VIT®

Vellore Institute of Technology

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

PROJECT:

PROBABILITY AND STATISTIC'S

(LAB)

COURSE CODE:BMAT202P

TEAM DETAILS:

- PREMCHAND YANDAPALLI(21BEC1049)
- SURYA CHAITANYA(21BEC1583)
- KURAKULA SRIRANGA MANIDHAR(21BEC1053)
- B.V.L.N.TRINATH(21BEC1054)
- PASUPULATI RAVI CHARUSHAYAA(21BLC1214)

Summery:

The complete data analysis of the of the online website based on the given data by the client (ram,rom,review,...etc).we have sorted the data and given a clear analysis based on given mobile features taking ram is the main feature of the given data set. We have found all the average(mean) sales of the mobiles which have various features and plotted them graphically. We also done many operation's i.e Variance ,correlation ,regression ,normal distribution

Link's:

R-Code of the project:

https://drive.google.com/file/d/1T210PS33DkuH-1JxbBV_H_1yi3qdJSxV/view?usp=sharing

Dataset:

https://drive.google.com/file/d/1CWr7xNYd5b89ExryaNql0w2GRgciN1QV/view?usp=sharing

In the coming slides &code:

L=no.of mobiles
L1=mean of the ratings(specified ram mobile)
L2=SD if the rating(specified ram mobile)
L3=mean of the reviews(specified ram mobile)
L4=SD if the reviews(specified ram mobile)
L5=mean of the list prize(specified ram mobile)
L6=SD if the list prize(specified ram mobile)
L7=mean of the selling prize(specified ram mobile)
L8=SD if the selling prize(specified ram mobile)
L9=mean of the stars(specified ram mobile)
L10=SD if the stars(specified ram mobile)

a1= ram
r=rating
r1=reviews
lp=listprize
sp=sellingprize
s=stars

R- CODE:

```
d=read.csv(file.choose())
d
a1=d$RAM
a1
r=d$Ratings
r1=d$Reviews
lp=d$List.Price
sp=d$Sales.Price
s=d$Stars
gb=c(1,2,3,4,6,8,10,12,16,20,24,32,64,512)
#============
l <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
      c=c+1
    }
  }
  l<-append(l,c)

}
l
gb
z=data.frame(gb,l)
```

```
z

mean(a1)
#------mean rating
l1 <-c()
y <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
      y <- append(y,r[i])
    }
  }
  l1<-append(l1,mean(y))

}
l1
z=data.frame(gb,l,l1)
z
#----------------sd
l2 <-c()
y <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
```

```
  {
    if(a1[i]==j)
    {
      y <- append(y,r[i])
    }
  }
  l2<-append(l2,sd(y))

}
l2
z=data.frame(gb,l,l1,l2)
z
#-------------mean&sd review
l3 <-c()
l4 <-c()
y1 <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
      y1 <- append(y1,r1[i])
    }
  }
  l3<-append(l3,mean(y1))
  l4<-append(l4,sd(y1))

}
```

```
l3
l4
z=data.frame(gb,l,l1,l2,l3,l4)
z
#------------mean&sd lp
l5 <-c()
l6 <-c()
y1 <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
      y1 <- append(y1,lp[i])
    }
  }
  l5<-append(l5,mean(y1))
  l6<-append(l6,sd(y1))

}
l5
l6
z=data.frame(gb,l,l1,l2,l3,l4,l5,l6)
z
#------------mean&sd sp
l7 <-c()
l8 <-c()
y1 <-c()
```

```
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
      y1 <- append(y1,sp[i])
    }
  }
  l7<-append(l7,mean(y1))
  l8<-append(l8,sd(y1))

}
l5
l6
z=data.frame(gb,l,l1,l2,l3,l4,l5,l6,l7,l8)
z
#------------mean&sd s
l9 <-c()
l10 <-c()
y1 <-c()
p=0
c=0
for (j in gb) {
  c=0
  for(i in 1:1100)
  {
    if(a1[i]==j)
    {
```

```
        y1 <- append(y1,s[i])
      }
    }
    19<-append(19,mean(y1))
    110<-append(110,sd(y1))

}
19
110
z=data.frame(gb,1,11,12,13,14,15,16,17,18,19,110)
z
```

SORTED DATA LIST:

```
> z=data.frame(gb,1,11,12,13,14,15,16,17,18,19,110)
> z
    gb   1        11        12        13        14        15        16       17        18       19        110
1    1   5 393794.80 260250.22 29570.800 18721.518 12799.000  4024.922 12099.00  3612.478 4.460000 0.05477226
2    2  73  50735.09 113216.21  4275.564  8698.919 17513.821 11777.651 15714.35 10916.732 4.328205 0.37724133
3    3  99  71646.74 128930.35  5666.633 10076.280 15514.192 10658.820 13632.53  9754.344 4.293785 0.38673432
4    4 297  59172.79 102570.26  4992.127  8483.230 13427.593 11129.028 12075.09  9653.695 4.280380 0.27305929
5    6 172  47616.08  91230.96  4024.745  7556.758 11976.588 11044.409 10919.78  9519.560 4.242260 0.27227523
6    8 100  43486.62  87755.75  3702.535  7348.913 11116.265 11127.593 10320.56  9596.667 4.221850 0.27644398
7   10   8  44289.96  89841.38  3785.190  7604.220 11128.086 11099.991 10302.57  9572.234 4.221618 0.27632964
8   12   7  43904.97  89516.63  3752.992  7576.604 11075.302 11084.624 10246.24  9552.132 4.219054 0.27666653
9   16   3  44553.37  90679.05  3818.139  7715.866 11090.698 11068.841 10231.92  9539.533 4.219241 0.27662817
10  20   2  44661.73  90690.96  3829.952  7720.899 11083.931 11055.155 10232.49  9527.062 4.219321 0.27637723
11  24  16  44118.77  90019.19  3784.288  7667.195 11020.324 11126.823 10156.67  9496.592 4.217775 0.27541827
12  32 260  39080.61  85812.17  3431.980  7563.580 10042.256 11673.649 10315.89 12207.456 4.187044 0.29245244
13  64  40  38034.16  84741.82  3345.101  7483.498  9965.151 11674.747 10122.82 12117.024 4.182163 0.29218267
14 512  14  37816.04  84465.87  3320.347  7447.764  9873.339 11638.493 10113.65 12088.364 4.181569 0.29125410
>
```

PLOTS FOR SORTED DATA:

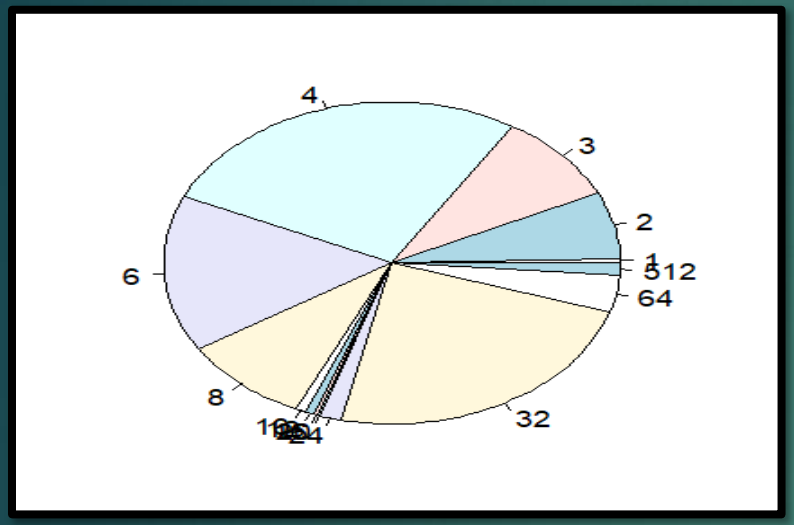
R-CODE:-

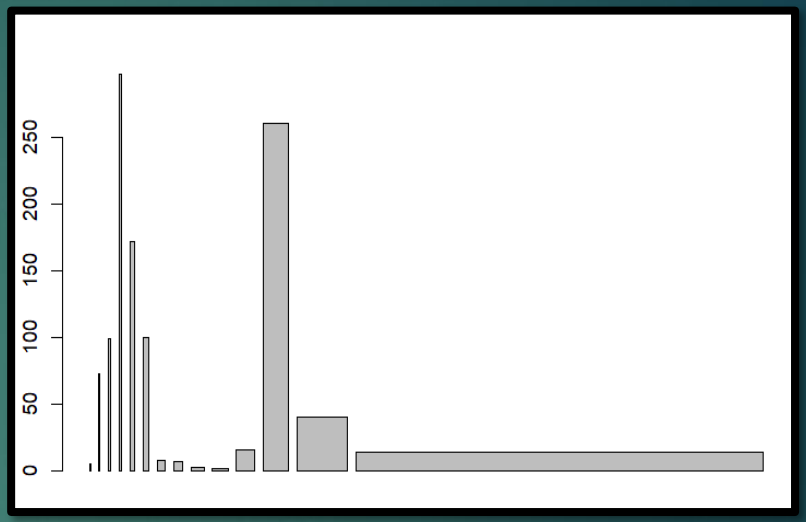
```
pie(l,gb)
barplot(l,gb)
boxplot(l,gb)
scatterplot3d(l,gb)

pie(l1,gb)
barplot(l1,gb)
boxplot(l1,gb)
scatterplot3d(l1,gb)

pie(l2,gb)
barplot(l2,gb)
boxplot(l2,gb)
scatterplot3d(l2,gb)

pie(l3,gb)
barplot(l3,gb)
boxplot(l3,gb)
scatterplot3d(l3,gb)

pie(l4,gb)
barplot(l4,gb)
boxplot(l4,gb)
scatterplot3d(l4,gb)

pie(l5,gb)
```

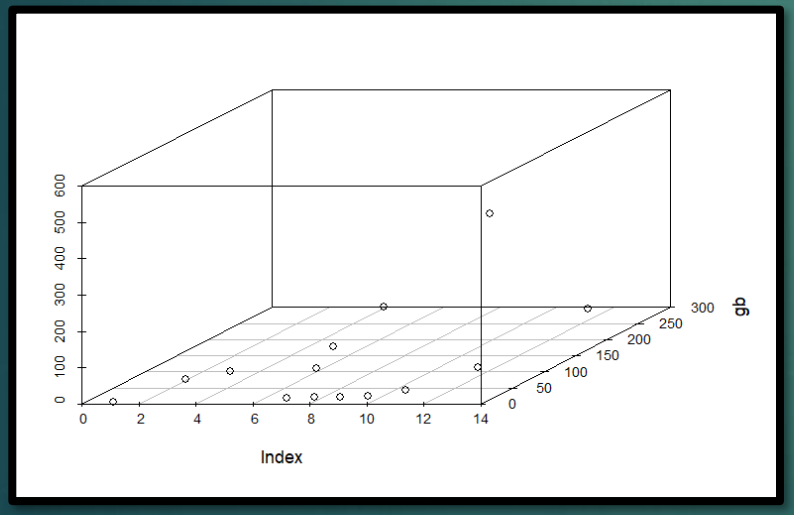
```
barplot(l5,gb)
boxplot(l5,gb)
scatterplot3d(l5,gb)

pie(l6,gb)
barplot(l6,gb)
boxplot(l6,gb)
scatterplot3d(l6,gb)

pie(l7,gb)
barplot(l7,gb)
boxplot(l7,gb)
scatterplot3d(l7,gb)

pie(l8,gb)
barplot(l8,gb)
boxplot(l8,gb)
scatterplot3d(l8,gb)

pie(l9,gb)
barplot(l9,gb)
boxplot(l9,gb)
scatterplot3d(l9,gb)

pie(l10,gb)
barplot(l10,gb)
boxplot(l10,gb)
scatterplot3d(l10,gb)
```

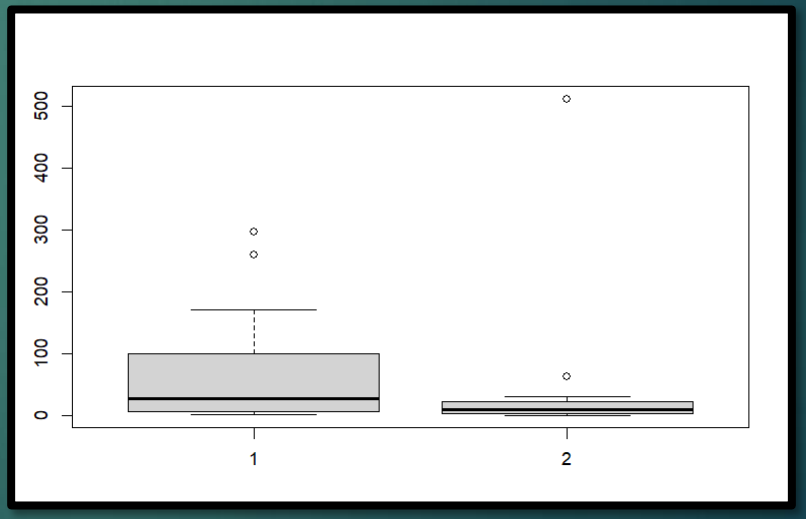
PLOTS OF MEAN AND STANDARD DEVIATION FOR SORTED DATA:-

PLOTS OF TYPES OF RAM & NO OF MOBILES:-

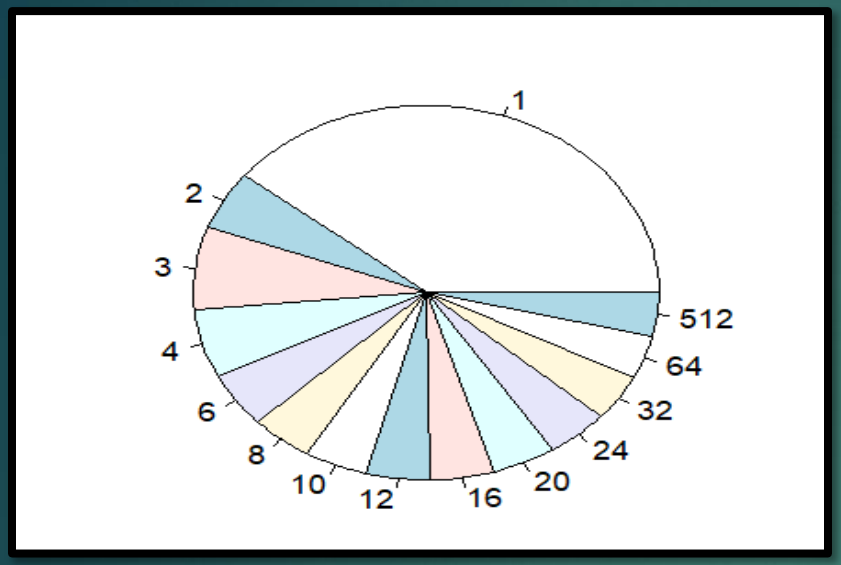


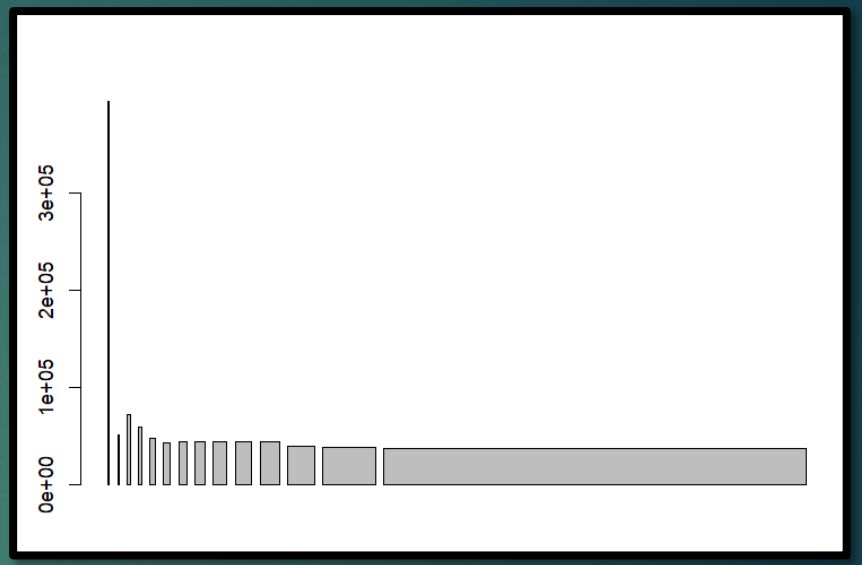


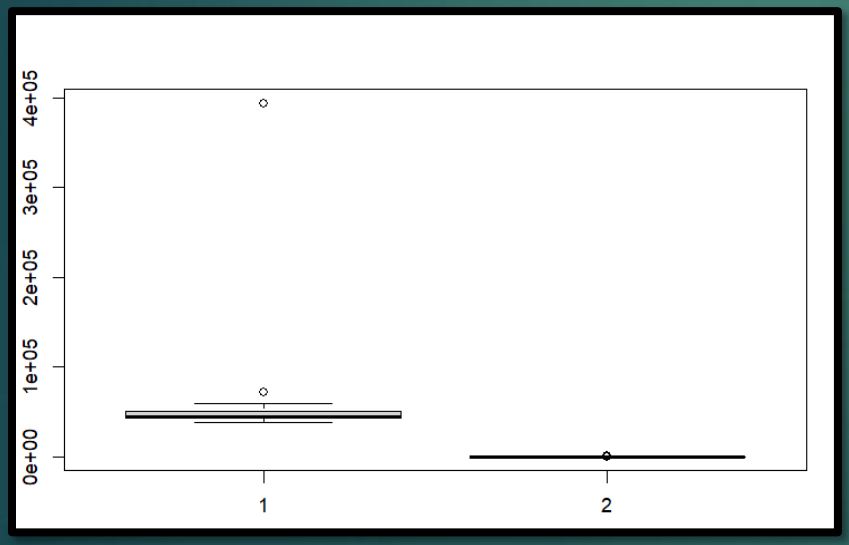


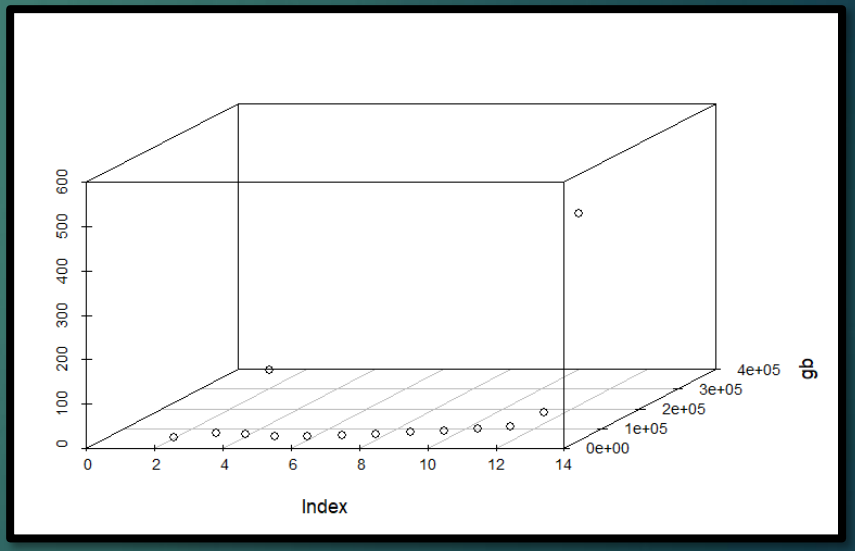


PLOTS OF TYPES OF RAM & MEANS OF RATINGS:-

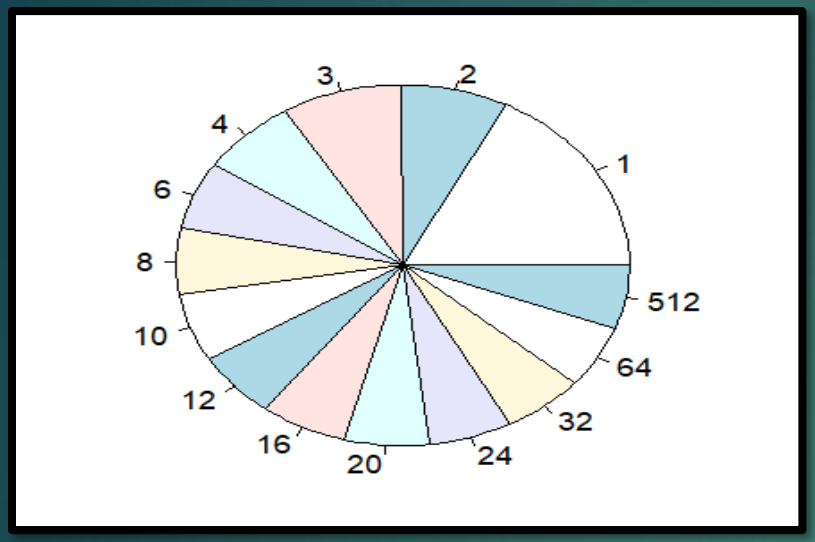


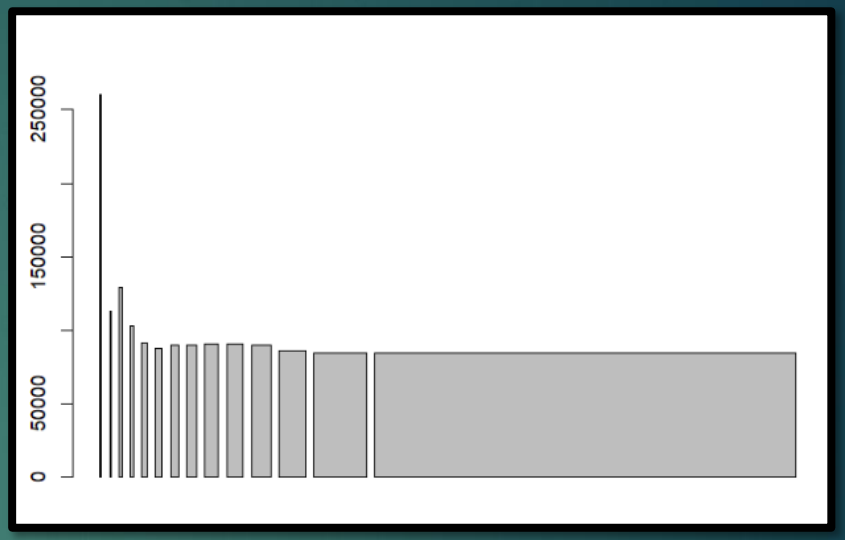


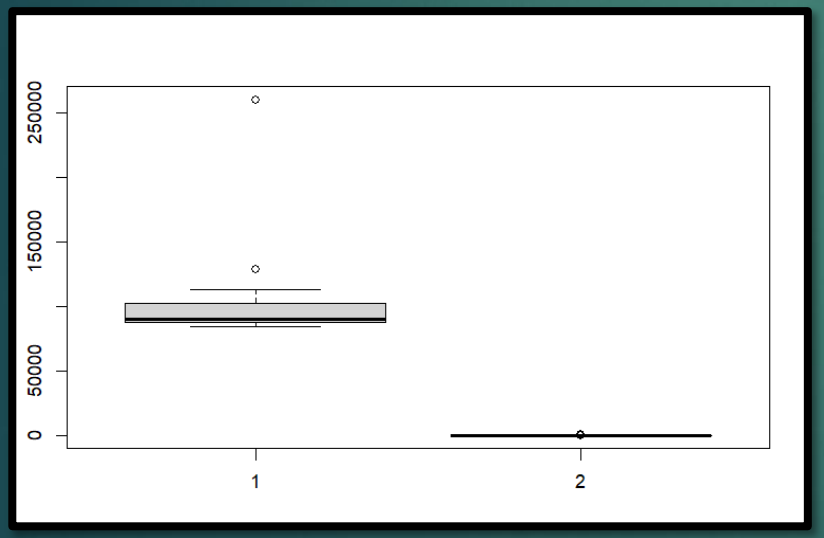


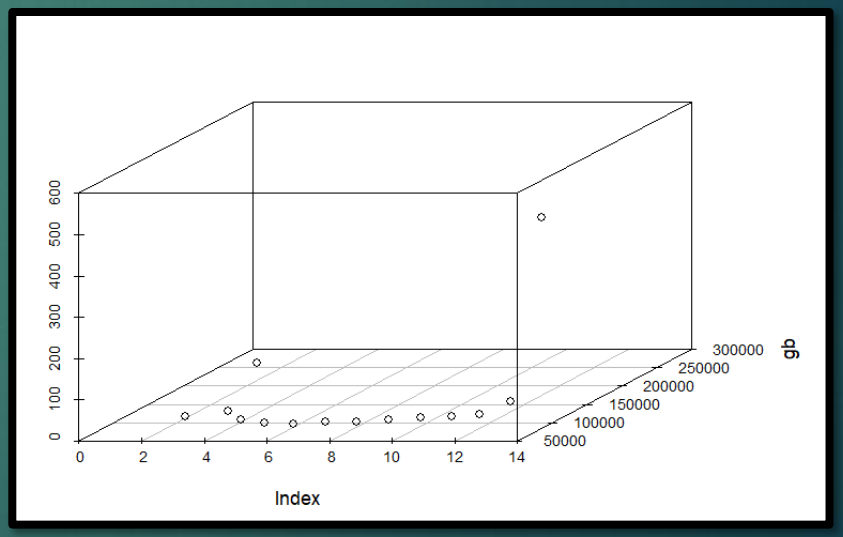


PLOTS OF TYPES OF RAM & SD OF RATINGS:-

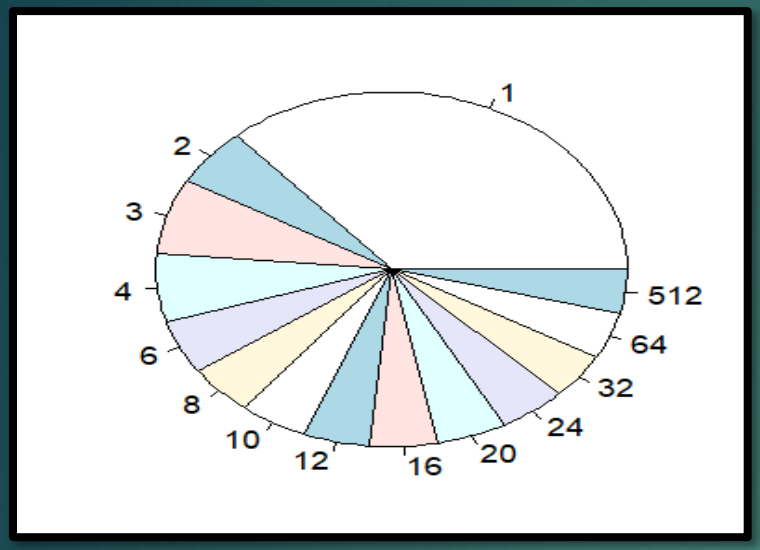


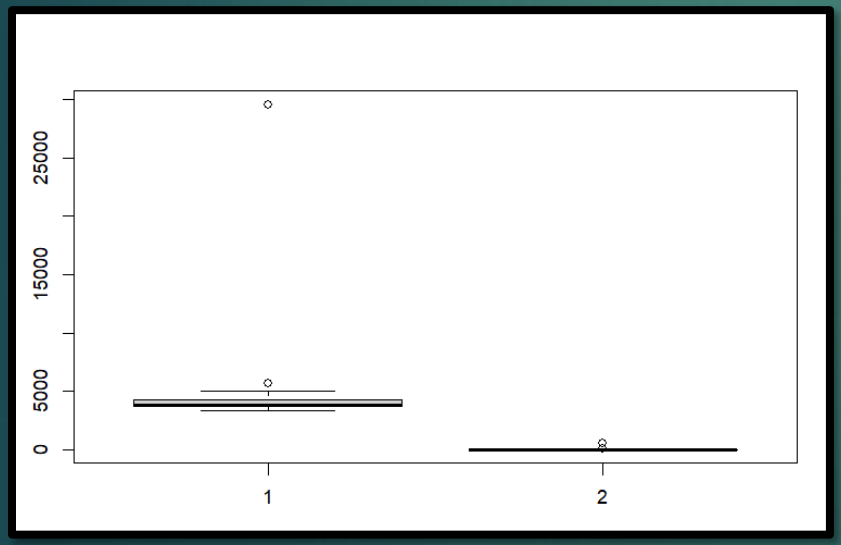


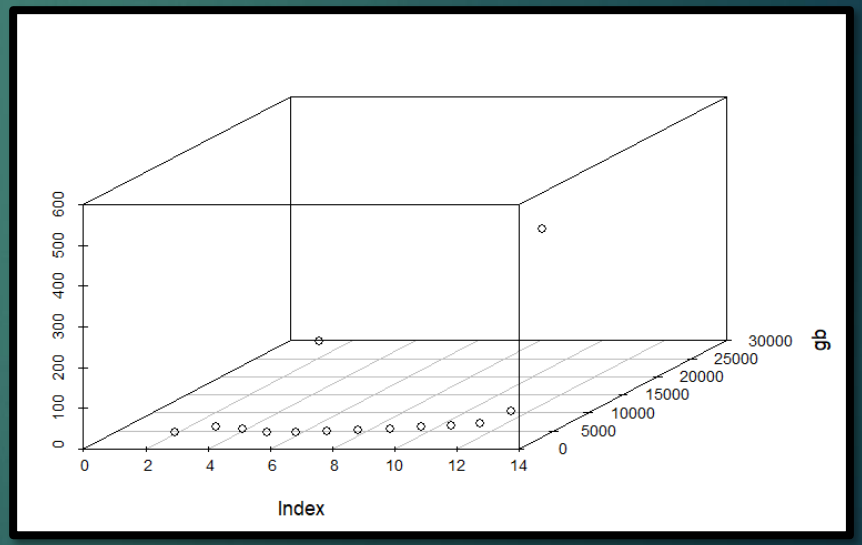




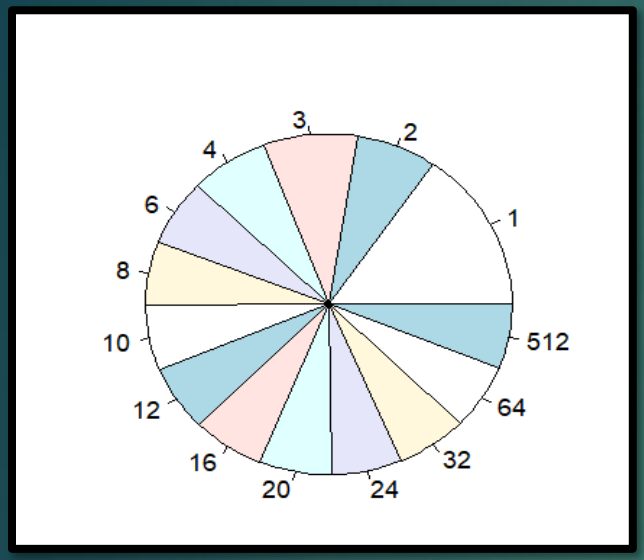
PLOTS OF TYPES OF RAM & MEANS OF REVIEWS:-

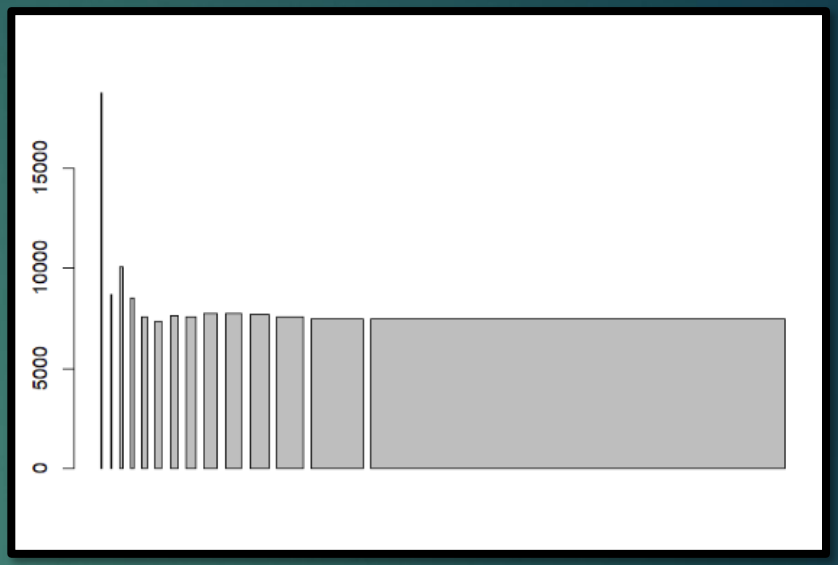


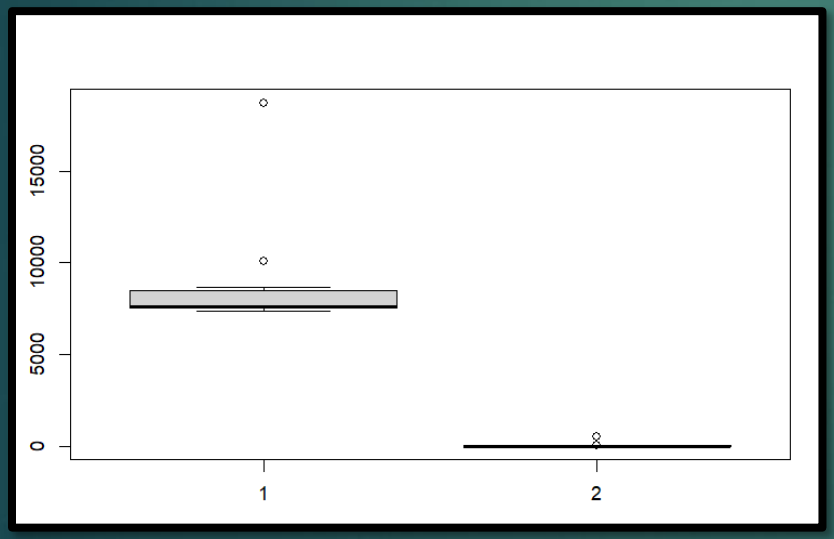


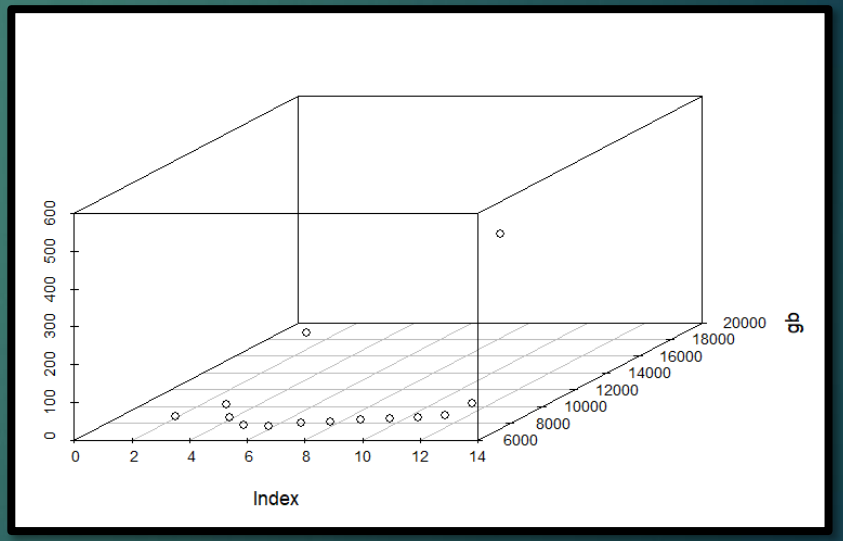


PLOTS OF TYPES OF RAM & SD OF REVIEWS:-

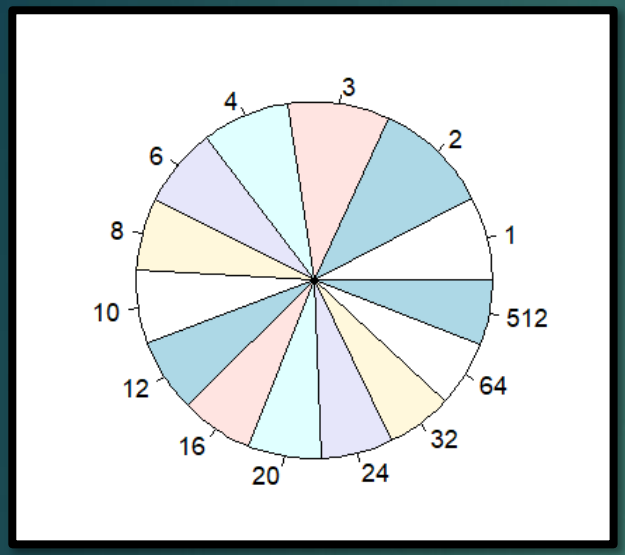


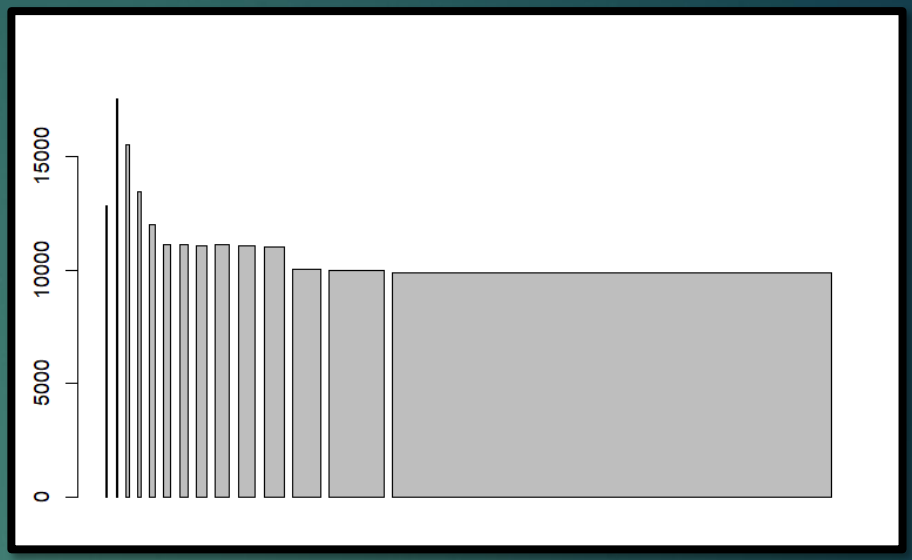


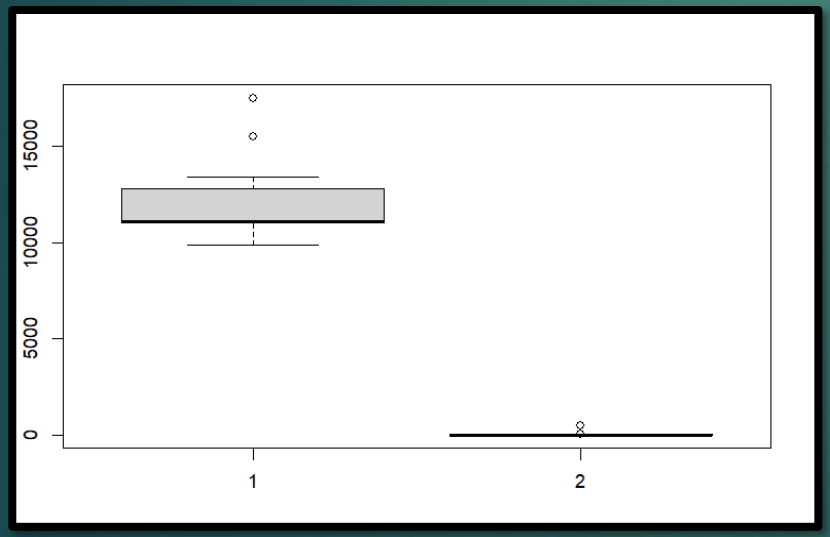


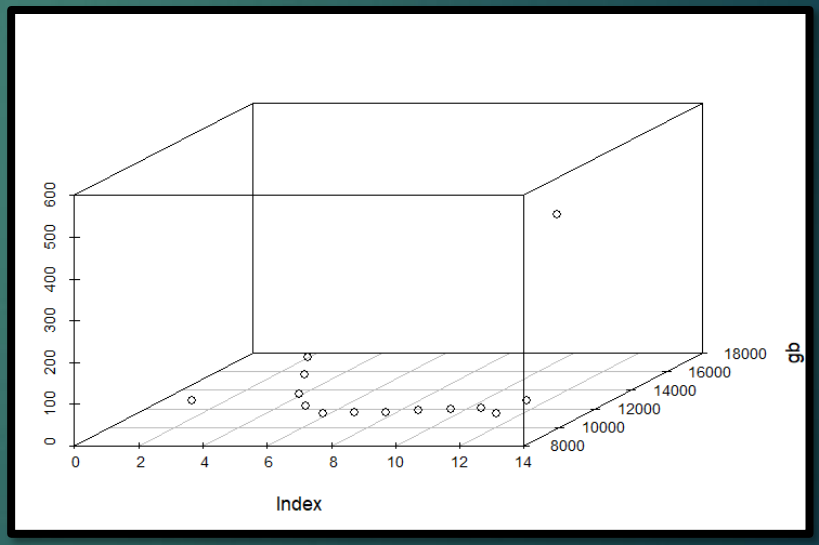


PLOTS OF TYPES OF RAM & MEANS OF LIST PRICE:-

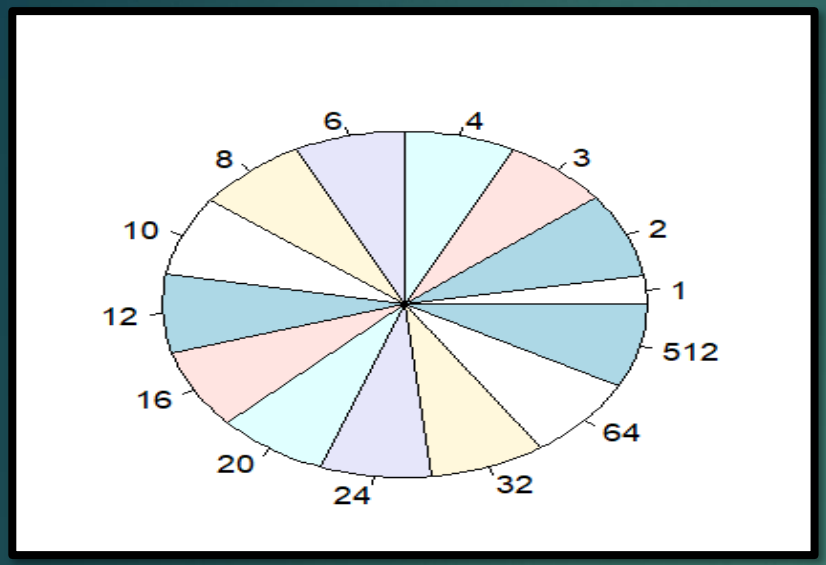


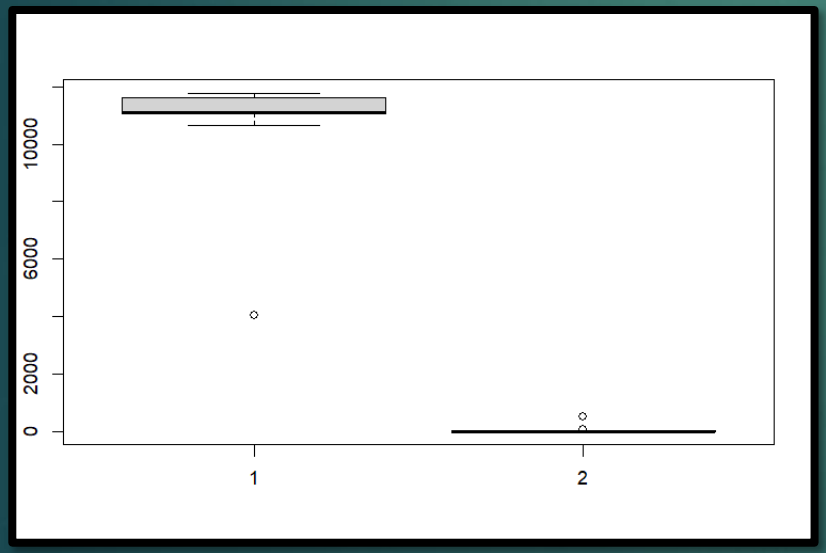


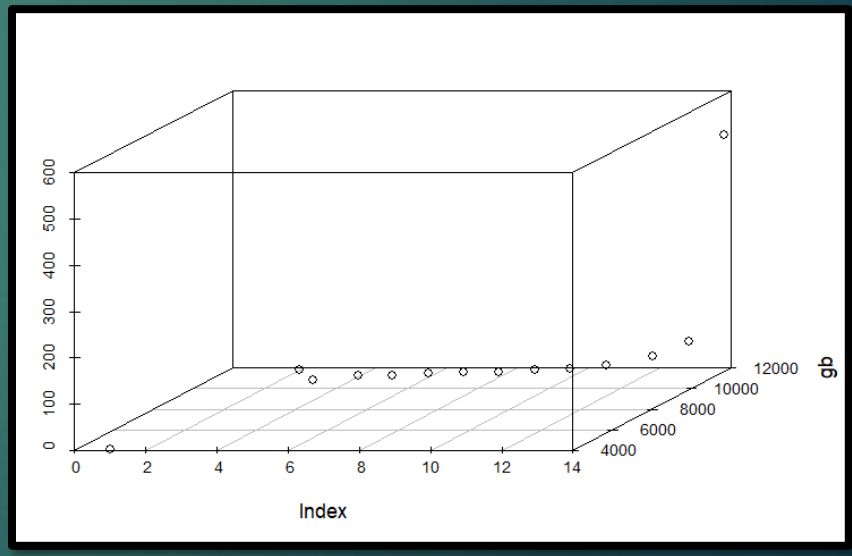




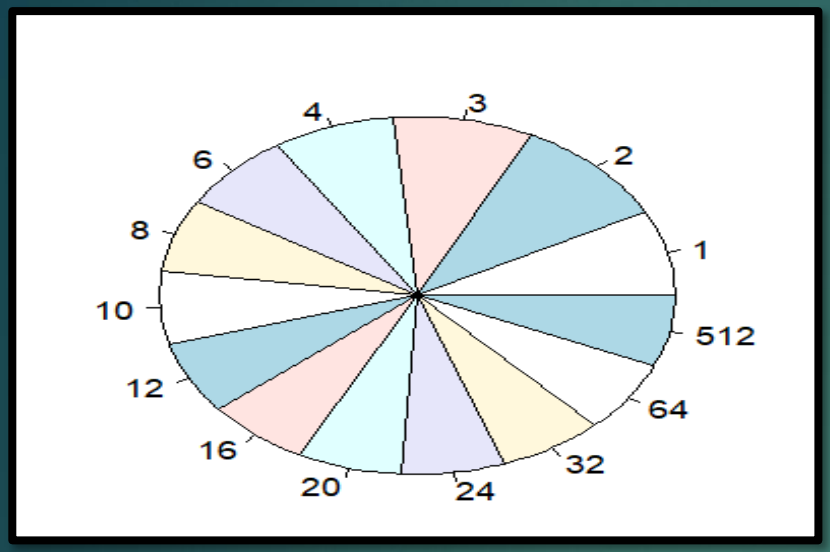
PLOTS OF TYPES OF RAM & SD OF LIST PRICE:-

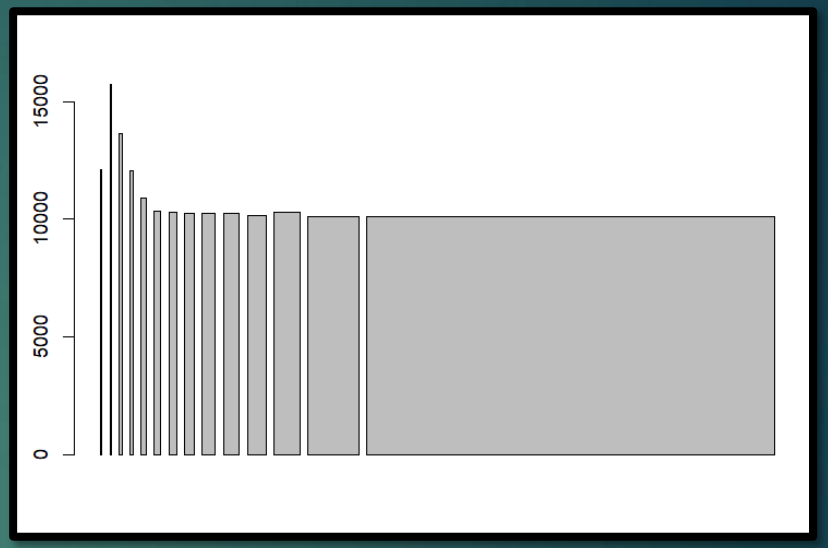


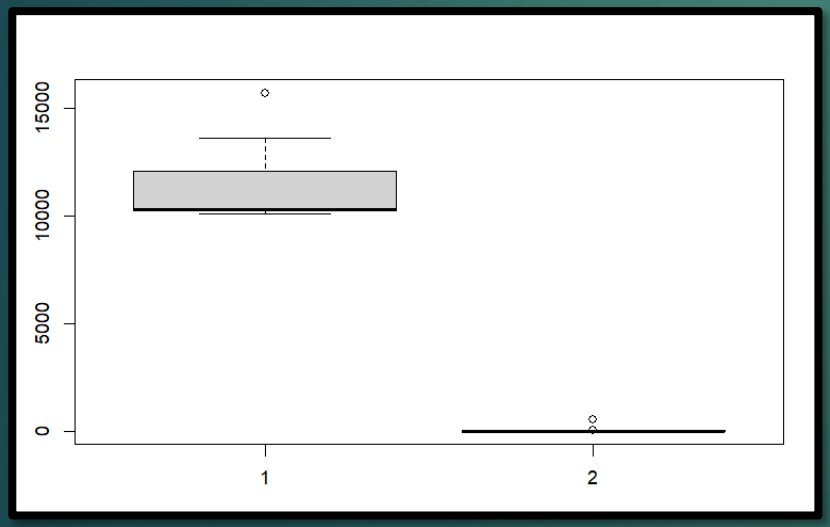


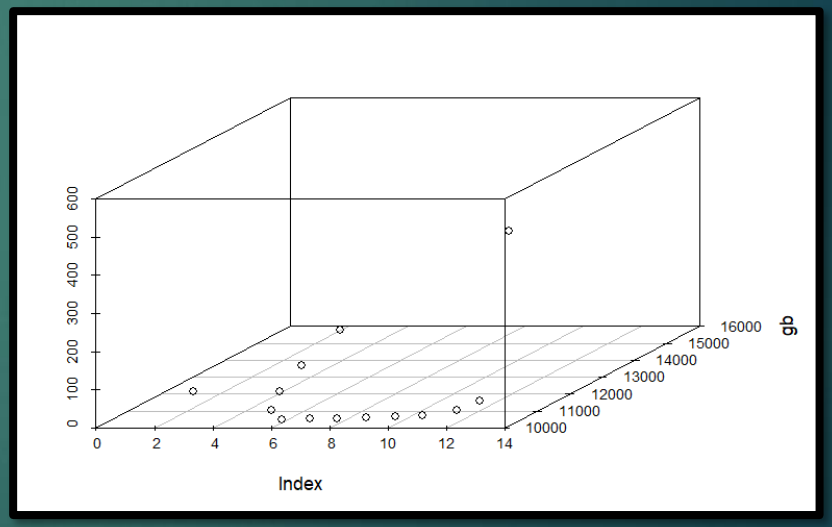


PLOTS OF TYPES OF RAM & MEANS OF SALES PRICE:-

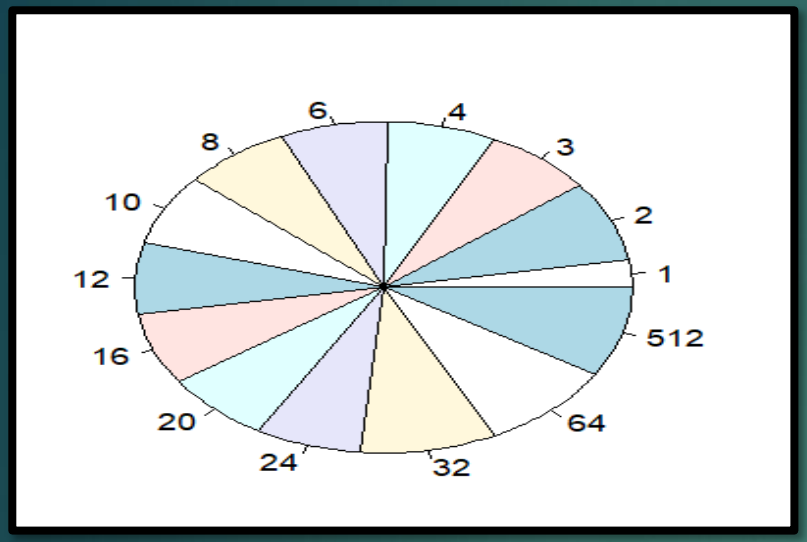


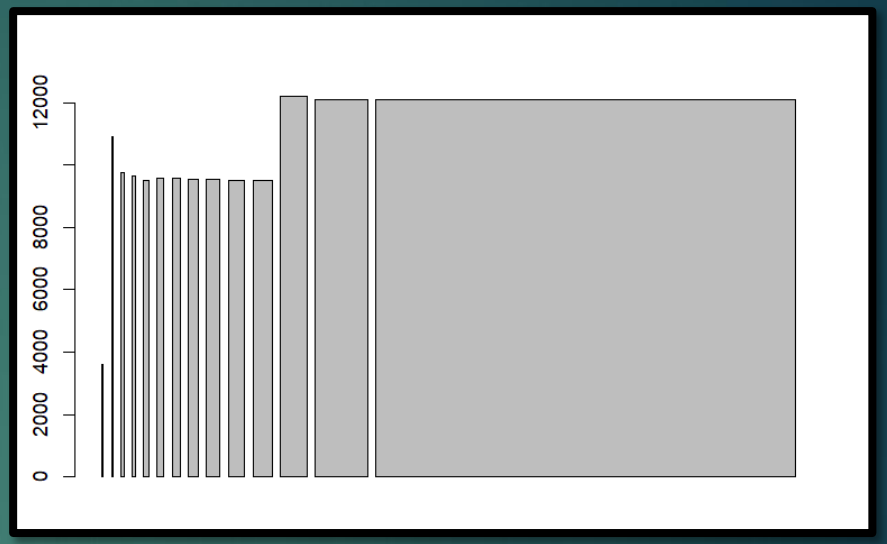


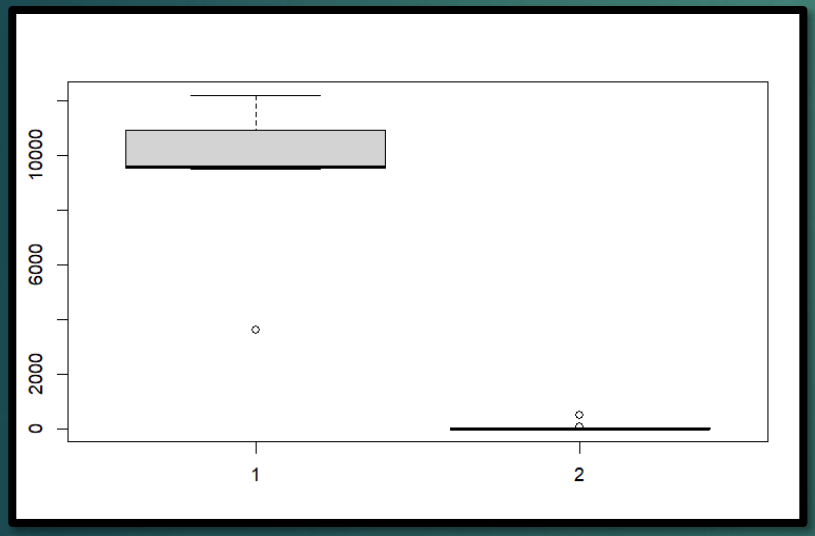


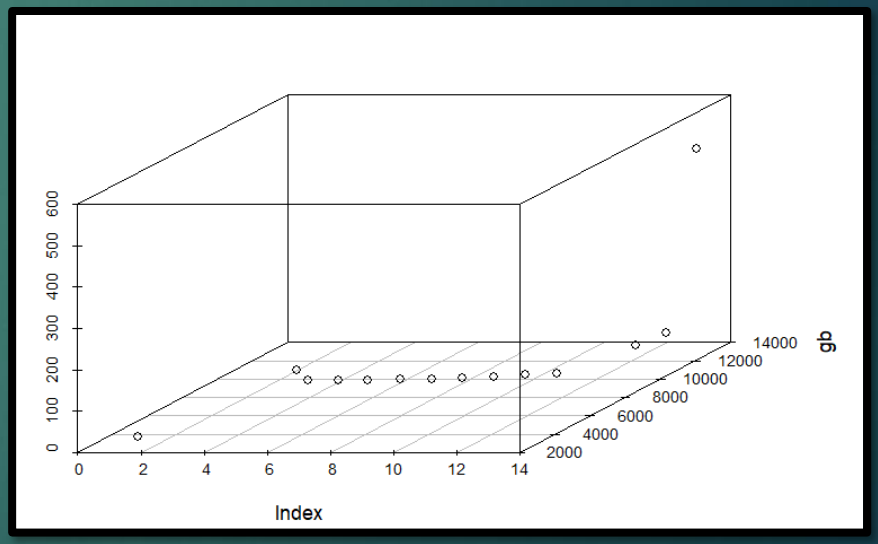


PLOTS OF TYPES OF RAM & SD OF SALES PRICE:-

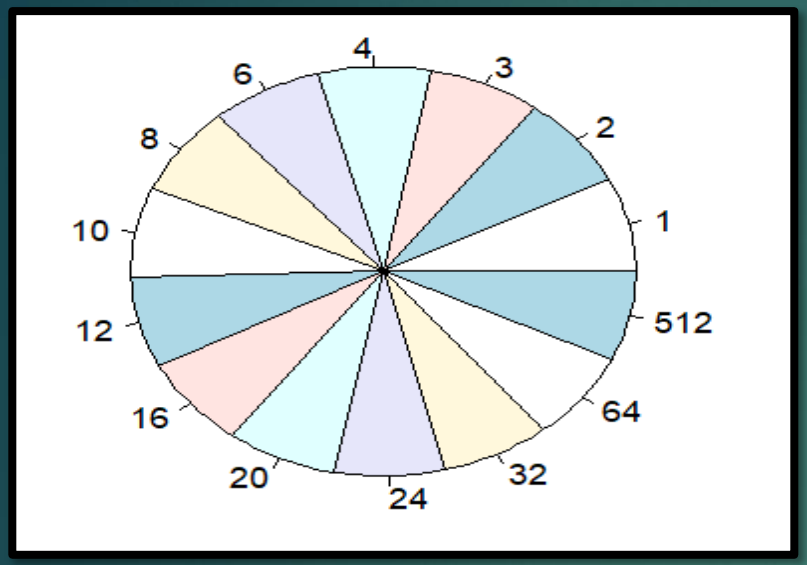


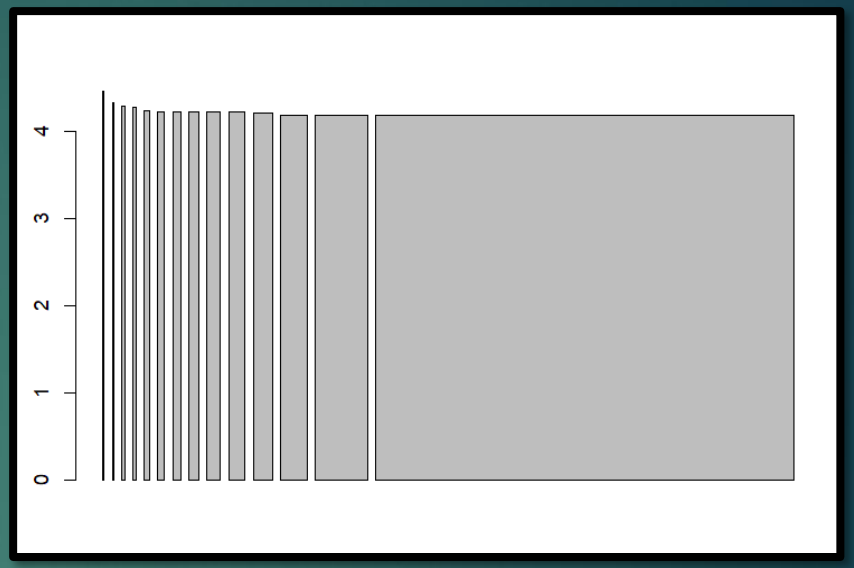


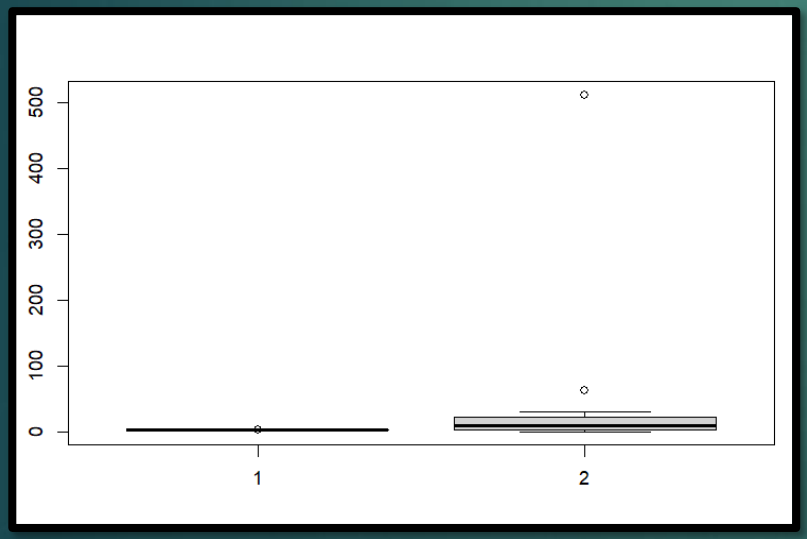


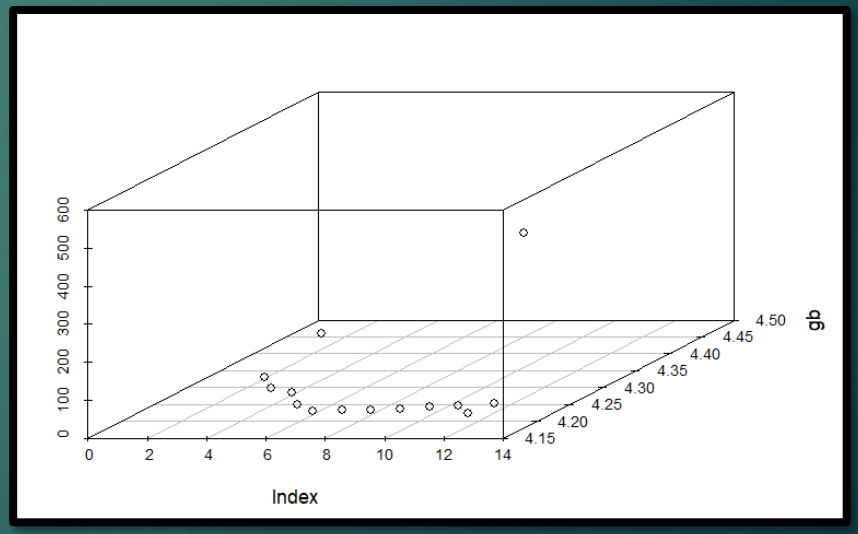


PLOTS OF TYPES OF RAM & MEANS OF STARS:-

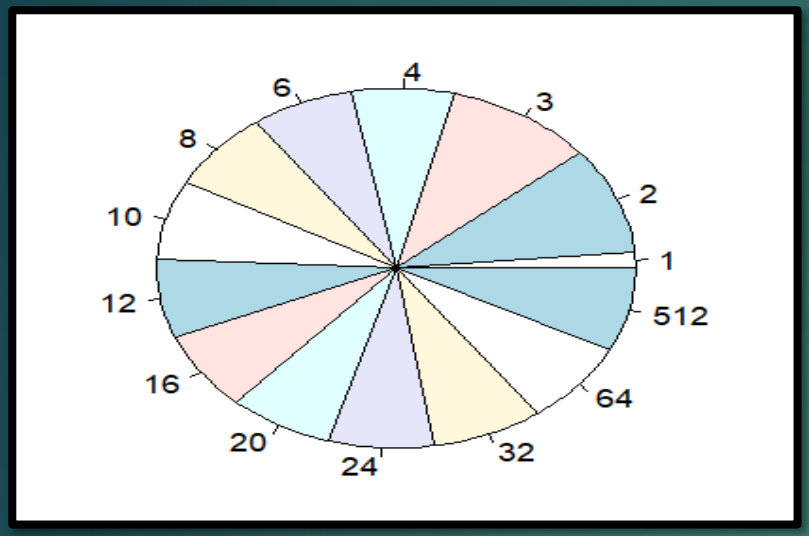


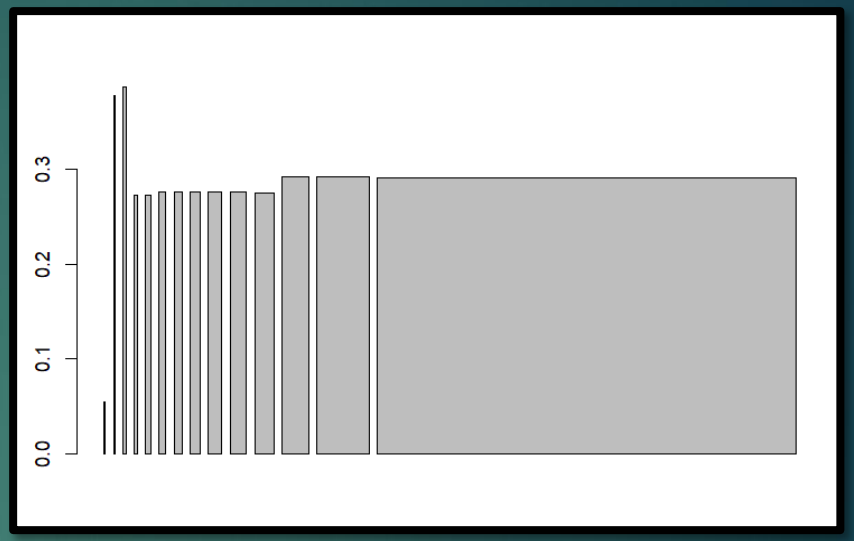


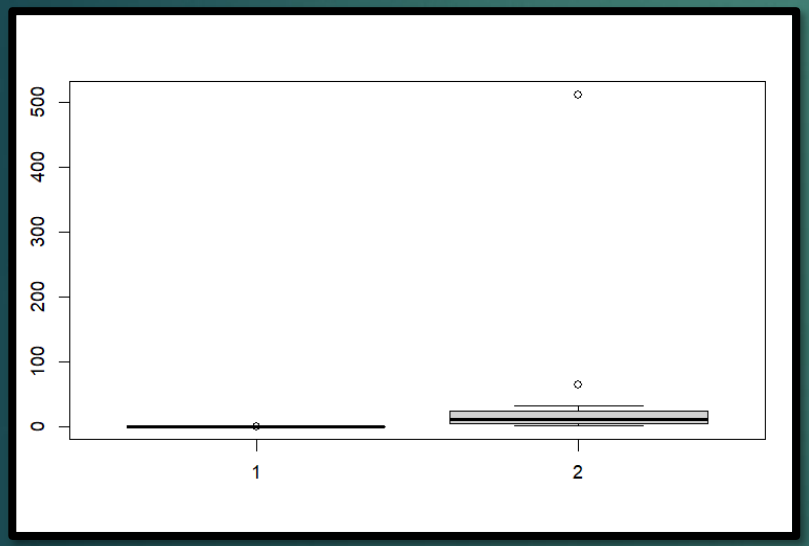


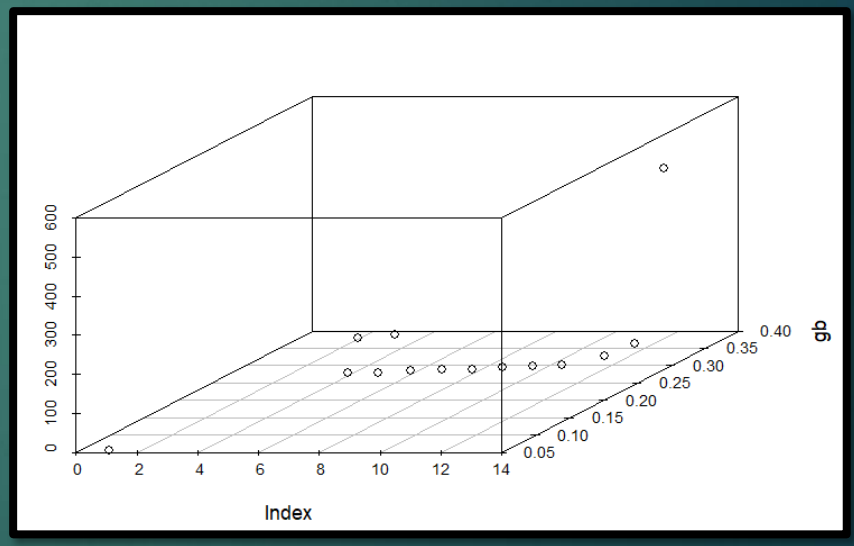


PLOTS OF TYPES OF RAM & SD OF STARS:-









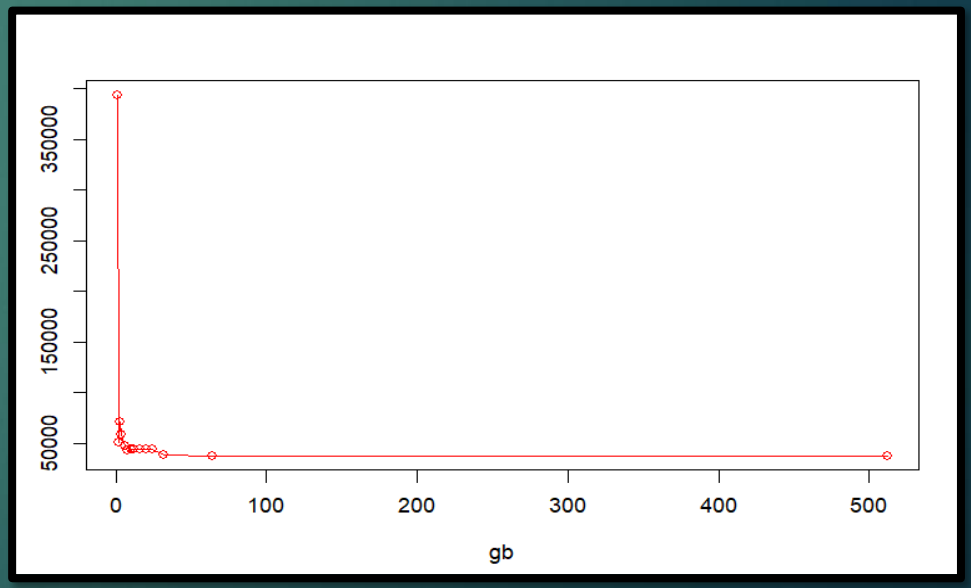
CORRELATION BETWEEN RAM&MEAN RATINGS:-

R-CODE:-

```
z1=data.frame(gb,l1) #l1--mean of ratings
summary(gb,l1)
aa=cov(gb,l1)
aa
bb=var(gb,l1)
bb
corr=cor(gb,l1)
corr1=cor(z1)
corr
corr1
plot(gb,l1,type = "o",col="red")
```

A negative correlation is a relationship between two variables such that as the value of one variable increases, the other decreases.as the gb increase's the review decrease's

```
> z1=data.frame(gb,l1) #l1--mean of ratings
> summary(gb,l1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.5    11.0    51.0    23.0   512.0
> aa=cov(gb,l1)
> aa
[1] -1760518
> bb=var(gb,l1)
> bb
[1] -1760518
> corr=cor(gb,l1)
> corr1=cor(z1)
> corr
[1] -0.1413282
> corr1
           gb          l1
gb  1.0000000 -0.1413282
l1 -0.1413282  1.0000000
> plot(gb,l1,type = "o",col="red")
>
```

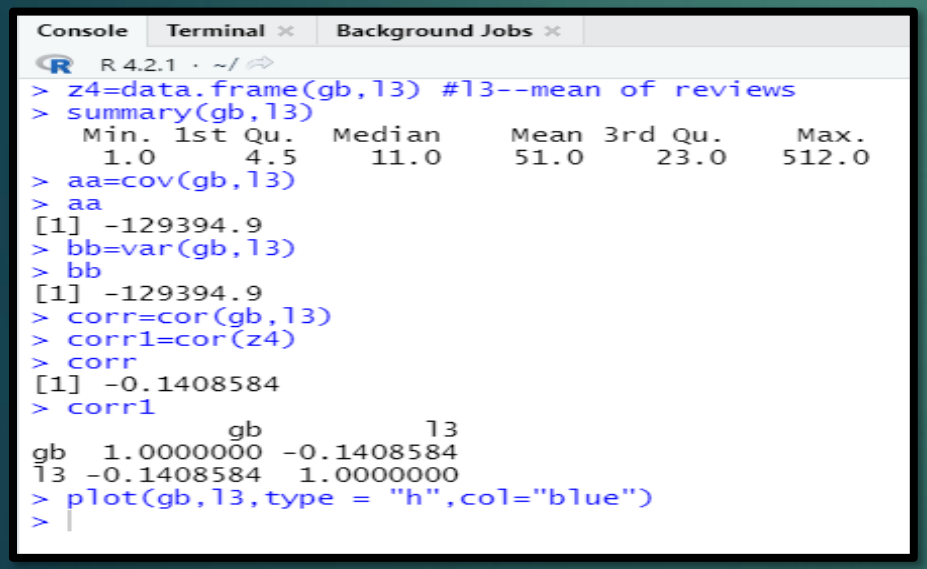


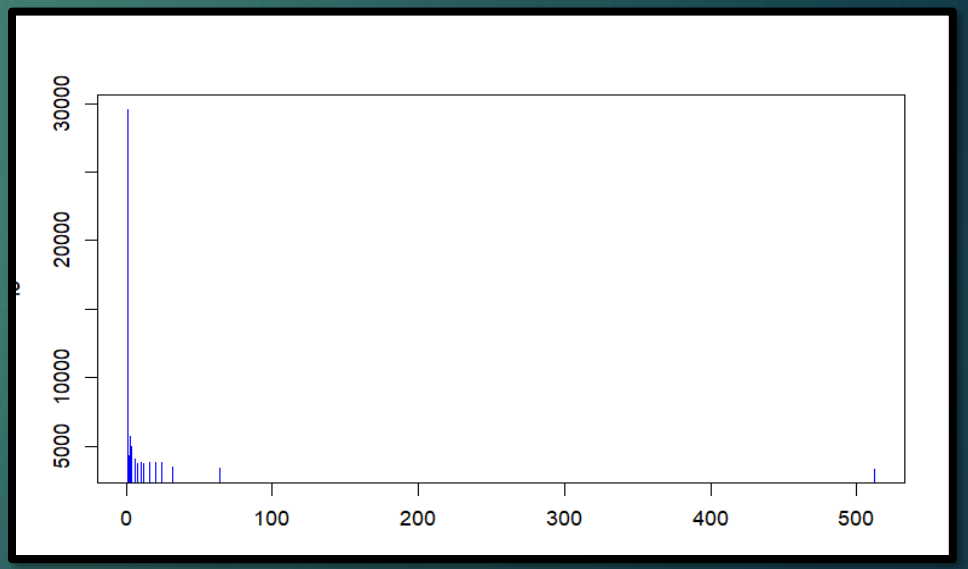
CORRELATION BETWEEN RAM&MEAN RATINGS:-

R-CODE:-

```
z4=data.frame(gb,l3) #l3--mean of reviews
summary(gb,l3)
aa=cov(gb,l3)
aa
bb=var(gb,l3)
bb
corr=cor(gb,l3)
corr1=cor(z4)
corr
corr1
plot(gb,l3,type = "h",col="blue")
```

*A negative correlation is a relationship between two variables such that **as the value of one variable increases, the other decreases**.as the gb increase's the avg.review decrease's*





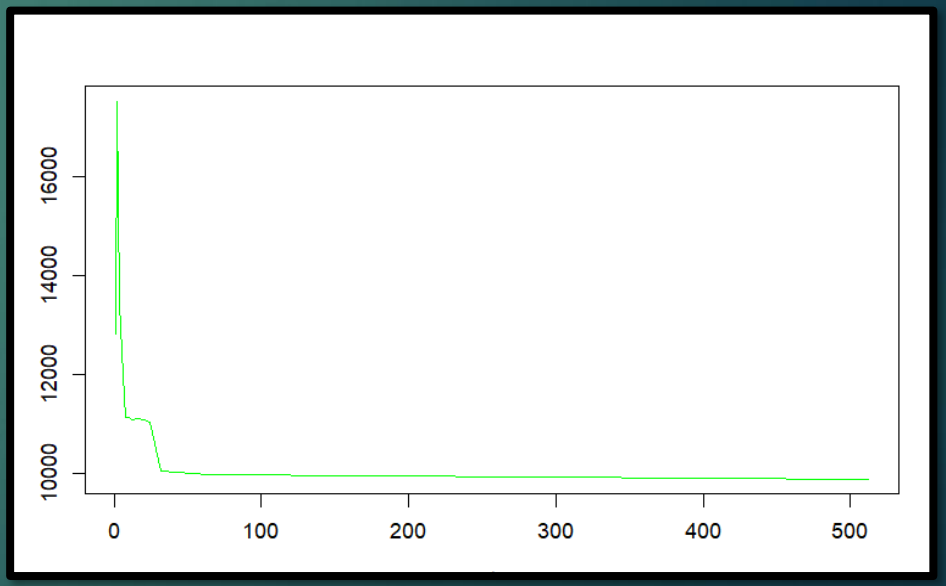
CORRELATION BETWEEN RAM & MEAN OF LIST PRICES :-

R-CODE:-

```
z2=data.frame(gb,l5)   #l5==mean of list
summary(gb,l5)         prices
aa=cov(gb,l5)
aa
bb=var(gb,l5)
bb
corr=cor(gb,l5)
corr1=cor(z2)
corr
corr1
plot(gb,l5,type = "l",col="green")
```

A negative correlation is a relationship between two variables such that as the value of one variable increases, the other decreases.as the gb increase's the avg.list prize decrease's

```
> z2=data.frame(gb,l5)  #l5==mean of list prices
> summary(gb,l5)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.5    11.0    51.0    23.0   512.0
> aa=cov(gb,l5)
> aa
[1] -101385.5
> bb=var(gb,l5)
> bb
[1] -101385.5
> corr=cor(gb,l5)
> corr1=cor(z2)
> corr
[1] -0.3445847
> corr1
           gb          l5
gb  1.0000000 -0.3445847
l5 -0.3445847  1.0000000
> plot(gb,l5,type = "l",col="green")
>
```

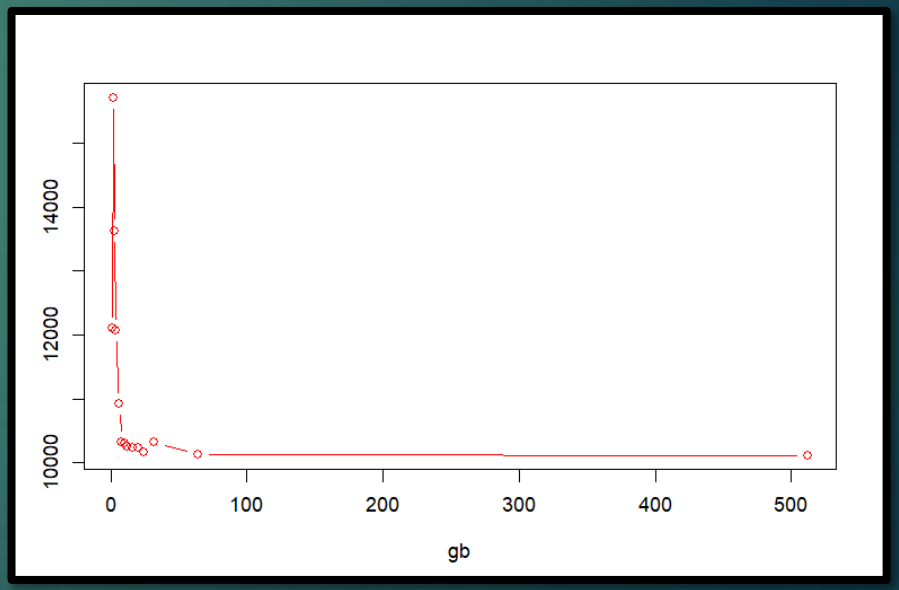


CORRELATION BETWEEN RAM & MEAN OF LIST PRICES :-

R-CODE:-

```
z3=data.frame(gb,l7)  #l7==mean of sales prices
summary(gb,l7)
aa=cov(gb,l7)
aa
bb=var(gb,l7)
bb
corr=cor(gb,l7)
corr1=cor(z3)
corr
corr1
plot(z3,type = "b",col="red")
```

```
> z3=data.frame(gb,l7)  #l7==mean of sales prices
> summary(gb,l7)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.5    11.0    51.0    23.0   512.0
> aa=cov(gb,l7)
> aa
[1] -54276.51
> bb=var(gb,l7)
> bb
[1] -54276.51
> corr=cor(gb,l7)
> corr1=cor(z3)
> corr
[1] -0.2421886
> corr1
           gb          l7
gb  1.0000000 -0.2421886
l7 -0.2421886  1.0000000
> plot(z3,type = "b",col="red")
>
```

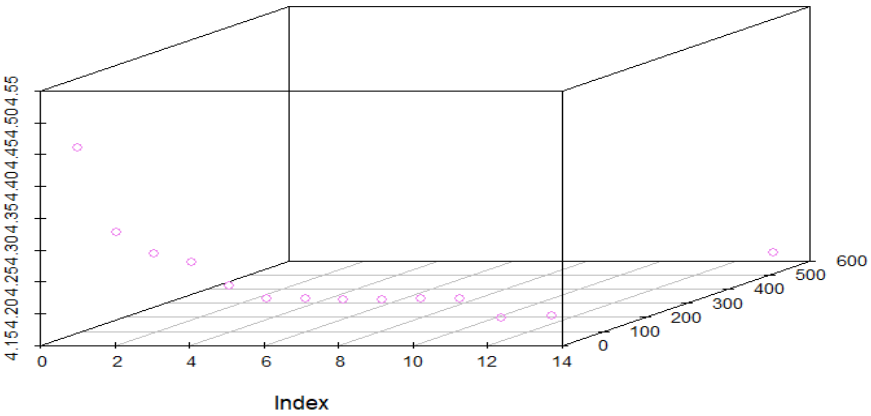


CORRELATION BETWEEN RAM & MEAN STARS:-

R-CODE:-

```
#--------correlation b/w ram&stars
z5=data.frame(gb,l9)  #l9==mean of stars
summary(gb,l9)
aa=cov(gb,l9)
aa
bb=var(gb,l9)
bb
corr=cor(gb,l9)
corr1=cor(z5)
corr
corr1
scatterplot3d(z5,color = "violet")
```

```
R 4.2.1 · ~/
> #--------correlation b/w ram&stars
> z5=data.frame(gb,l9)  #l9==mean of stars
> summary(gb,l9)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.5    11.0    51.0    23.0   512.0
> aa=cov(gb,l9)
> aa
[1] -3.251296
> bb=var(gb,l9)
> bb
[1] -3.251296
> corr=cor(gb,l9)
> corr1=cor(z5)
> corr
[1] -0.3271694
> corr1
           gb         l9
gb  1.0000000 -0.3271694
l9 -0.3271694  1.0000000
> scatterplot3d(z5,color = "violet")
>
```



SUMMARY:-

```
summary(gb,l1,l3,l5,l7)
```

```
> summary(gb,l1,l3,l5,l7)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.5    11.0    51.0    23.0   512.0
> |
```

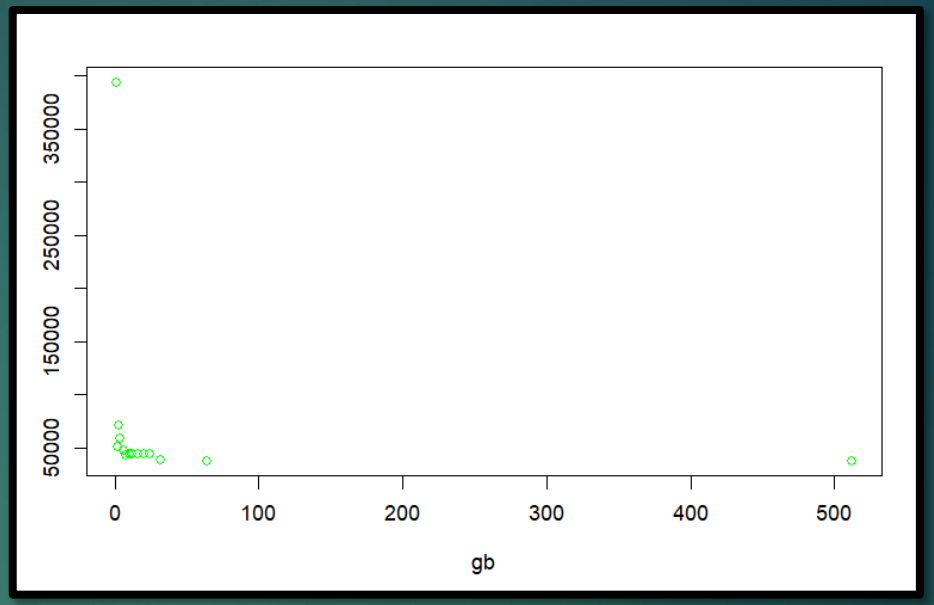
LINEAR REGRESSION:-

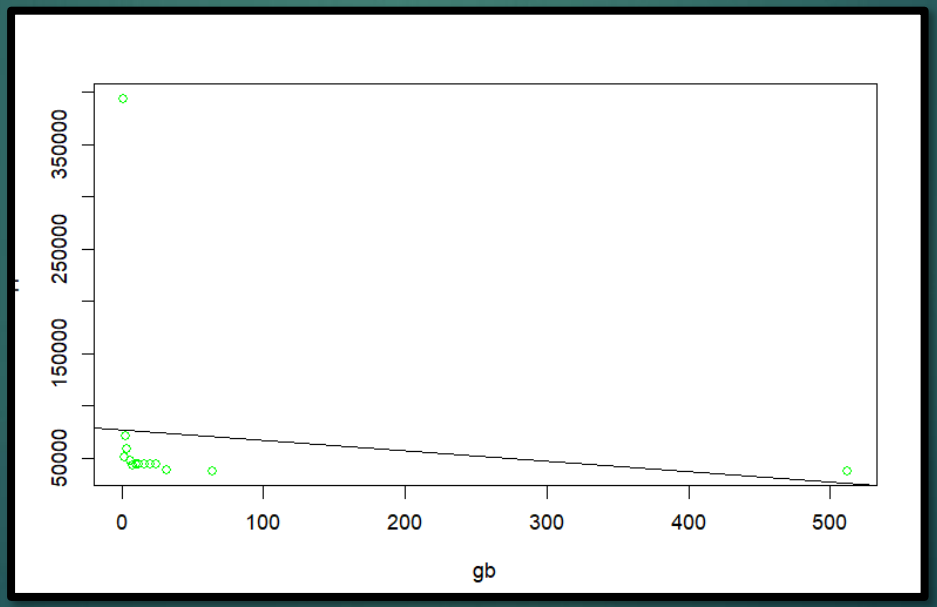
TO FIND LINEAR REGRESSION COEFFICIENT l1 ON gb:-

R-CODE:-

```
#LINEAR REGRESSION MODEL
#1
#TO FIND LINEAR REGRESSION COEFFICIENT l1 ON gb
mean(gb)
mean(l1)
REGRESSION = lm(l1~gb)
REGRESSION
plot(gb,l1,col="green")
abline(REGRESSION)
```

```
R 4.2.1 · ~/
> mean(gb)
[1] 51
> mean(l1)
[1] 71636.55
> REGRESSION = lm(l1~gb)
> REGRESSION

Call:
lm(formula = l1 ~ gb)

Coefficients:
(Intercept)           gb
   76657.82       -98.46

> plot(gb,l1,col="green")
> abline(REGRESSION)
>
```





TO FIND LINEAR REGRESSION COEFFICIENT GB ON L3:-

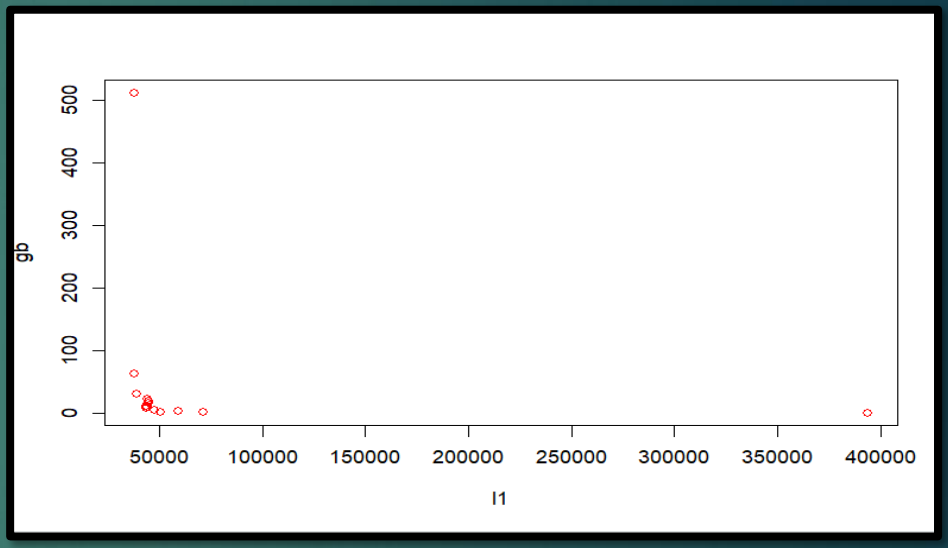
```
#TO FIND LINEAR REGRESSION COEFFICIENT GB ON L3
REGRESSION = lm(gb~l1)
REGRESSION
plot(l1,gb,col="red")
abline(REGRESSION)
```

By giving the particular review we can find most of the people preferred "ram(gb)" mobile.

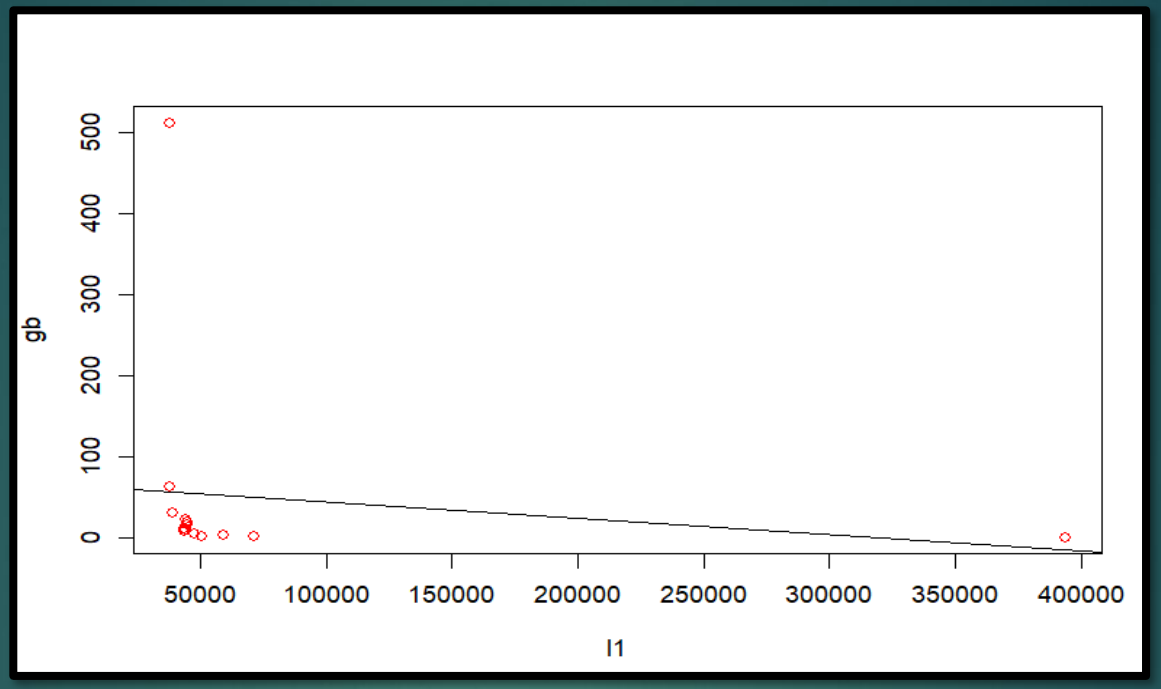
Regression relation

gb=65.5327976+l1(-0.0002029)

```
> #TO FIND LINEAR REGRESSION COEFFICIENT GB ON L3
> REGRESSION = lm(gb~l1)
> REGRESSION

Call:
lm(formula = gb ~ l1)

Coefficients:
(Intercept)           l1
 65.5327976   -0.0002029

> plot(l1,gb,col="red")
> abline(REGRESSION)
>
```



In statistical analysis, regression is used to **identify the associations between variables occurring in some data**. It can show both the magnitude of such an association and also determine its statistical significance (i.e., whether or not the association is likely due to chance).



TO FIND LINEAR REGRESSION COEFFICIENT l3 ON gb:-

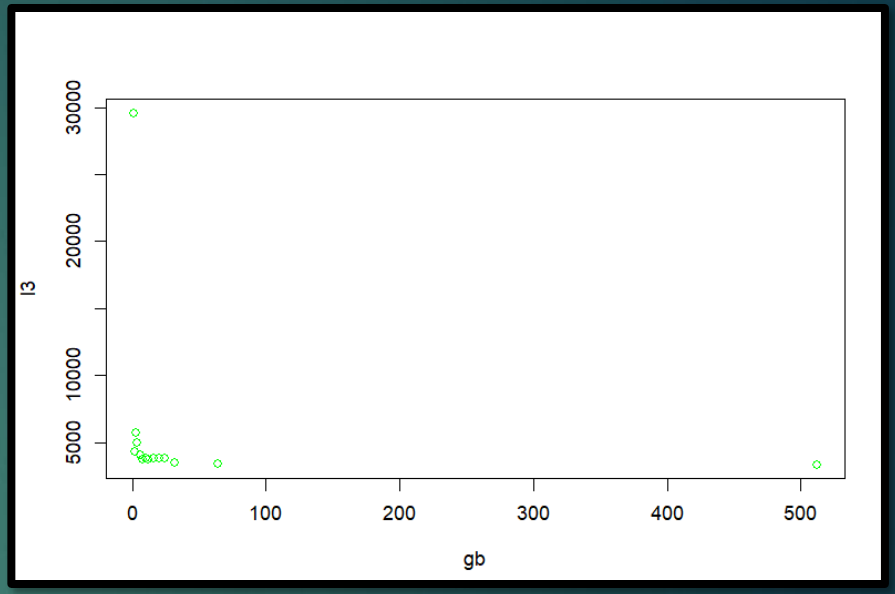
```
TO FIND LINEAR REGRESSION COEFFICIENT l3 ON gb
mean(gb)
mean(l3)
REGRESSION = lm(l3~gb)
REGRESSION
plot(gb,l3,col="green")
abline(REGRESSION)
```

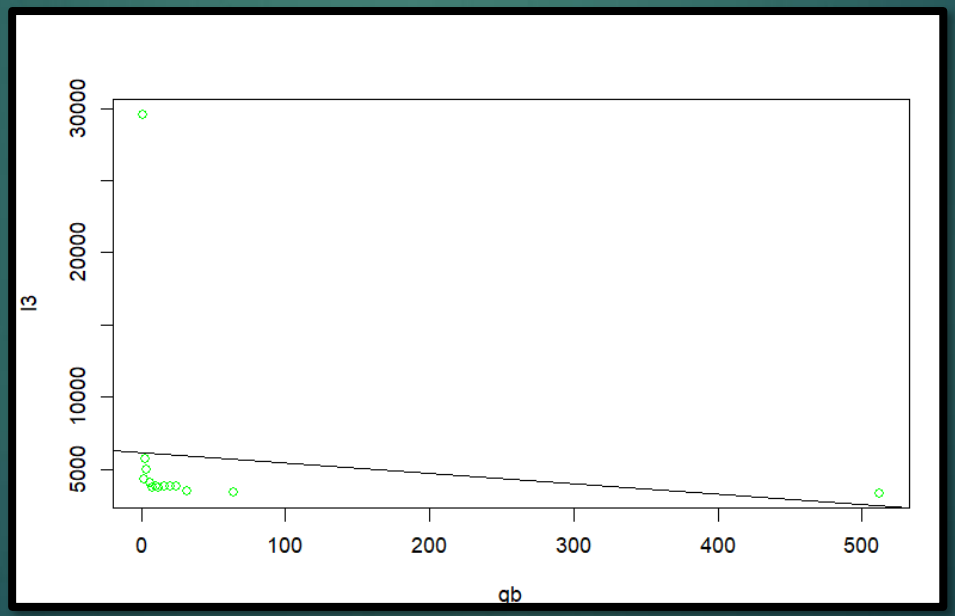
By giving the particular rating we can find most of the people preferred “ram(gb)” mobile.

Regression relation

gb=65.5327976+l1(-0.0002029)

```
> #TO FIND LINEAR REGRESSION COEFFICIENT GB ON L3
> REGRESSION = lm(gb~l1)
> REGRESSION

Call:
lm(formula = gb ~ l1)

Coefficients:
(Intercept)           l1
 65.5327976   -0.0002029

> plot(l1,gb,col="red")
> abline(REGRESSION)
> plot(l1,gb,col="red")
> abline(REGRESSION)
>
```

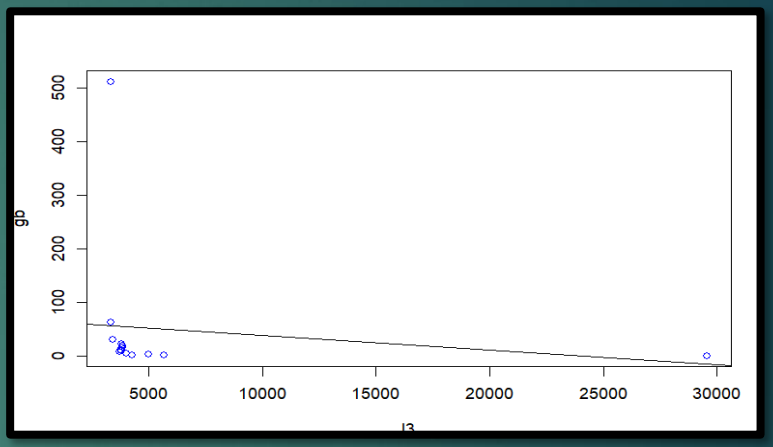


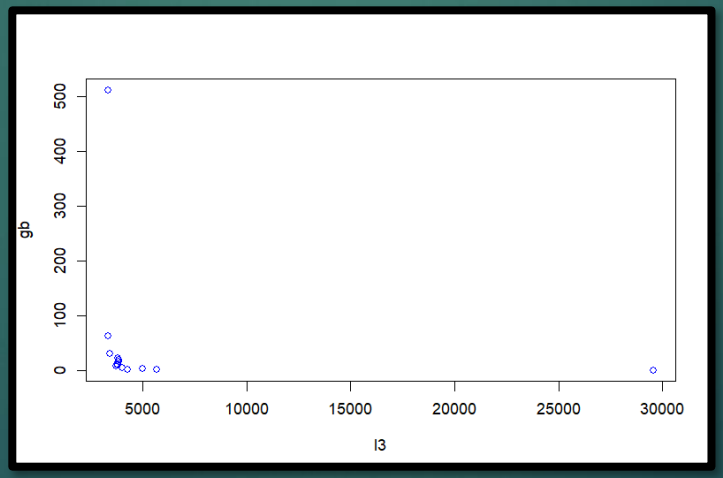


TO FIND LINEAR REGRESSION COEFFICIENT gb ON l3:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT gb ON l3
REGRESSION = lm(gb~l3)
REGRESSION
plot(l3,gb,,col="blue")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT gb ON l3
> REGRESSION = lm(gb~l3)
> REGRESSION

Call:
lm(formula = gb ~ l3)

Coefficients:
(Intercept)           l3
  66.922476    -0.002742

> plot(l3,gb,,col="blue")
> abline(REGRESSION)
>
```





By giving the particular review we can find most of the people preferred "ram(gb)" mobile.

Regression relation

gb=66.922476+l1(-0.00027)

TO FIND LINEAR REGRESSION COEFFICIENT l5 ON gb:-

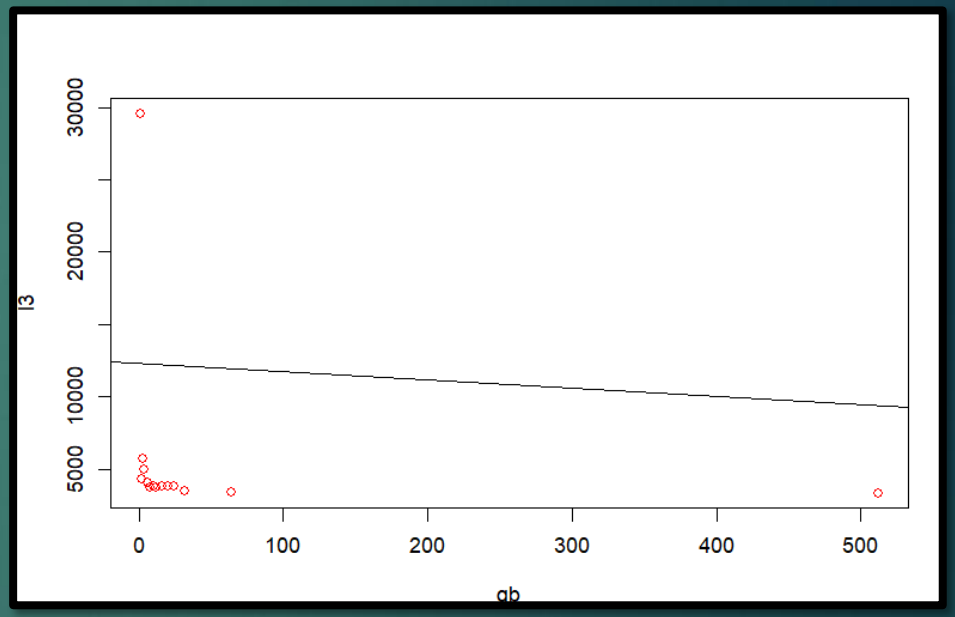
```
#TO FIND LINEAR REGRESSION COEFFICIENT l5 ON gb
mean(gb)
mean(l5)
REGRESSION = lm(l5~gb)
REGRESSION
plot(gb,l3,col="red")
abline(REGRESSION)
```

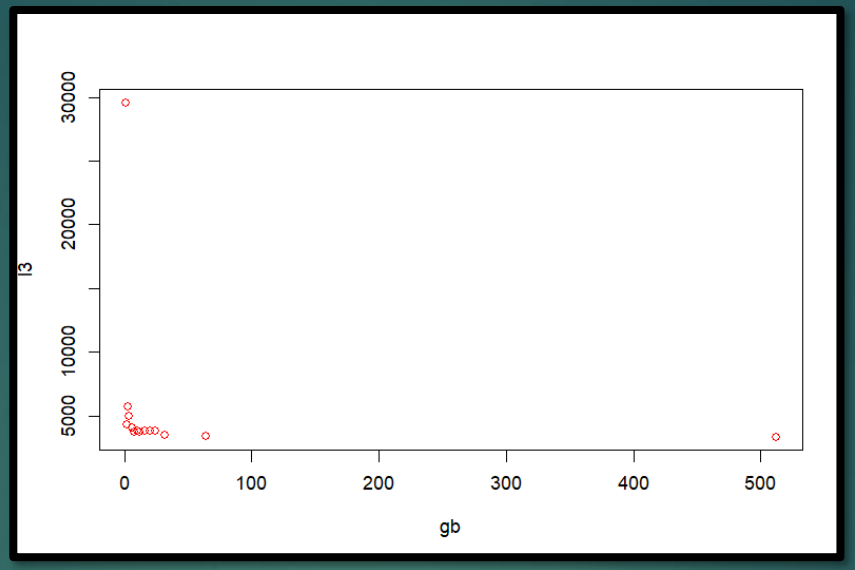
By giving the particular ram(gb) we can find most of the people preferred "list prize" mobile.

Regression relation

gb=12262.49+l1(-5.67)

```
> #TO FIND LINEAR REGRESSION COEFFICIENT l5 ON gb
> mean(gb)
[1] 51
> mean(l5)
[1] 11973.32
> REGRESSION = lm(l5~gb)
> REGRESSION

Call:
lm(formula = l5 ~ gb)

Coefficients:
(Intercept)           gb
   12262.49        -5.67

> plot(gb,l3,col="red")
> abline(REGRESSION)
>
```

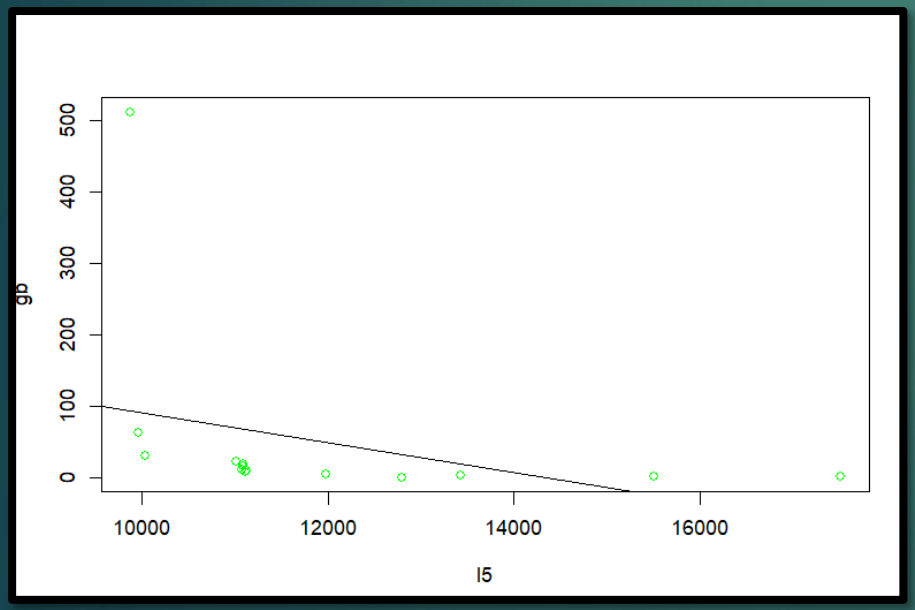


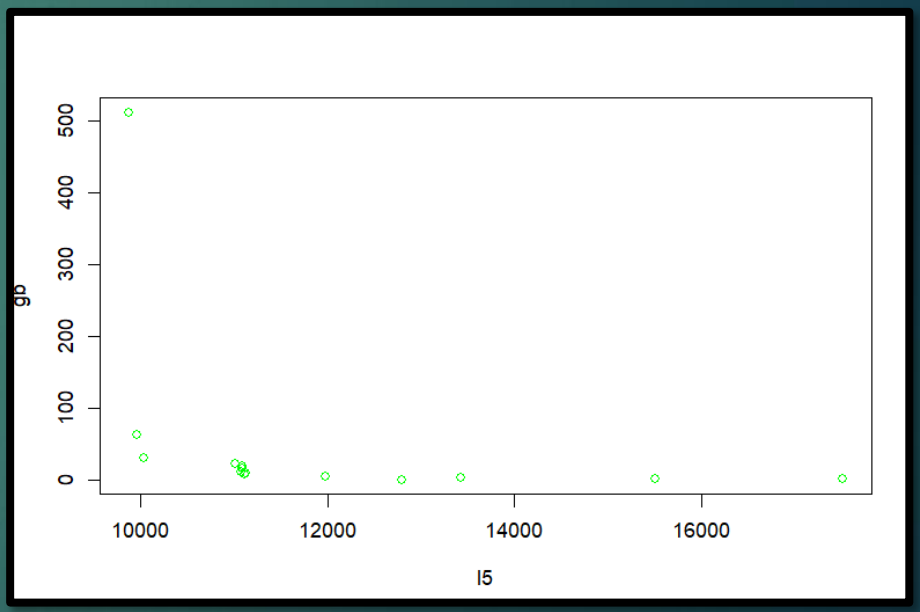


TO FIND LINEAR REGRESSION COEFFICIENT gb ON l5:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT gb ON l5
REGRESSION = lm(gb~l5)
REGRESSION
plot(l5,gb,col="green")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT gb ON l5
> REGRESSION = lm(gb~l5)
> REGRESSION

Call:
lm(formula = gb ~ l5)

Coefficients:
(Intercept)           l5
  301.74280     -0.02094

> plot(l5,gb,col="green")
> abline(REGRESSION)
>
```

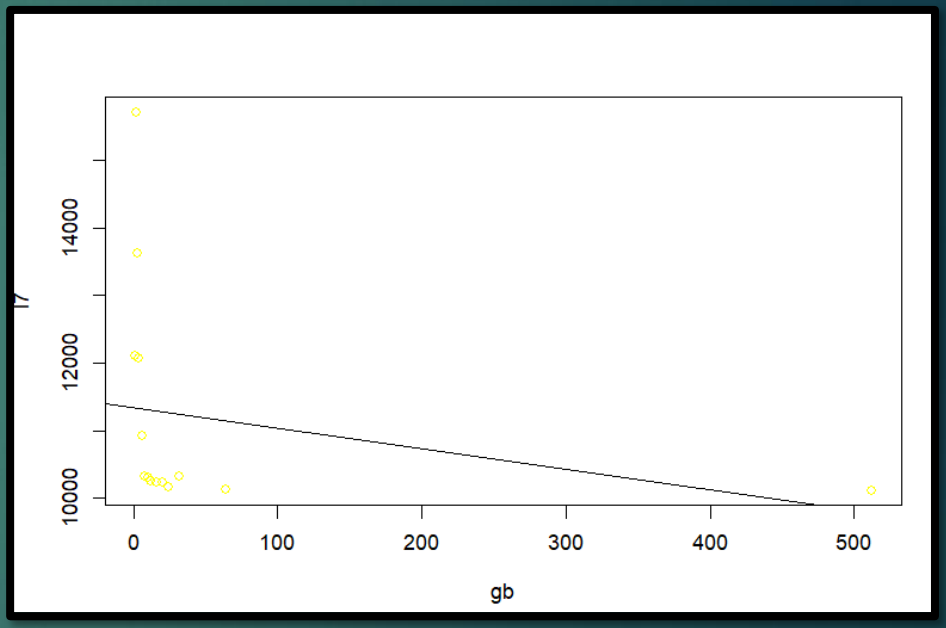


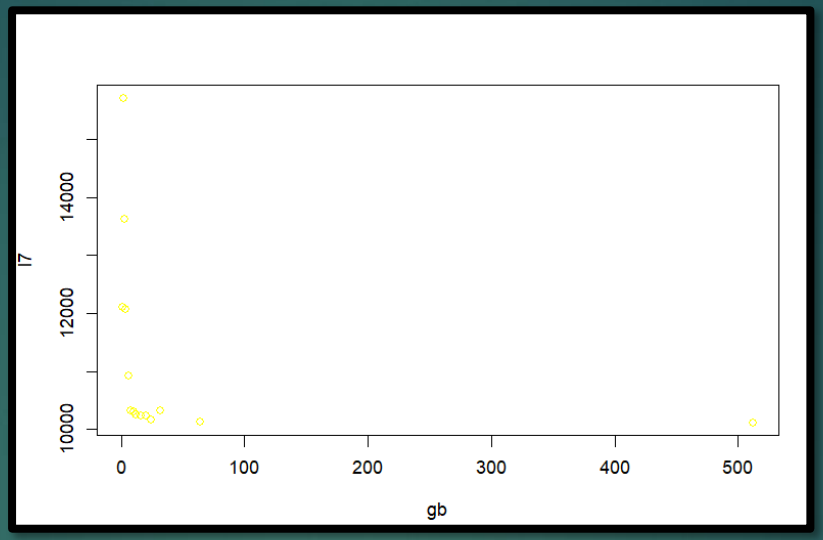


TO FIND LINEAR REGRESSION COEFFICIENT l7 ON gb:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT l7 ON gb
mean(gb)
mean(l7)
REGRESSION = lm(l7~gb)
REGRESSION
plot(gb,l7,col="yellow")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT l7 ON gb
> mean(gb)
[1] 51
> mean(l7)
[1] 11177.4
> REGRESSION = lm(l7~gb)
> REGRESSION

Call:
lm(formula = l7 ~ gb)

Coefficients:
(Intercept)           gb
  11332.201       -3.035

> plot(gb,l7,col="yellow")
> abline(REGRESSION)
>
```

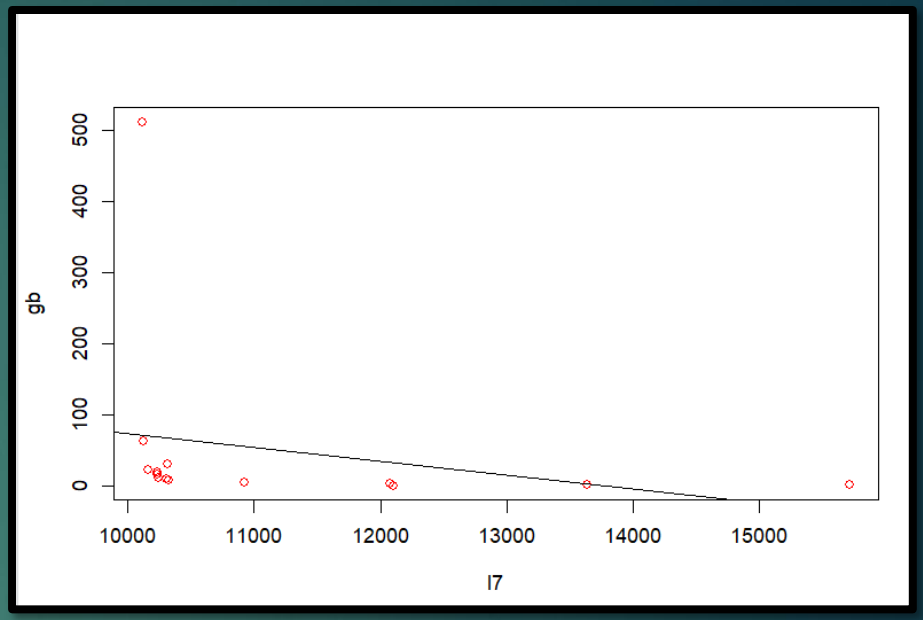


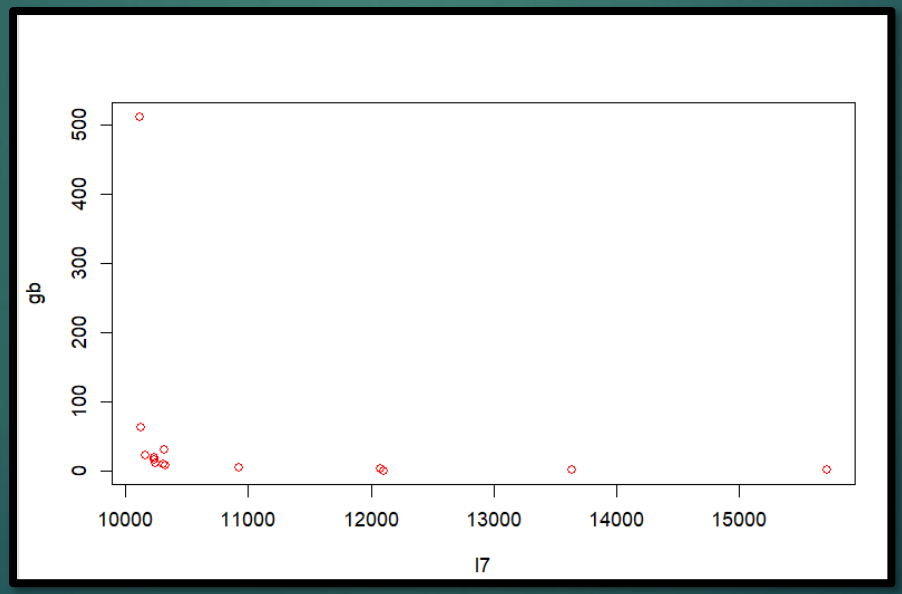


TO FIND LINEAR REGRESSION COEFFICIENT gb ON l7:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT gb ON l7
REGRESSION = lm(gb~l7)
REGRESSION
plot(l7,gb,col="red")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT gb ON l7
> REGRESSION = lm(gb~l7)
> REGRESSION

Call:
lm(formula = gb ~ l7)

Coefficients:
(Intercept)           l7
  266.98993     -0.01932

> plot(l7,gb,col="red")
> abline(REGRESSION)
>
```

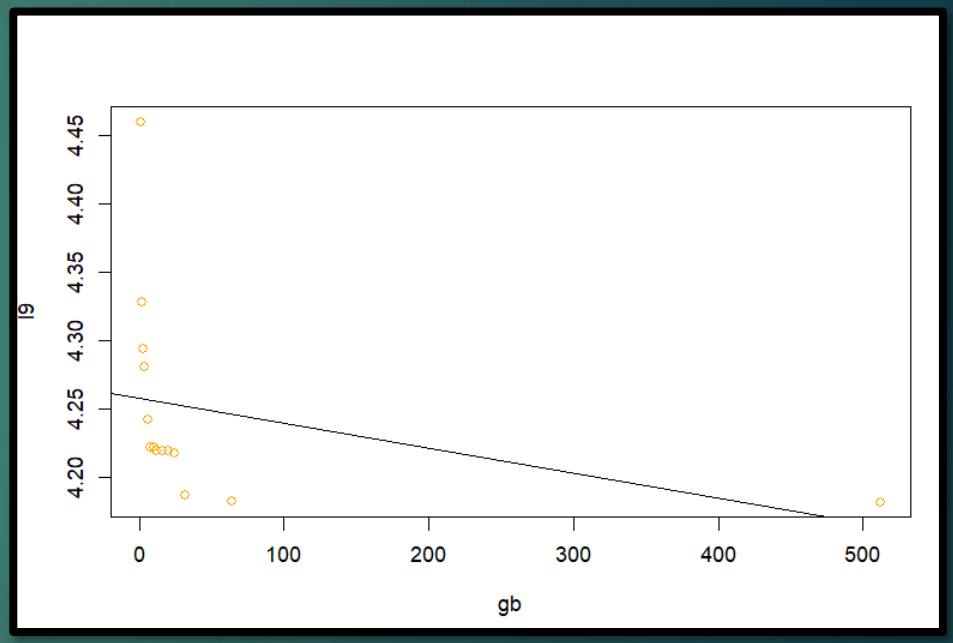


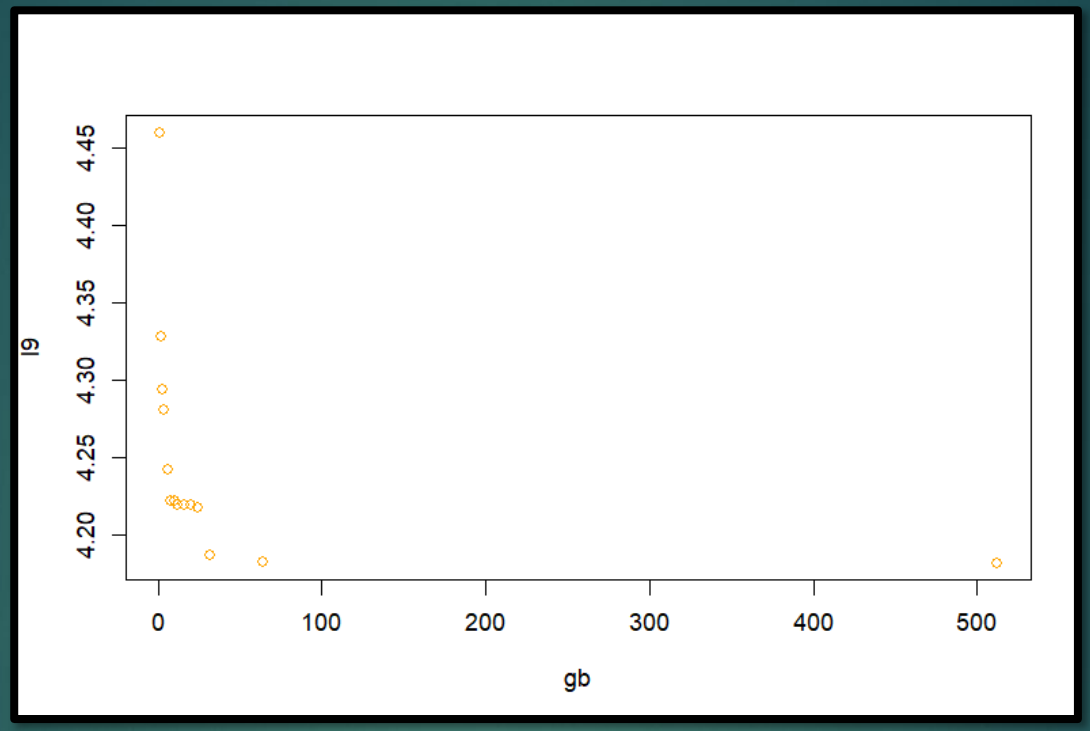


TO FIND LINEAR REGRESSION COEFFICIENT l9 ON gb:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT l9 ON gb
mean(gb)
mean(l9)
REGRESSION = lm(l9~gb)
REGRESSION
plot(gb,l9,col="orange")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT l9 ON gb
> mean(gb)
[1] 51
> mean(l9)
[1] 4.248162
> REGRESSION = lm(l9~gb)
> REGRESSION

Call:
lm(formula = l9 ~ gb)

Coefficients:
(Intercept)           gb
  4.2574350   -0.0001818

> plot(gb,l9,col="orange")
> abline(REGRESSION)
>
```

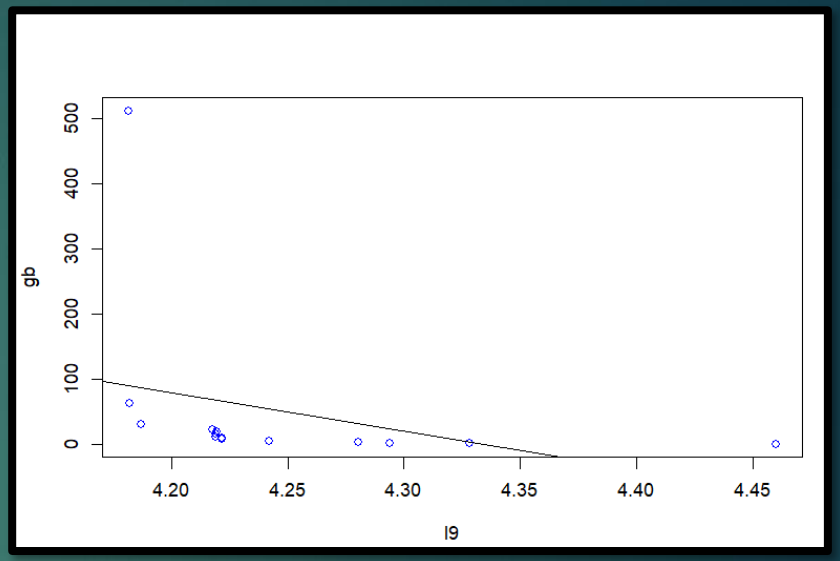


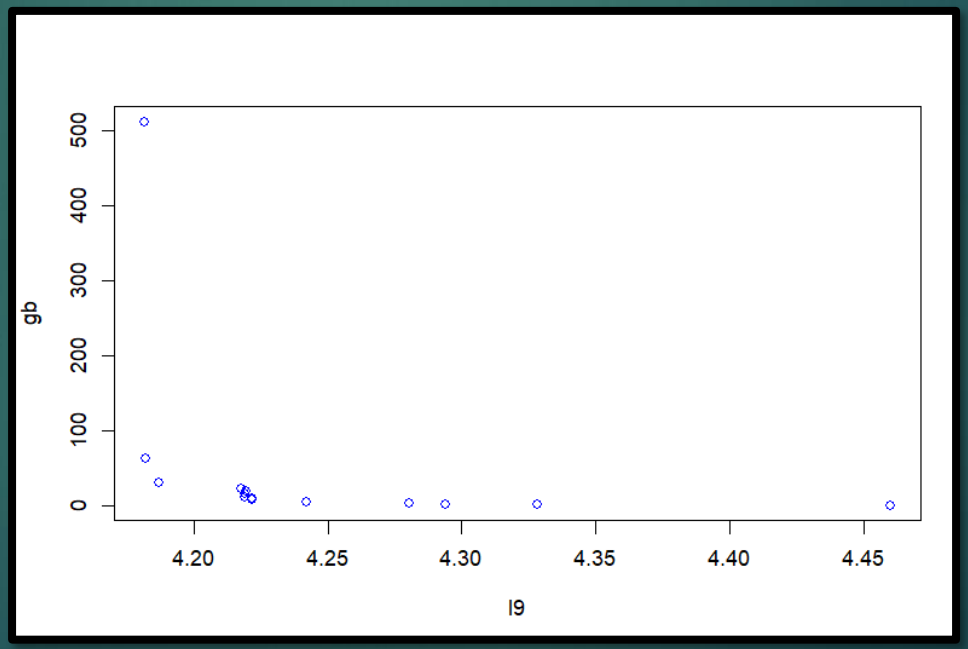


TO FIND LINEAR REGRESSION COEFFICIENT gb ON l9:-

```
#TO FIND LINEAR REGRESSION COEFFICIENT gb ON l9
REGRESSION = lm(gb~l9)
REGRESSION
plot(l9,gb,col="blue")
abline(REGRESSION)
```

```
> #TO FIND LINEAR REGRESSION COEFFICIENT gb ON l9
> REGRESSION = lm(gb~l9)
> REGRESSION

Call:
lm(formula = gb ~ l9)

Coefficients:
(Intercept)           l9
     2551.8       -588.7

> plot(l9,gb,col="blue")
> abline(REGRESSION)
>
```





MULTIPLE REGRESSION:-

##MULTIPLE REGRESSION OF gb ON l1 AND l3 & l5 & l7
REGRESSION = lm(gb~l1+l3+l5+l7)
REGRESSION
summary(REGRESSION)

```
> ##MULTIPLE REGRESSION OF gb ON l1 AND l3 & l5 & l7
> REGRESSION = lm(gb~l1+l3+l5+l7)
> REGRESSION

Call:
lm(formula = gb ~ l1 + l3 + l5 + l7)

Coefficients:
(Intercept)           l1           l3           l5           l7
    43.8039       0.0232      -0.3184      -0.1663       0.1955

> summary(REGRESSION)

Call:
lm(formula = gb ~ l1 + l3 + l5 + l7)

Residuals:
    Min      1Q  Median      3Q     Max
-172.57  -19.13   -3.11    6.93  312.67

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 43.80394  499.88564   0.088   0.9321
l1           0.02320    0.05449   0.426   0.6802
l3          -0.31835    0.73882  -0.431   0.6767
l5          -0.16633    0.08014  -2.076   0.0677 .
l7           0.19552    0.10495   1.863   0.0954 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 127.7 on 9 degrees of freedom
Multiple R-squared:  0.369,     Adjusted R-squared:  0.08862
F-statistic: 1.316 on 4 and 9 DF,  p-value: 0.335
```

```
##MULTIPLE REGRESSION OF l1 ON gb AND l3 & l5 & l7
REGRESSION = lm(l1~gb+l3+l5+l7)
REGRESSION
summary(REGRESSION)
```

```
> ##MULTIPLE REGRESSION OF l1 ON gb AND l3 & l5 & l7
> REGRESSION = lm(l1~gb+l3+l5+l7)
> REGRESSION

Call:
lm(formula = l1 ~ gb + l3 + l5 + l7)

Coefficients:
(Intercept)           gb           l3           l5           l7
 -7159.9747       0.8512      13.5628       0.4816      -0.5166

> summary(REGRESSION)

Call:
lm(formula = l1 ~ gb + l3 + l5 + l7)

Residuals:
     Min       1Q   Median       3Q      Max
-1606.10  -136.82    18.24   147.83  1520.25

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.160e+03  1.865e+03  -3.838  0.00398 **
gb           8.512e-01  1.999e+00   0.426  0.68024
l3           1.356e+01  3.339e-02 406.175  < 2e-16 ***
l5           4.816e-01  5.680e-01   0.848  0.41849
l7          -5.166e-01  7.282e-01  -0.709  0.49604
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 773.2 on 9 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:  0.9999
F-statistic: 4.717e+04 on 4 and 9 DF,  p-value: < 2.2e-16
```

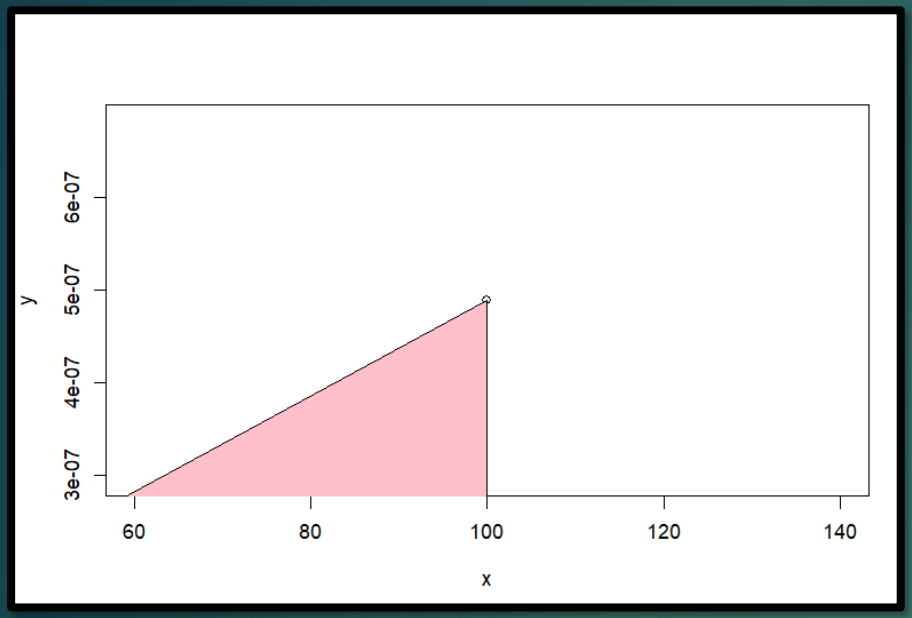
```
##MULTIPLE REGRESSION OF l3 ON l1 AND gb & l5 & l7
REGRESSION = lm(l3~l1+gb+l5+l7)
REGRESSION
summary(REGRESSION)
```

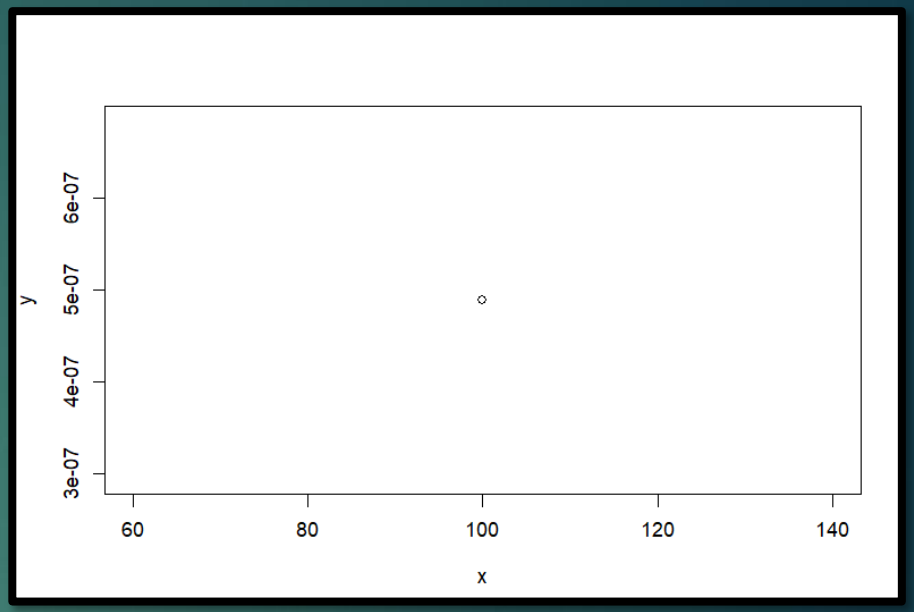
```
> ##MULTIPLE REGRESSION OF l3 ON l1 AND gb & l5 & l7
> REGRESSION = lm(l3~l1+gb+l5+l7)
> REGRESSION

Call:
lm(formula = l3 ~ l1 + gb + l5 + l7)

Coefficients:
(Intercept)           l1           gb           l5           l7
  527.25108      0.07373     -0.06349     -0.03576      0.03845

> summary(REGRESSION)

Call:
lm(formula = l3 ~ l1 + gb + l5 + l7)

Residuals:
     Min       1Q   Median       3Q      Max
-112.115  -11.148   -1.277   10.072  118.379

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.273e+02  1.378e+02   3.826  0.00405 **
l1           7.373e-02  1.815e-04 406.175  < 2e-16 ***
gb          -6.349e-02  1.474e-01  -0.431  0.67668
l5          -3.576e-02  4.185e-02  -0.854  0.41508
l7           3.845e-02  5.366e-02   0.716  0.49189
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 57.01 on 9 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:  0.9999
F-statistic: 4.719e+04 on 4 and 9 DF,  p-value: < 2.2e-16
```

NORMAL DISTRIBUTION:-

```
#A sequence of l[1] (1 gb mobiles)with x=-5 to 5 with the
#mean is l1[1]and standard
#deviation is l1[2], then find the area under the normal curve to
#i. the right of x=100, and
#ii. the left of x=100
#Area under the normal curve to the right of x=2

x=seq(100,l[1],length=1)
x
y=dnorm(x,l1[1],l2[1])
y
plot(x,y,type='o',col='black')
polygon(c(100,x,5),c(0,y,0),col='pink')
pnorm(500,mean=l1[1],sd=l2[1])-pnorm(100,mean=l1[1],sd=l2[1])
```

```
> #A sequence of l[1] (1 gb mobiles)with x=-5 to 5 with the
> #mean is l1[1]and standard
> #deviation is l1[2], then find the area under the normal curve to
> #i. the right of x=100, and
> #ii. the left of x=100
> #Area under the normal curve to the right of x=2
>
> x=seq(100,l[1],length=1)
> x
[1] 100
> y=dnorm(x,l1[1],l2[1])
> y
[1] 4.881949e-07
> plot(x,y,type='o',col='black')
> polygon(c(100,x,5),c(0,y,0),col='pink')
> pnorm(500,mean=l1[1],sd=l2[1])-pnorm(100,mean=l1[1],sd=l2[1])
[1] 0.0001955051
>
```


y
3e-07
4e-07
5e-07
6e-07
60
80
100
120
140
x


y
3e-07
4e-07
5e-07
6e-07
60
80
100
120
140
x