VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

MATHS LAB REPORT

NAME Mainak Chattopadhyay
SLOT L21+L22
COURSE NAME BMAT202P
COURSE TITLE PROBABILITY AND STATISTICS LAB
FACULTY Dr. Dhivya P

EXPERIMENT 1

AIM:THE PURPOSE OF THIS EXPERIMENT IS TO LEARN THE INPUT DATA TYPES,VARIOUS ARITHMETIC OPERATIONS OF DATASET ANDIMPORTING/EXPORTING DATA IN R

AIM: THE PURPOSE OF THIS EXPERIMENT IS TO LEARN THE DIFFERENT ALIGNMENT OF DATA SET AND VARIOUS GRAPHICAL REPRESENTATIONS IN R

```
> a=10
> a
[1] 10
> b=9
> b
[1] 9
> m = a+b
> m
[1] 19
> a-b
[1] 1
> a*b
[1] 90
> a/b
[1] 1.111111
> 5/0
[1] Inf
> 0/5
[1] 0
> 0/0
[1] NaN
```

```
> sqrt(25)
[1] 5
> sqrt(100)
[1] 10
> sqrt(5)
[1] 2.236068
> log(2)
[1] 0.6931472
> log(10)
[1] 2.302585
> log10(10)
[1] 1
> cos(pi)
[1] -1
> cos(pi/2)
[1] 6.123032e-17
```

```
> round(cos(pi/2))
[1] 0
> 2^2
[1] 4
> round(44/13,digits=2)
[1] 3.38
>
```

```
> a = c(10,20,30,40)
> b = c(2,4,5,6)
> a+b
[1] 12 24 35 46
> a/b
[1] 5.000000 5.000000 6.000000 6.666667
> a*b
[1]  20  80 150 240
> a-b
[1]  8 16 25 34
```

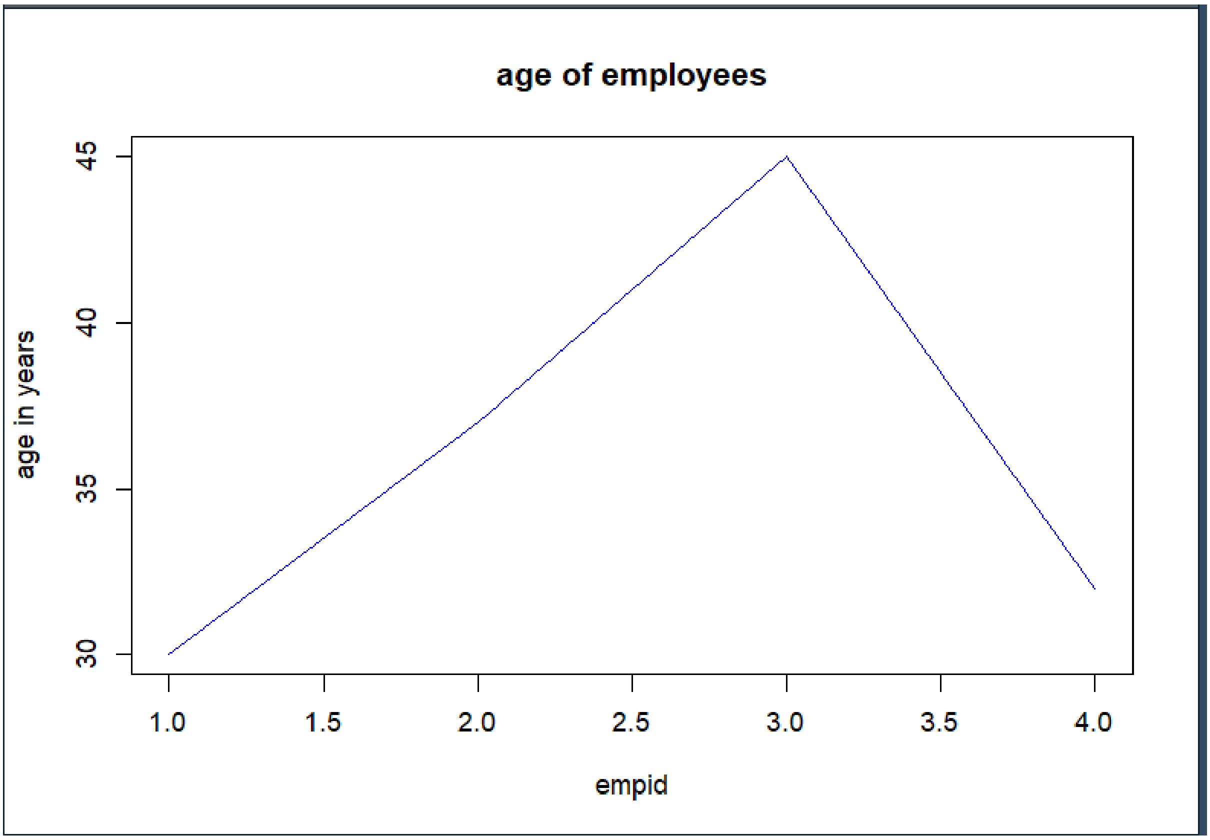
```
> a = c(10,20,30,40)
> b = c(2,4,5,6,10)
> a+b
[1] 12 24 35 46 20
Warning message:
In a + b : longer object length is not a multiple of shorter object length
> round(a/b,digits=2)
[1] 5.00 5.00 6.00 6.67 1.00
Warning message:
In a/b : longer object length is not a multiple of shorter object length
> a*b
[1]  20  80 150 240 100
Warning message:
In a * b : longer object length is not a multiple of shorter object length
> a-b
[1]  8 16 25 34  0
Warning message:
In a - b : longer object length is not a multiple of shorter object length
```

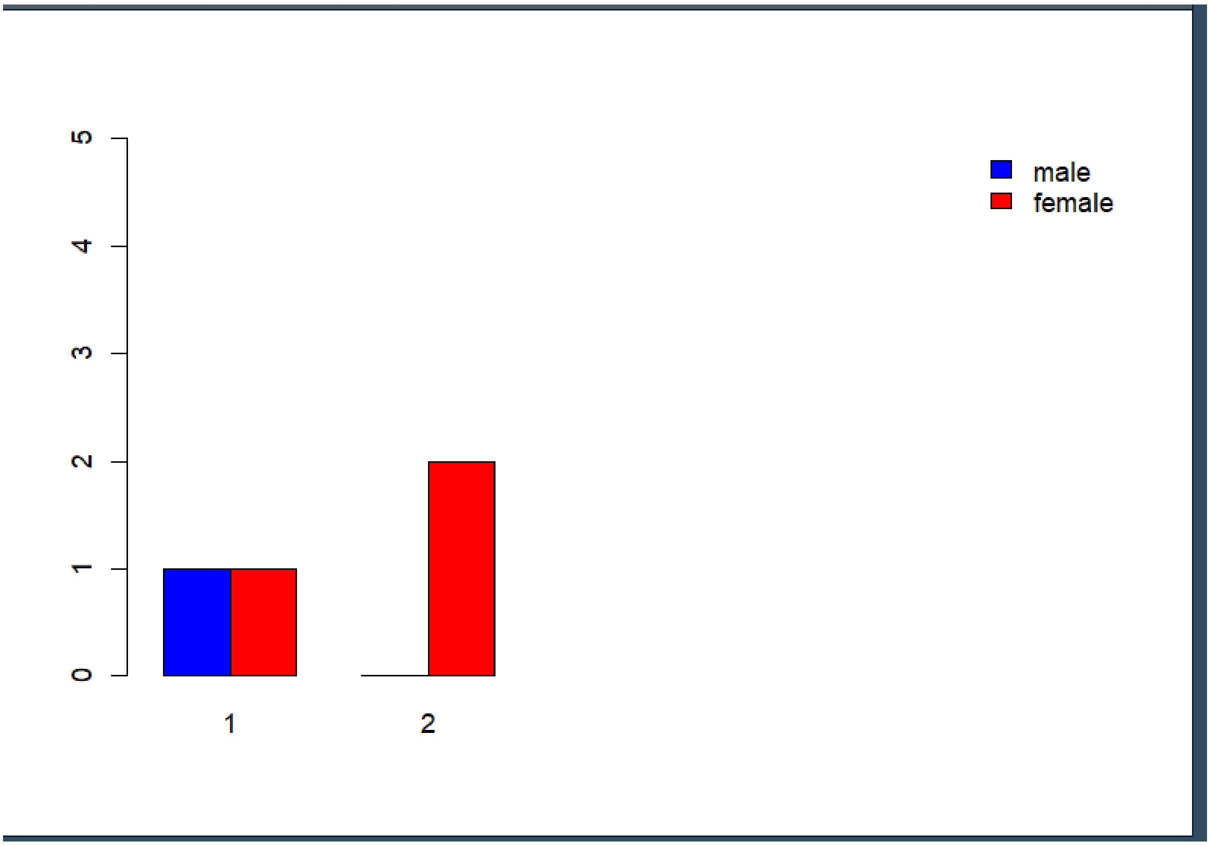
```
empid = c(1,2,3,4)
age = c(30,37,45,32)
gender = c(0,1,1,1)
gender
status = c(1,1,2,2)
status
empinfo = data.frame(empid,age,gender,status)
empinfo
empinfo$gender
empinfo$gender = factor(empinfo$gender,labels=c("male","female"))
empinfo
genderm = subset(empinfo,empinfo$gender=='male')
genderm
sexf=subset(empinfo,empinfo$gender=='female')
sexf
#creating one way table
table1 = table(empinfo$gender)
table1
table2 = table(empinfo$status)
table2
#creating 2 way table
table3 = table(empinfo$gender,empinfo$status)
table3
```

```
> empid = c(1,2,3,4)
> age = c(30,37,45,32)
> gender = c(0,1,1,1)
> gender
[1] 0 1 1 1
> status = c(1,1,2,2)
> status
[1] 1 1 2 2
> empinfo = data.frame(empid,age,gender,status)
> empinfo
  empid age gender status
1     1  30      0      1
2     2  37      1      1
3     3  45      1      2
4     4  32      1      2
> empinfo$gender
[1] 0 1 1 1
> empinfo$gender = factor(empinfo$gender,labels=c("male","female"))
> empinfo
  empid age gender status
1     1  30   male      1
2     2  37 female      1
3     3  45 female      2
4     4  32 female      2
> genderm = subset(empinfo,empinfo$gender=='male')
> genderm
  empid age gender status
1     1  30   male      1
> sexf=subset(empinfo,empinfo$gender=='female')
> sexf
  empid age gender status
2     2  37 female      1
3     3  45 female      2
4     4  32 female      2
> #creating one way table
> table1 = table(empinfo$gender)
> table1

  male female
     1      3
> table2 = table(empinfo$status)
> table2

1 2
2 2
> #creating 2 way table
> table3 = table(empinfo$gender,empinfo$status)
> table3

         1 2
  male   1 0
  female 1 2
> 
```

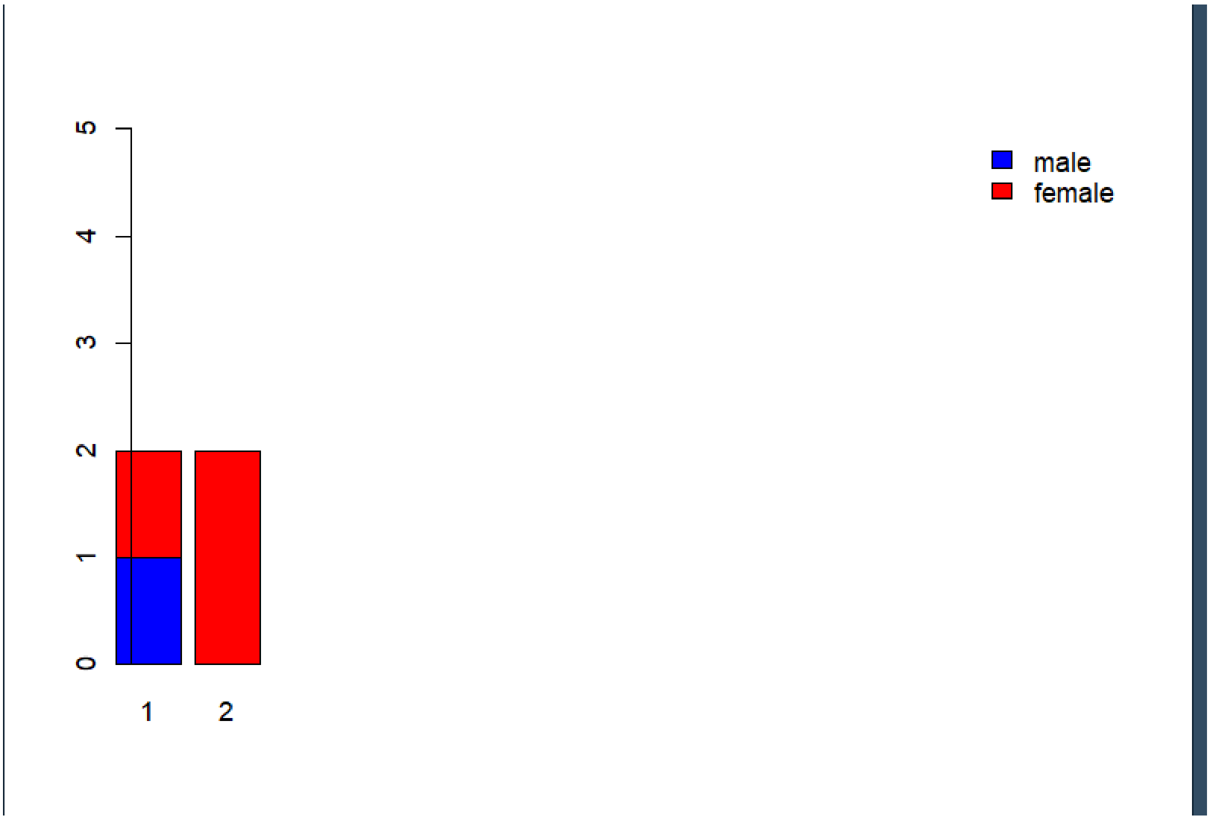
```
> empid = c(1,2,3,4)
> age = c(30,37,45,32)
> gender = c(0,1,1,1)
> gender
[1] 0 1 1 1
> status = c(1,1,2,2)
> status
[1] 1 1 2 2
> empinfo = data.frame(empid,age,gender,status)
> empinfo
  empid age gender status
1     1  30      0      1
2     2  37      1      1
3     3  45      1      2
4     4  32      1      2
> empinfo$gender
[1] 0 1 1 1
> empinfo$gender = factor(empinfo$gender,labels=c("male","female"))
> empinfo
  empid age gender status
1     1  30   male      1
2     2  37 female      1
3     3  45 female      2
4     4  32 female      2
> genderm = subset(empinfo,empinfo$gender=='male')
> genderm
  empid age gender status
1     1  30   male      1
```

```
> sexf=subset(empinfo,empinfo$gender=='female')
> sexf
  empid age gender status
2     2  37 female      1
3     3  45 female      2
4     4  32 female      2
> #creating one way table
> table1 = table(empinfo$gender)
> table1

  male female
     1      3
> table2 = table(empinfo$status)
> table2

1 2
2 2
> #creating 2 way table
> table3 = table(empinfo$gender,empinfo$status)
> table3

         1 2
  male   1 0
  female 1 2
> plot(empinfo$age,type="l",main='age of employees',xlab='empid',ylab='age in years',col='blue')
> barplot(table3,beside=T,xlim=c(1,15),ylim=c(0,5),col=c("blue","red"))
> legend("topright",legend=rownames(table3),fill=c('blue','red'),bty='n')
> 
```


age of employees
age in years
30
35
40
45
1.0
1.5
2.0
2.5
3.0
3.5
4.0
empid

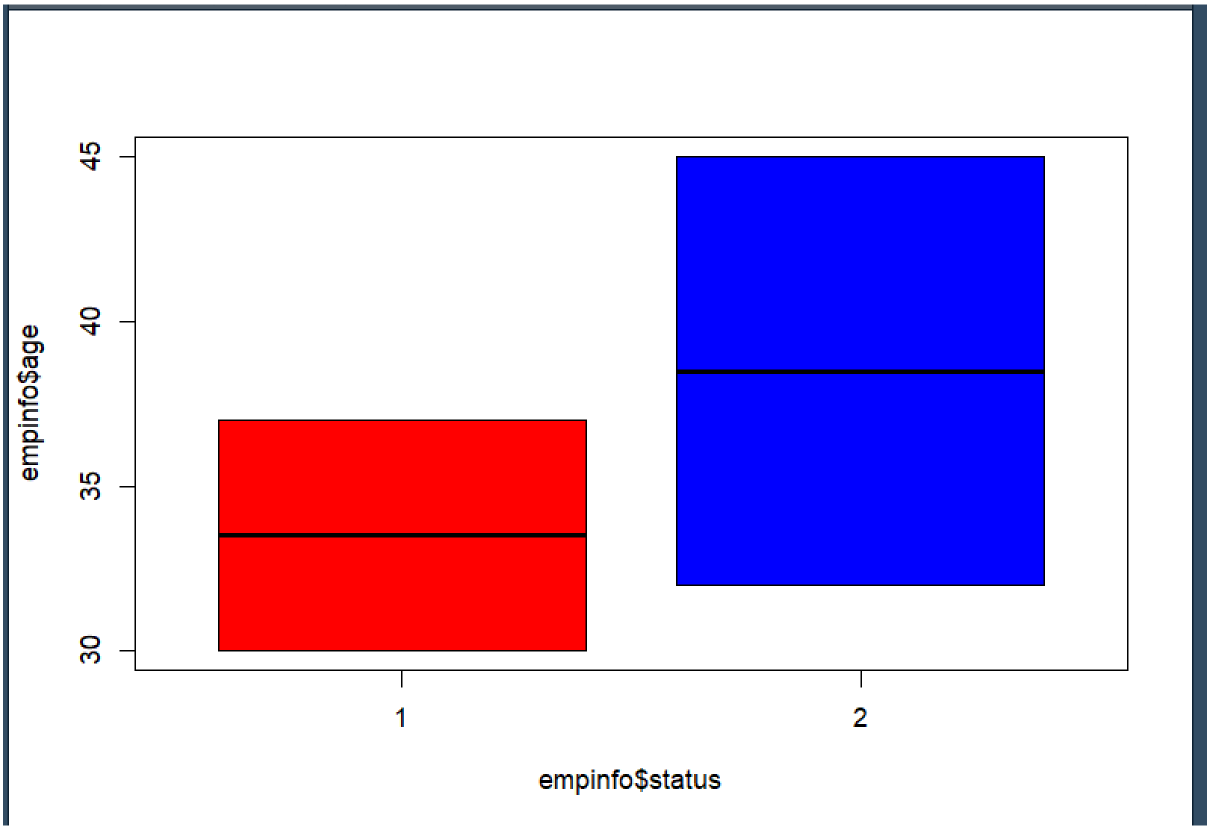

0
1
2
3
4
5
1
2
male
female

WITHOUT BESIDE=T:



boxplot(empinfo$age~empinfo$status,col=c('red','blue'))

//the line in the middle is median



EXPERIMENT 2

AIM – THE PURPOSE OF THIS EXPERIMENT IS TO LEARN THE DIFFERENT ALIGNMENT OF DATA SET AND VARIOUS GRAPHICAL REPRESENTATIONS IN R

```
> summary(table3)
Number of cases in table: 4
Number of factors: 2
Test for independence of all factors:
        Chisq = 1.3333, df = 1, p-value = 0.2482
        Chi-squared approximation may be incorrect
> x=c(18,19,19,19,19,20,20,20,20,20,21,21,21,21,22,23,24,27,30,31)
> mean(x)
[1] 21.75
> median(x)
[1] 20.5
> y=x[x<25]
> median(y)
[1] 20
> mean(y)
[1] 20.41176
> x=c(18,19,19,19,19,20,20,20,20,20,21,21,21,21,22,23,24,27,30,31)
> xr=table(x)
> mode=which(xr==max(xr))
> mode
20
 3
> x=c(0,1,2,3)
> x
[1] 0 1 2 3
> f=c(8,11,5,1)
> f
[1]  8 11  5  1
```

```
> y=rep(x,f)
> y
 [1] 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3
> mean=(sum(y))/(length(y))
> mean
[1] 0.96
> median(y)
[1] 1
> yr=table(y)
> yr
y
 0  1  2  3
 8 11  5  1
> mode=which(yr==max(yr))
> mode
1
2
> #seq function takes the start value, end value and the step value and stores them
> mid=seq(147.5,182.5,5)
> mid
[1] 147.5 152.5 157.5 162.5 167.5 172.5 177.5 182.5
> f=c(4,6,28,58,64,30,5,5)
> f
[1]   4   6  28  58  64  30   5   5
```

```
> fr.distr=data.frame(mid,f)
> fr.distr
    mid  f
1 147.5  4
2 152.5  6
3 157.5 28
4 162.5 58
5 167.5 64
6 172.5 30
7 177.5  5
8 182.5  5
> mean=((sum(mid*f))/sum(f))
> mean
[1] 165.175
> mid=seq(147.5,182.5,5)
> mid
[1] 147.5 152.5 157.5 162.5 167.5 172.5 177.5 182.5
> f1=c(4,6,28,58,64,30,5,5)
> #for median calculation
> cl=cumsum(f1)
> cl
[1]   4  10  38  96 160 190 195 200
> N=sum(f1)
> N
[1] 200
> ml=min(which(cl>=N/2)) #the serial number of median class
> ml
[1] 5
```

```
> h=5
> f1=f1[ml]  #frequency of median class
> f1
[1] 64
> c=cl[ml-1]   #cumulative frequency of median class
> c
[1] 96
> l=mid[ml]-h/2
> l
[1] 165
> median=l+(((N/2)-c)/f1)*h
> median
[1] 165.3125
> #calculating mode
> m=which(f==max(f))
> m
[1] 5
> fm=f[m]
> fm
[1] 64
> fa=f[m-1] #frequency of prev median class
> fb=f[m+1] #frquency of post median class
> m=which(f==max(f))
> m
[1] 5
> fm=f[m]
> fm
[1] 64
```

```
> fa=f[m-1] #frequency of prev median class
> fb=f[m+1] #frequency of post median class
> fa
[1] 58
> fb
[1] 30
> l=mid[m]-h/2
> mode=l+((fm-fa)/(2*fm-fa-fb))*h
> mode
[1] 165.75
>
```

EXPERIMENT 3

AIM - Applying correlation and simple linear regression model to real data set; computing and interpreting the coefficient of determination

```
R 4.3.0 · ~/
> data=cars
> data
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
23    14   80
24    15   20
25    15   26
26    15   54
27    16   32
28    16   40
29    17   32
```

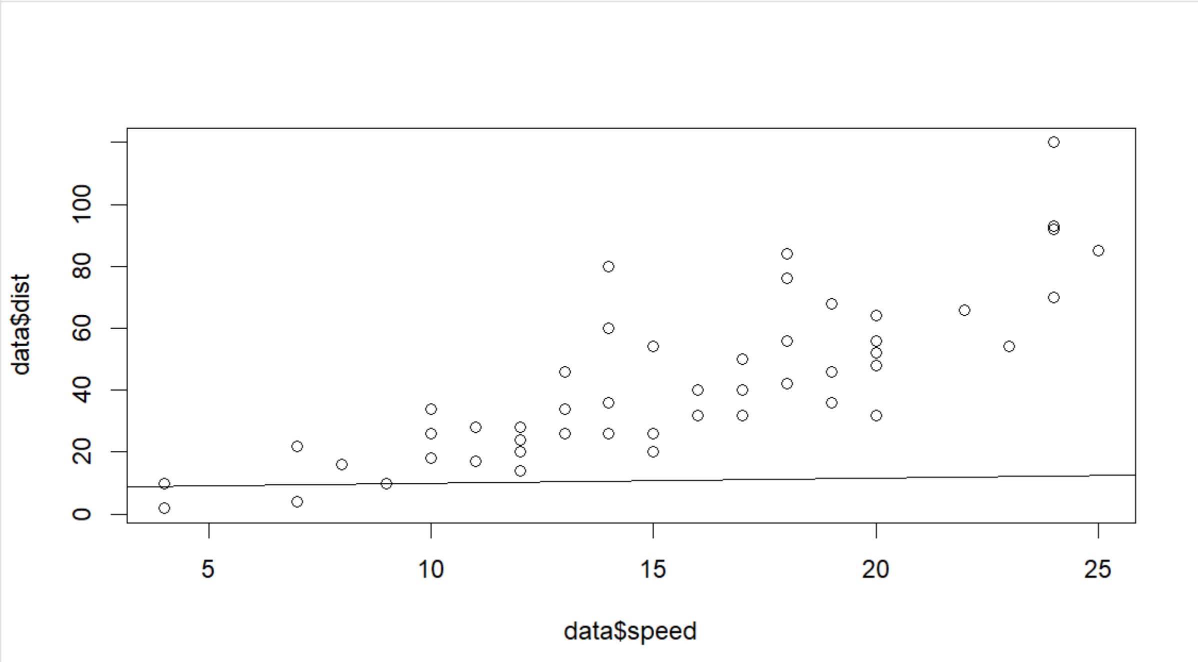
```
> summary(data)
     speed           dist       
 Min.   : 4.0   Min.   :  2.00  
 1st Qu.:12.0   1st Qu.: 26.00  
 Median :15.0   Median : 36.00  
 Mean   :15.4   Mean   : 42.98  
 3rd Qu.:19.0   3rd Qu.: 56.00  
 Max.   :25.0   Max.   :120.00  
> v1=var(data$speed)
> v1
[1] 27.95918
> v2=var(data$dist)
> v2
[1] 664.0608
> covariance=cov(data$speed,data$dist)
> covariance
[1] 109.9469
> covariance=var(data$speed,data$dist)
> covariance
[1] 109.9469
> corr=covariance/(sd(data$speed)*sd(data$dist))
> corr
[1] 0.8068949
> corr=cor(data$speed,data$dist)
> corr
[1] 0.8068949
```

```
> cor.test(data$speed,data$dist)

        Pearson's product-moment correlation

data:  data$speed and data$dist
t = 9.464, df = 48, p-value = 1.49e-12
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6816422 0.8862036
sample estimates:
      cor
0.8068949

> cor.test(data$speed,data$dist,method="pearson")

        Pearson's product-moment correlation

data:  data$speed and data$dist
t = 9.464, df = 48, p-value = 1.49e-12
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6816422 0.8862036
sample estimates:
      cor
0.8068949
```

```
> cor.test(data$speed,data$dist,method="spearman")

        Spearman's rank correlation rho

data:  data$speed and data$dist
S = 3532.8, p-value = 8.825e-14
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.8303568
```

```
R 4.3.0 · ~/
> regression1

Call:
lm(formula = data$speed ~ data$dist)

Coefficients:
(Intercept)    data$dist
     8.2839       0.1656

> abline(regression1)
> summary(regression1)

Call:
lm(formula = data$speed ~ data$dist)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5293 -2.1550  0.3615  2.4377  6.4179

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.28391    0.87438   9.474 1.44e-12 ***
data$dist    0.16557    0.01749   9.464 1.49e-12 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.156 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
R 4.3.0 · ~/
> regression2=lm(data$dist~data$speed)
> regression2

Call:
lm(formula = data$dist ~ data$speed)

Coefficients:
(Intercept)   data$speed
    -17.579        3.932

> summary(regression2)

Call:
lm(formula = data$dist ~ data$speed)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
data$speed    3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```




```
> weight=c(15,26,27,25,25.5,27,32,18,22,20,26,24)
> weight
 [1] 15.0 26.0 27.0 25.0 25.5 27.0 32.0 18.0 22.0 20.0
[11] 26.0 24.0
> bmi=c(13.35,16.12,16.74,16,13.59,15.73,15.65,13.85,16.07,12.8,13.65,14.42)
> bmi
 [1] 13.35 16.12 16.74 16.00 13.59 15.73 15.65 13.85
 [9] 16.07 12.80 13.65 14.42
> cor(weight,bmi)
[1] 0.5790235
> model1<-lm(bmi~weight)
> summary.lm(model1)

Call:
lm(formula = bmi ~ weight)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.52988 -0.75527  0.04426  0.95286  1.57397 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.73487    1.85405   5.790 0.000175 ***
weight       0.17096    0.07612   2.246 0.048524 *  
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.155 on 10 degrees of freedom
Multiple R-squared:  0.3353,	Adjusted R-squared:  0.2688 
F-statistic: 5.044 on 1 and 10 DF,  p-value: 0.04852
```

EXPERIMENT 4

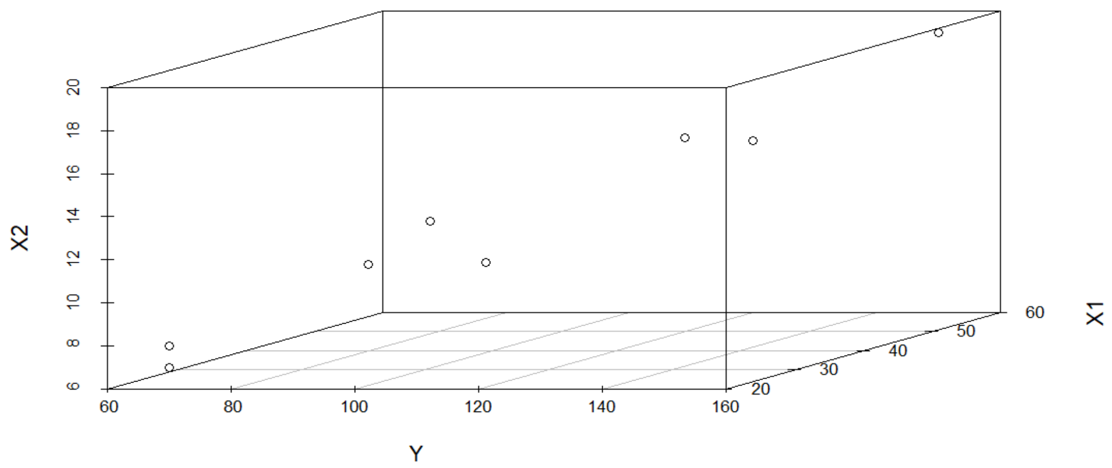
AIM - Applying multiple linear regression model to real dataset; computing and interpreting the multiple coefficients of determination

Console Terminal Background Jobs

R 4.3.0 · ~/

```
> Y=c(110,80,70,120,150,90,70,120)
> Y
[1] 110  80  70 120 150  90  70 120
> X1=c(30,40,20,50,60,40,20,60)
> X1
[1] 30 40 20 50 60 40 20 60
> X2=c(11,10,7,15,19,12,8,14)
> X2
[1] 11 10  7 15 19 12  8 14
> RegModel=lm(Y~X1+X2)
> RegModel

Call:
lm(formula = Y ~ X1 + X2)

Coefficients:
(Intercept)           X1           X2  
    16.8314      -0.2442       7.8488  
```

```
> summary(RegModel)

Call:
lm(formula = Y ~ X1 + X2)

Residuals:
      1       2       3       4       5       6       7       8 
 14.157  -5.552   3.110  -2.355  -1.308 -11.250  -4.738   7.936 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  16.8314    11.8290   1.423   0.2140  
X1           -0.2442     0.5375  -0.454   0.6687  
X2            7.8488     2.1945   3.577   0.0159 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.593 on 5 degrees of freedom
Multiple R-squared:  0.9191,	Adjusted R-squared:  0.8867 
F-statistic:  28.4 on 2 and 5 DF,  p-value: 0.001862

> library(scatterplot3d)
> scatterplot3d(Y,X1,X2)
> 
```



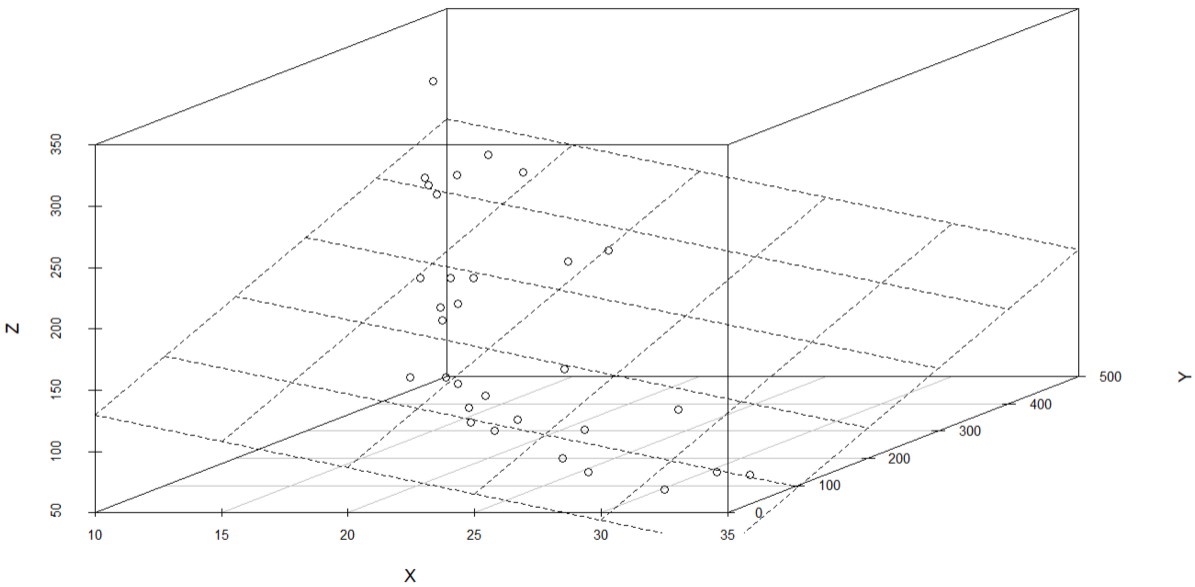
Console Terminal Background Jobs

R 4.3.0 · ~/

```
> data=mtcars
> data
                     mpg cyl  disp  hp drat    wt  qsec vs am
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1
Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1
Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1
                    gear carb
```

```
> X=mtcars$mpg
> X
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4
[13] 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3
[25] 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
> Y=mtcars$disp
> Y
 [1] 160.0 160.0 108.0 258.0 360.0 225.0 360.0 146.7 140.8 167.6
[11] 167.6 275.8 275.8 275.8 472.0 460.0 440.0  78.7  75.7  71.1
[21] 120.1 318.0 304.0 350.0 400.0  79.0 120.3  95.1 351.0 145.0
[31] 301.0 121.0
> Z=mtcars$hp
> Z
 [1] 110 110  93 110 175 105 245  62  95 123 123 180 180 180 205
[16] 215 230  66  52  65  97 150 150 245 175  66  91 113 264 175
[31] 335 109
> RegModel<- lm(Z~X+Y)
> RegModel

Call:
lm(formula = Z ~ X + Y)

Coefficients:
(Intercept)            X            Y
   172.2204      -4.2732       0.2614
```

```
> summary(RegModel)

Call:
lm(formula = Z ~ X + Y)

Residuals:
   Min     1Q Median     3Q    Max
-48.70 -17.67 -10.16  10.12 148.19

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 172.2204    69.9014   2.464   0.0199 *
X            -4.2732     2.3027  -1.856   0.0737 .
Y             0.2614     0.1120   2.335   0.0267 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 41.01 on 29 degrees of freedom
Multiple R-squared:  0.6653,    Adjusted R-squared:  0.6423
F-statistic: 28.83 on 2 and 29 DF,  p-value: 1.279e-07

> library(scatterplot3d)
> graph=scatterplot3d(X,Y,Z)
> graph$plane3d(RegModel)
>
```



Z
350
300
250
200
150
100
50
10
15
20
25
30
35
X
0
100
200
300
400
500
Y

EXPERIMENT 5

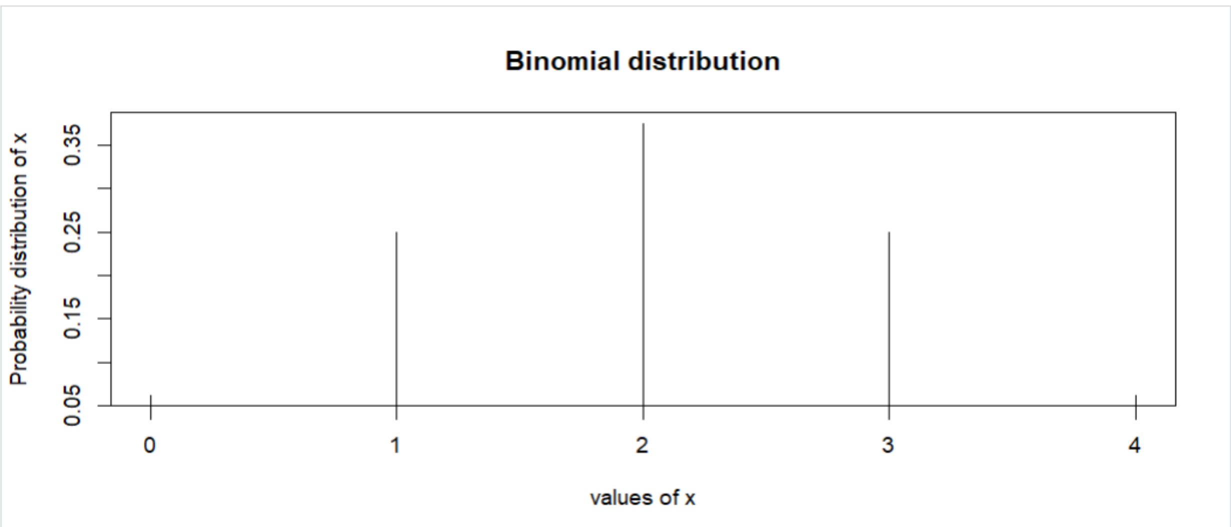
AIM - Fitting the probability distributions: Binomial distribution

Console Terminal × Background Jobs ×

R 4.3.0 · ~/

```
> n=4
> n
[1] 4
> p=0.5
> p
[1] 0.5
> dbinom(2,n,p)
[1] 0.375
> sum(dbinom(2:4,n,p))
[1] 0.6875
> 1-pbinom(1,n,p)
[1] 0.6875
> sum(dbinom(0:2,n,p))
[1] 0.6875
> pbinom(2,n,p)
[1] 0.6875
> x=0:n
> px=dbinom(x,n,p)
> Ex=weighted.mean(x,px)
> Ex
[1] 2
> Varx=weighted.mean(x*x,px)-(weighted.mean(x ,px))^2
> Varx
[1] 1
> plot(x,px,type="h",xlab="values of x",ylab="Probability distribution of x",main="Binomial distribution")
> n=4
> n
[1] 4
> p=0.5
> p
[1] 0.5
> dbinom(2,n,p)
[1] 0.375
> sum(dbinom(2:4,n,p))
[1] 0.6875
> 1-pbinom(1,n,p)
[1] 0.6875
> sum(dbinom(0:2,n,p))
[1] 0.6875
> pbinom(2,n,p)
[1] 0.6875
```

```
> x=0:n
> px=dbinom(x,n,p)
> Ex=weighted.mean(x,px)
> Ex
[1] 2
> Varx=weighted.mean(x*x,px)-(weighted.mean(x ,px))^2
> Varx
[1] 1
> plot(x,px,type="h",xlab="values of x",ylab="Probability distribution of x",main="Binomial distribution")
>
```

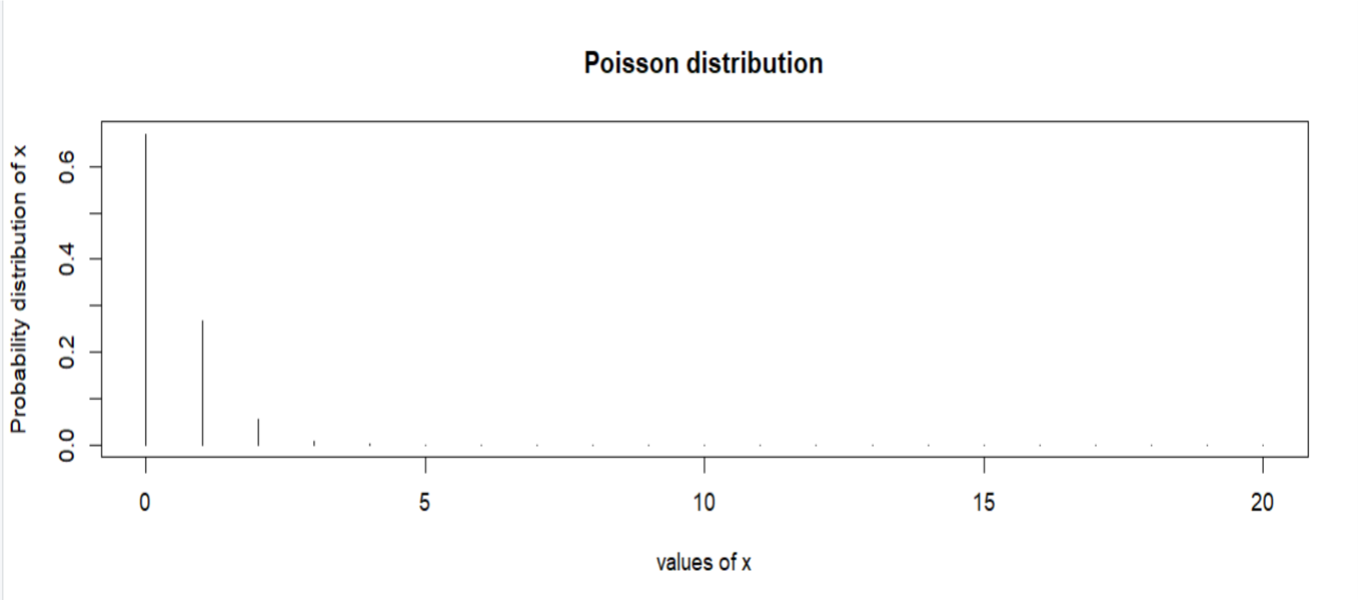


EXPERIMENT 6

AIM - To understand Poisson distribution and Normal distribution using R functions

Poisson Distribution

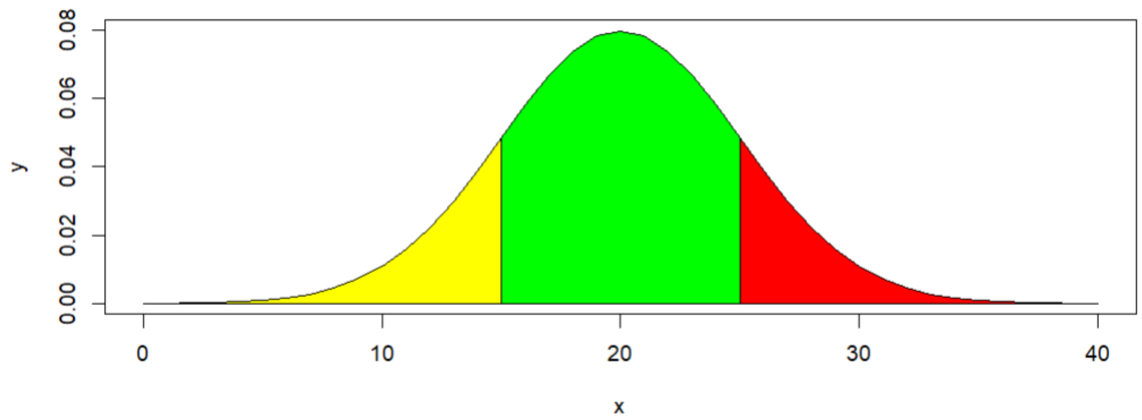
```
Console  Terminal  Background Jobs
R 4.3.0 · ~/
> m=20
> m
[1] 20
> ps=0.02
> lambda=m*ps
> lambda
[1] 0.4
> p1=sum(dpois(2:m,lambda))
> p1
[1] 0.06155194
> round(1000*p1)
[1] 62
> p2=dpois(2,lambda)
> p2
[1] 0.0536256
> round(1000*p2)
[1] 54
> p3=sum(dpois(0:2, lambda))
> p3
[1] 0.9920737
> round(1000*p3)
[1] 992
> x1=0:m
> px1=dpois(x1,lambda)
> plot(x1,px1,type="h",xlab="values of x",ylab="Probability distribution of x",main="Poisson distribution")
> Ex1=weighted.mean(x1,px1)
> Ex1
[1] 0.4
> Varx1=weighted.mean(x1*x1,px1)-(weighted.mean(x1 ,px1))^2
> Varx1
[1] 0.4
> 
```



Normal Distribution

```
R 4.3.0 · ~/
> x=seq(0,40)
> x
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
[18] 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
[35] 34 35 36 37 38 39 40
> y=dnorm(x,mean=20,sd=5)
> y
 [1] 2.676605e-05 5.838939e-05 1.223804e-04 2.464438e-04
 [5] 4.768176e-04 8.863697e-04 1.583090e-03 2.716594e-03
 [9] 4.478906e-03 7.094919e-03 1.079819e-02 1.579003e-02
[13] 2.218417e-02 2.994549e-02 3.883721e-02 4.839414e-02
[17] 5.793831e-02 6.664492e-02 7.365403e-02 7.820854e-02
[21] 7.978846e-02 7.820854e-02 7.365403e-02 6.664492e-02
[25] 5.793831e-02 4.839414e-02 3.883721e-02 2.994549e-02
[29] 2.218417e-02 1.579003e-02 1.079819e-02 7.094919e-03
[33] 4.478906e-03 2.716594e-03 1.583090e-03 8.863697e-04
[37] 4.768176e-04 2.464438e-04 1.223804e-04 5.838939e-05
[41] 2.676605e-05
> plot(x,y,type='l')
> p1=pnorm(15,mean=20,sd=5)
> p1
[1] 0.1586553
> x2=seq(0,15)
> x2
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
> y2=dnorm(x2,mean=20,sd=5)
> y2
 [1] 2.676605e-05 5.838939e-05 1.223804e-04 2.464438e-04
 [5] 4.768176e-04 8.863697e-04 1.583090e-03 2.716594e-03
 [9] 4.478906e-03 7.094919e-03 1.079819e-02 1.579003e-02
[13] 2.218417e-02 2.994549e-02 3.883721e-02 4.839414e-02
> polygon(c(0,x2,15),c(0,y2,0),col='yellow')
> p2=pnorm(40,mean=20,sd=5)-pnorm(25,mean=20,sd=5)
> p2
[1] 0.1586236
> x1=seq(25,40)
> x1
 [1] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

```
> y1=dnorm(x1,mean=20,sd=5)
> y1
 [1] 4.839414e-02 3.883721e-02 2.994549e-02 2.218417e-02
 [5] 1.579003e-02 1.079819e-02 7.094919e-03 4.478906e-03
 [9] 2.716594e-03 1.583090e-03 8.863697e-04 4.768176e-04
[13] 2.464438e-04 1.223804e-04 5.838939e-05 2.676605e-05
> polygon(c(25,x1,40),c(0,y1,0),col='red')
> p3=pnorm(25,mean=20,sd=5)-pnorm(15,mean=20,sd=5)
> p3
[1] 0.6826895
> x3=seq(15,25)
> x3
 [1] 15 16 17 18 19 20 21 22 23 24 25
> y3=dnorm(x3,mean=20,sd=5)
> y3
 [1] 0.04839414 0.05793831 0.06664492 0.07365403
 [5] 0.07820854 0.07978846 0.07820854 0.07365403
 [9] 0.06664492 0.05793831 0.04839414
> polygon(c(15,x3,25),c(0,y3,0),col='green')
> data.frame(p1,p2,p3)
         p1        p2        p3
1 0.1586553 0.1586236 0.6826895
>
```



y
0.00
0.02
0.04
0.06
0.08
0
10
20
30
40
x

EXPERIMENT 7

AIM - To understand the testing of hypothesis for large sample tests using R functions

```
> xbar=14.6
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
3: In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
> xbar
[1] 14.6
> mu0=15.4
> mu0
[1] 15.4
> sigma=2.5
> sigma
[1] 2.5
> n=35
> n
[1] 35
> z=(xbar-mu0)/(sigma/sqrt(n))
> z
[1] -1.893146
> alpha=0.05
> alpha
[1] 0.05
> zhalfalpha=qnorm(1-(alpha/2))
> zhalfalpha
[1] 1.959964
> c(-zhalfalpha,zhalfalpha)
[1] -1.959964  1.959964
> pval=2*pnorm(z)
> pval
[1] 0.05833852
> if(pval>alpha){print("Accept Null hypothesis")} else{print("Reject Null
+ hypothesis")}
[1] "Accept Null hypothesis"
> 
```

Testing of Hypothesis - Large Sample proportion Test

```
Console   Terminal ×   Background Jobs ×
R 4.3.0 · ~/
> n=640
> n
[1] 640
> Sprop=63/n
> Sprop
[1] 0.0984375
> Pprop=0.1726
> Pprop
[1] 0.1726
> q=1-Pprop
> q
[1] 0.8274
> z=(Sprop-Pprop)/sqrt(Pprop*q/n)
> z
[1] -4.964736
> E=qnorm(.975)
> c(-E,E)
[1] -1.959964  1.959964
> Sprop+c(-E,E)*sqrt(Pprop*(1-Pprop)/n)
[1] 0.06915985 0.12771515
> if(z>-E && z<E){print("Hospital is not efficient")} else{print("Hospital is efficient")}
[1] "Hospital is efficient"
> 
```

EXPERIMENT 8

AIM - Testing of hypothesis for two sample means and proportion from real time problems

Console Terminal Background Jobs

R 4.3.0 · ~/

```
> xbar=20
> xbar
[1] 20
> ybar=15
> ybar
[1] 15
> sigma=4
> sigma
[1] 4
> n1=500
> n1
[1] 500
> n2=400
> n2
[1] 400
> z=(xbar-ybar)/(sigma*sqrt((1/n1)+(1/n2)))
> z
[1] 18.6339
> alpha=0.05
> alpha
[1] 0.05
> zalpha=qnorm(1-(alpha/2))
> zalpha
[1] 1.959964
> if(z<=zalpha){print("Accept Null hypothesis")} else{print("Reject Null hypothesis")}
[1] "Reject Null hypothesis"
> 
```

Testing of Hypothesis - Two Sample proportion Test

Console Terminal Background Jobs

R 4.3.0 · ~/

```
> p1=0.20
> p1
[1] 0.2
> p2=0.185
> p2
[1] 0.185
> n1=900
> n1
[1] 900
> n2=1600
> n2
[1] 1600
> P=(n1*p1+n2*p2)/(n1+n2)
> P
[1] 0.1904
> Q=1-P
> z=(p1-p2)/sqrt(P*Q*sqrt((1/n1)+(1/n2)))
> z
[1] 0.1871665
> alpha=0.05
> alpha
[1] 0.05
> zalpha=qnorm(1-(alpha/2))
> zalpha
[1] 1.959964
> if(z<=zalpha){print("Accept Null hypothesis")} else{print("Reject Null hypothesis")}
[1] "Accept Null hypothesis"
> 
```

EXPERIMENT 9

AIM - Applying the t-test for independent and dependent samples

```
> sample1=c(19,17,15,21,16,18,16,14)
> sample1
[1] 19 17 15 21 16 18 16 14
> sample2=c(15,14,15,19,15,18,16,20)
> sample2
[1] 15 14 15 19 15 18 16 20
> t=t.test(sample1,sample2)
> t

        Welch Two Sample t-test

data:  sample1 and sample2
t = 0.44721, df = 13.989, p-value = 0.6616
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.898128  2.898128
sample estimates:
mean of x mean of y 
     17.0      16.5 

> cv=t$statistic
> cv
        t 
0.4472136 
> tv=qt(0.975,14)
> tv
[1] 2.144787
> if(cv <= tv){print("Accept Ho")} else{print("Reject Ho")}
[1] "Accept Ho"
> 
```

```
> test1=c(19,17,15,21,16,18,16,14,19,20)
> test1
 [1] 19 17 15 21 16 18 16 14 19 20
> test2=c(15,14,15,19,15,18,16,20,22,19)
> test2
 [1] 15 14 15 19 15 18 16 20 22 19
> t=t.test(sample1,sample2,paired=TRUE)
> t

        Paired t-test

data:  sample1 and sample2
t = 0.46771, df = 7, p-value = 0.6542
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -2.02789  3.02789
sample estimates:
mean difference 
            0.5 

> alpha=0.05
> tv=t$p.value
> tv
[1] 0.6542055
> if(tv >alpha){print("Accept Ho")} else{print("Reject Ho")}
[1] "Accept Ho"
> 
```

R 4.3.0 · ~/

```
> sample1=c(19,17,15,21,16,18,16,14)
> sample1
[1] 19 17 15 21 16 18 16 14
> sample2=c(15,14,15,19,15,18,16,20)
> sample2
[1] 15 14 15 19 15 18 16 20
> f=var.test(sample1,sample2)
> f

	F test to compare two variances

data:  sample1 and sample2
F = 1.0588, num df = 7, denom df = 7, p-value = 0.9418
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2119805 5.2887274
sample estimates:
ratio of variances
          1.058824

> cv=f$statistic
> cv
       F
1.058824
> tv=qf(0.95,7,7)
> tv
[1] 3.787044
> if(cv <= tv){print("Accept Ho")} else{print("Reject Ho")}
[1] "Accept Ho"
>
```

EXPERIMENT 10

AIM - To understand Chi-square test for goodness of fit and independent of attributes using R

```
R 4.3.0 · ~/
> # Problem : 1
> # Goodness of fit
> # Number of coins
> n=5
> n
[1] 5
> ## [1] 5
> # level of significance
> alpha=0.05
> alpha
[1] 0.05
> ## [1] 0.05
> N=256
> # Total number of tosses
> N
[1] 256
> ## [1] 256
> P =0.5
> # probability of getting head
> P
[1] 0.5
> ## [1] 0.5
> x =c(0:n);x
[1] 0 1 2 3 4 5
> ## [1]0 1 2 3 4 5
> obf =c(5,35,75,84,45,12)
> # observed frequencies
> obf
[1]  5 35 75 84 45 12
> ## [1]  5 35 75 84 45 12
> exf =(dbinom(x,n,P)*256)
> # expected frequencies
> exf
[1]  8 40 80 80 40  8
> ## [1]  8 40 80 80 40  8
> # check the condition if the observed and expected frequencies sum are equal
> sum(obf)
[1] 256
> ##[1] 256
> sum(exf)
[1] 256
> ## [1] 256
```

```
> # output using Chisq-distribution
> chisq<-sum((obf-exf)^2/exf)
> cv =chisq;cv
[1] 4.8875
> ## [1] 4.8875
> # critical value using Chisq-distribution
> tv =qchisq(1-alpha,n);tv
[1] 11.0705
> ## [1] 11.0705
> # Hypothesis conclusion
> if(cv <=tv){print("Accept H0/Fit is good")} else{print("Reject H0/Fit is not good")}
[1] "Accept H0/Fit is good"
> ## [1] "Accept H0/Fit is good"
> 
```

Console Terminal Background Jobs

R 4.3.0 · ~/

```
> # Problem : 2
> # Independent of attributes
> # Input the data
> data<-matrix(c(69,51,81,20,35,44),ncol=2,byrow=T)
> data
     [,1] [,2]
[1,]   69   51
[2,]   81   20
[3,]   35   44
> ##      [,1] [,2]
> ## [1,]   69   51
> ##[2,]   81   20
> ## [3,]   35   44
> # number of data
> alpha=0.05
> alpha
[1] 0.05
> l=length(data);l
[1] 6
> ## [1] 6
> # output by Chisq-distribution
> cv=chisq.test(data)
> cv

	Pearson's Chi-squared test

data:  data
X-squared = 25.629, df = 2, p-value = 2.721e-06

> ##
> ##  Pearson's Chi-squared test
> ##
> ## data:  data
> ## X-squared = 25.629, df = 2, p-value = 2.721e-06
> # p-value
> cv=cv$p.value
> cv
[1] 2.72114e-06
> ## [1] 2.72114e-06
> # Hypothesis conclusion
> if(cv >alpha){print("Attributes are independent")} else{print("Attributes are not independent")}
[1] "Attributes are not independent"
> ## [1] "Attributes are not independent"
> 
```