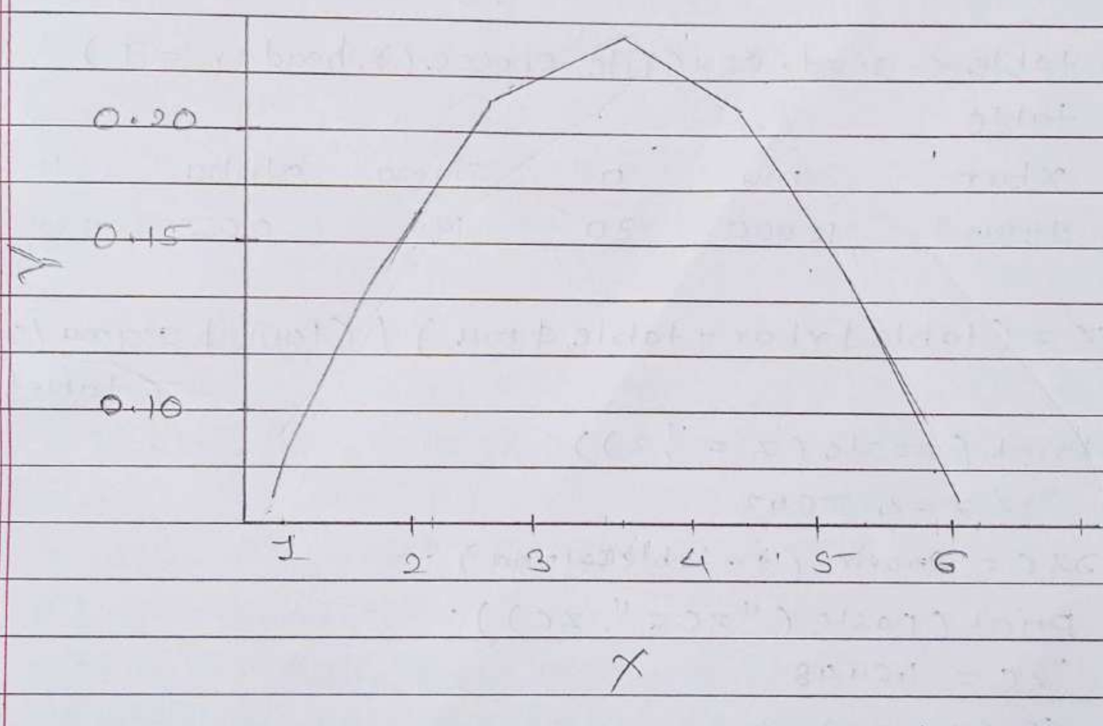
- Using R import the data from Excel/.CSV file and draw the skewness.

1) Skewness for Ungruped data

```
uskewness = function(x)
{
n= length(x)
a= sum(x)
mean = a/n
print(mean)
s= sort(x)
c= n/2
d= c+0.5
median = s[d]
print(median)
e = sum(x*x)
sd = sqrt((e/n)-(a/n)^2)
print(sd)
skew = 3*(mean-median)/sd
print(skew)
}
x = c(3.1, 4.1, 2.2, 5.7, 6.7)
uskewness(x)
[1]   4.36
[1]   4.1
[1]   1.648757
[1]   0.4730836
```

e) Draw skewness

```
x = c(1, 2, 3, 3.5, 4, 5, 6)
y = dnorm (x, mean (x), sd (x))
print (y);
plot (x,y)
plot (x, y, type = "l")
[1] 0.08001127  0.15883751  0.22379689  0.23359668
    0.22379689
[1] 0.15883751  0.08001127
```



Roll No: 54

Practical No. 07

- Import the data from Excel / .csv and perform the hypothetical testing.

1) The manufacturer claim that the mean lifetime of light bulb is 10000 hours in an sample of 30 light bulbs it was found that they only last 9900 hours on average. Assume the population standard deviation is 120 hours at 5% level of significance. Can we reject the claims by the manufacturer?

→ Code :

```
# H0: mu = 10,000, we can not reject hypothesis
# H1: mu != 10,000, we can reject hypothesis

> table <- read.csv(file.choose(), header = T)
> table
  xbar     mu      n    sigma   alpha
  9900    10000   30    120     0.05

> z = (table$xbar - table$mu) / (table$sigma / sqrt(table$n));
> Print (paste ("z = ", z));
[1] " z = -4.5643
> zc = qnorm (1 - table$alpha)
> print (paste ("zc = ", zc));
[1] "zc = 1.6448
> if (z > zc)
  {
   print ("ho is accepted");
  } else
  {
   print (" ho is rejected");
```

}

[1] " ho is rejected "

2) An examination was given two 2 classes consisting of 40 and 50 students respectively in the first class mean grade was 74. with sd of 8. while in the second class mean grade was 78 with sd of 7. is their significance differnce between performance of two classes at 0.05 significance level.

→ Code :

```
# H0 : mu1 = mu2 there is no significance diff between
  two classes
# H1 : m1 != m2 there is significance diff between two
  classes

> table <- read.csv(file.choose(), header = T)
> table
  xbar1   xbar2   s1   s2   n1   n2   alpha
    74      78     8    7   40   50   0.05
> sig = sqrt(((table$s1^2)/table$n1) + ((table$s2^2)
                               /table$n2));
> print(sig);
[1]  1.606238
> z = ((table$xbar1 - table$xbar2) / sig);
> print (z);
[1]    -2.490291
> zc = qnorm (1- (table$alpha/2));
 print (zc);
[1]  1.959964
>  if (-zc<z && z<zc)
```

```
{
print (" ho is accepted ")
} else
{
Print ("ho is rejected ")
}
[1] "ho is rejected
```

Hypothetical testing by using p-value

1) A group of 50 internet shoppers were ask how much they spend per year on the internet their response are shown in below table. It is desired to test that they spend Rupees 325 per year vs it is different from 325. Find the p-value for the test of hypothesis. what is your conclusion for $\alpha = 0.05$

→

```
> table <- read.csv(file.choose(), header =T)
> table
```

x				
418	379	77	212	378
363	434	348	245	341
331	356	423	330	247
351	151	220	383	257
307	297	448	391	210
158	310	331	348	124
523	356	210	364	406
331	364	352	299	221
466	150	282	221	432
366	195	96	219	202

```
> n = length (table$x)
> mu = 325
> xbar = mean (table$x)
> S = sd (table$x)
> z = (xbar - mu) / (S/sqrt(n))
> pvalue = 2 * pnorm (z)
> Print (pvalue)
[1] 0.1524827
> if (Pvalue > 0.05)
  {
  Print ("ho is accepted")
  } else
  Print ("ho is rejected")
  }
[1] "ho is accepted"
```

Practical No. 08

- Import the data from Excel / .csv and perform the Chi-squared Test

1) In the past, the S.D of weight of certain 40.0. ounce packages filled by a machine was 0.25 ounce A random sample of 20 package showed a s.d of 0.32 oz. Is the apparent increase in variability significant at the 0.05 level ?

→ code :

```
> table <- read.csv(file.choose(), header =T)
> table
    S        sigma      n      alpha
  0.32       0.25      20      0.05
> chisq = function ( s, sigma , n, alpha)
. {
+ V = (table $ n - 1);
+ x2 = (table$ n * table$ s * table$ s) / (table$ sigma^2);
+ Print (paste ("x2=", x2 ));
+ Pvalue = 1- pchisq (x2, V)
+ Print (Paste ("Pvalue =", Pvalue );
+ if (Pvalue < alpha)
  {
  Print ("hypothesis reject ")
  } else
  {
  Print ("hypothesis accept")
  }
  }
> Chisq ($ S, sigma , n, alpha )
```

```
[1]  "x2 = 32.768"
[1]  " pvalue = 0.025563
[1]  " hypothesis reject "
```

2) In 200 tosses of coin 85 tail, 115 heads were observed test the hypothesis that coin is fair at significance level 0.05

→ Code :

```
> table <- read.csv (file.choose(), header =T)
> table
      O      E
      115    100
      85     100
> K = 2;
> V = K-1;
> x2 = 0;
> alpha = 0.05;
> for (i in 1:K)
  {
  a = (table$O[i]-table$E[i])^2 /(table$E[i]);
  x2 = x2+a;
> Print (paste ("x2 =", x2));
[1]  " x2 = 4.5 "

  Pvalue = 1- pchisq (x2, v);
 Print (paste ("pvalue =", pvalue));
[1]  " pvalue = 0.03389
> if (pvalue < alpha)
  {
   Print ("hypothesis reject ")
```

```
} else {
Print ("hypothesis accept")
}
[1] "hypothesis reject"
```

Roll No. 54

Practical No. 10

- Perform linear regression using R.

1) The given table purchasing power the dollar as measured by consumer prize according to the US Bureau of labor stastitic survey of Current Business.

x	1	2	3	4	5	6
y	0.581	0.565	0.556	0.544	0.530	0.512

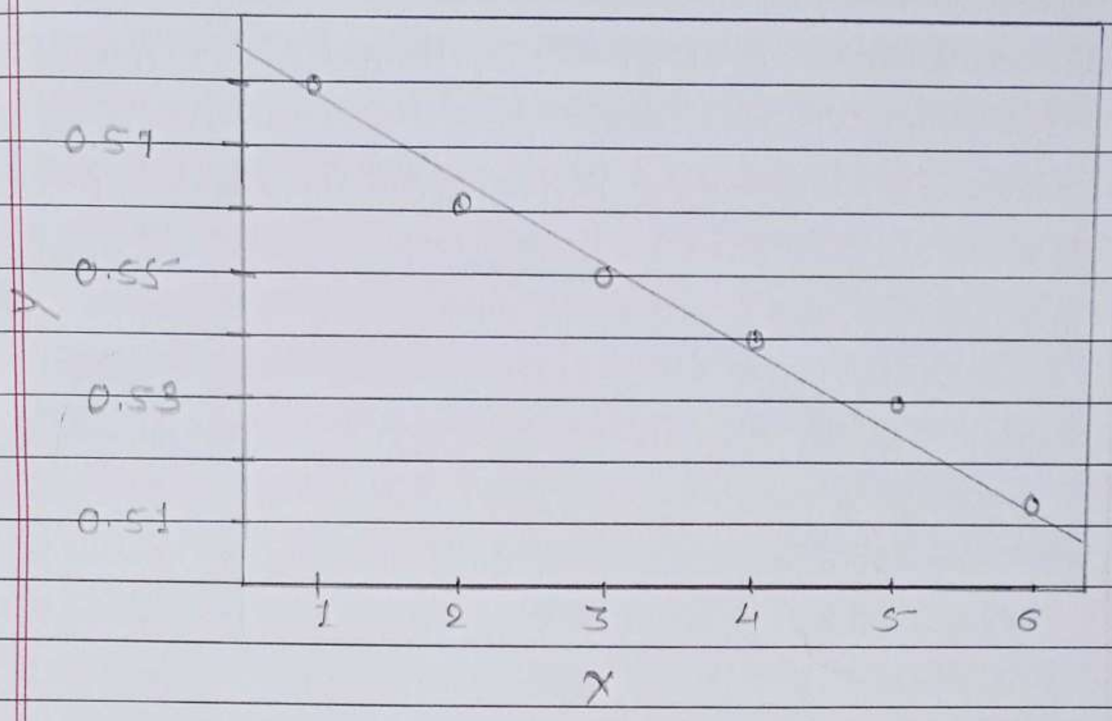
→ Code :

```
> x = c (1, 2, 3, 4, 5, 6);
> y = c (0.581, 0.565, 0.556, 0.544, 0530, 0.512);
> plot (x, y);
> abline ( lsfit (x, y));
> reg = lm(y ~ x)
> Print (reg);

Call :
lm (formula = y ~ x)

Coefficients :
(Intercept)            x
  0.5942          -0.0132

> a0 = 0.5942
> a1 = -0.0132
> Print (paste (" y=", a0, " + ", a1, "x"));
[1]  " y = 0.5942 + -0.0132 x "
```


0.57
0.55
0.53
0.51
y
1
2
3
4
5
6
x