MAT1011
APPLIED STATISTICS

FINAL LAB REPORT

NAME: B. MOHAN SRINIVASA SARMA

REG NO: 19BCN7015

SLOT: L6

Guided by: Dr. Santanu Mandal

Index

Day 1: Data Analysis using R

1. Simple Operations

a) Enter the data {2,5,3,7,1,9,6} directly and store it in a variable x.

Code:

```
> x<-c(2,5,3,7,1,9,6)
> x
```

Output:

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

b) Find the number of elements in x, i.e. in the data list.

Code:

```
> length(x)
```

Output:

```
[1] 7
```

c) Find the last element of x.

Code:

```
> x[7]
```

Output:

```
[1] 6
```

#or

Code:

```
> x[length(x)]
```

Output:

```
[1] 6
```

d) Find the minimum element of x.

Code:

```
> min(x)
```

Output:

```
[1] 1
```

e) Find the maximum element of x.

Code:

```
max(x)
```

Output:

```
[1] 9
```

2. Enter the data {1, 2, ... ,19,20} in a variable x.

Code:

```
> x<-1:20
> x
```

Output:

```
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

a) Find the 3rd element in the data list.

Code:

```
> x[3]
```

Output:

```
[1] 3
```

b) Find 3rd to 5th element in the data list.

Code:

```
> x[3:5]
```

Output:

```
[1] 3 4 5
```

c) Find 2nd, 5th, 6th, and 12th element in the list.

Code:

```
> x[c(2,5,6,12)]
>x
```

Output:

```
[1]  2  5  6 12
```

d) Print the data as {20, 19, ..., 2, 1} without again entering the data.

Ans:

```
> x<-20:1
> x
```

Output:

```
 [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

3.

a) Create a data list (4, 4, 4, 4, 3, 3, 3, 5, 5, 5) using ‘rep’ function.

Code:

```
> x<-c(rep(4,4),rep(3,3),rep(5,3))
> x
```

Output:

```
 [1] 4 4 4 4 3 3 3 5 5 5
```

b) Create a list (4, 6, 3, 4, 6, 3, ..., 4, 6, 3) where there 10 occurrences of 4, 6, and 3 in the given order.

Code:

```
> x<-c(rep(x,10))
> x
```

Output:

```
 [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3
```

c) Create a list (3, 1, 5, 3, 2, 3, 4, 5, 7,7, 7, 7, 7,7, 6, 5, 4, 3, 2, 1, 34, 21, 54) using one line command.

Code:

```
> x<-c(3,1,5,3,2:4,rep(7,6),6:1,34,21,54)
> x
 [1]  3  1  5  3  2  3  4  7  7  7  7  7  7  6  5  4  3  2  1 34 21 54
```

d) First create a list (2, 1, 3, 4). Then append this list at the end with another list (5, 7, 12, 6, -8). Check whether the number of elements in the augmented list is 11.

Code:

```
>  x<-c(2,1,3,4)
> x<-c(5,7,12,6,-8,x)
> x
[1]  5  7 12  6 -8  2  1  3  4
> length(x)==11
```

Output:

```
[1] FALSE
```

4.

(a) Print all numbers starting with 3 and ending with 7 with an increment of 0:5. Store these numbers in x.

Code:

```
> x<-seq(3,7,0.5)
> x
[1] 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0
```

(b)Print all even numbers between 2 and 14 (both inclusive)

```
> seq(2,14,2)
[1]  2  4  6  8 10 12 14
```

(a) Type 2*x and see what you get. Each element of x is multiplied by 2.

Code:

```
> 2*x
[1]  6  7  8  9 10 11 12 13 14   #The recent x stored was
#seq(3,7,0.5)
```

5. Few simple statistical measures:

(a) Enter data as 1,2, ... ,10.

Ans:

```
> x<-1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
```

(b)Find sum of the numbers.

Ans:

```
> sum(x)
[1] 55
```

(c) Find mean, median.

```
> mean(x)
[1] 5.5
> median(x)
```

```
[1] 5.5
```

(d)Find sum of squares of these values.

Ans:

```
>sum(x)
[1] 385
```

e) Find the value of $1/\mathrm{n}(\Sigma_{i=1ton}|xi-\overline{x}|)$ This is known as mean deviation about mean ($MD\overline{x}$).

Ans:

```
> sum(abs(x-mean(x)))/length(x)
[1] 2.5
```

(f) Check whether $MD\overline{x}$ is less than or equal to standard deviation

Ans:

```
> sum(abs(x-mean(x)))/length(x)<=sd(x)
[1] TRUE
```

Day 2: Statistical measures and Graph plot for a set of data

1. Reading a data file and working with it:

a) Read the file first and store it in a.

Ans:

```
> a<-read.csv(file="house_data_1.csv",header=TRUE)
> a
```

Output:

```
   Price FloorArea Rooms Age CentralHeating
1  52.00      1225     3 6.2             no
2  54.75      1230     3 7.5             no
3  57.50      1200     3 4.2             no
4  57.50      1000     2 8.8             no
5  59.75      1420     4 1.9            yes
6  62.50      1450     3 5.2             no
7  64.75      1380     4 6.6            yes
8  67.25      1510     4 2.3             no
9  67.50      1400     5 6.1             no
10 69.75      1550     6 9.2             no
11 70.00      1720     6 4.3            yes
12 75.50      1700     5 4.3             no
13 77.50      1660     6 1.0            yes
14 78.00      1800     7 7.0            yes
15 81.25      1830     6 3.6            yes
16 82.50      1790     6 1.7            yes
```

```
17 86.25      2010     6 1.2          yes
18 87.50      2000     6 0.0          yes
19 88.00      2100     8 2.3          yes
20 92.00      2240     7 0.7          yes
```

b) How many rows are there in this table? How many columns are there?

Ans:

```
> nrow(a)
[1] 20
> ncol(a)
[1] 5
```

c) How to find the number of rows and number of columns by a single command?

Ans:

```
> dim(a)
[1] 20  5
```

d) What are the variables in the data file?

Ans:

```
> names(a)
[1] "Price"        "FloorArea"     "Rooms"         "Age"
[5] "CentralHeating"
```

e) If the file is very large, naturally we cannot simply type `a', because it will cover the entire screen and we won't be able to understand anything. So how to see the top or bottom few lines in this file?

Ans:

```
> head(a)
  Price FloorArea Rooms Age CentralHeating
1 52.00      1225     3 6.2             no
2 54.75      1230     3 7.5             no
3 57.50      1200     3 4.2             no
4 57.50      1000     2 8.8             no
5 59.75      1420     4 1.9            yes
6 62.50      1450     3 5.2             no
> tail(a)
   Price FloorArea Rooms Age CentralHeating
15 81.25      1830     6 3.6            yes
16 82.50      1790     6 1.7            yes
17 86.25      2010     6 1.2            yes
18 87.50      2000     6 0.0            yes
19 88.00      2100     8 2.3            yes
20 92.00      2240     7 0.7            yes
```

f) If the number of columns is too large, again we may face the same problem. So how to see the first 5 rows and first 3 columns?

Ans:

```
> a[1:5,1:3]
```

```
  Price FloorArea Rooms
1 52.00      1225     3
2 54.75      1230     3
3 57.50      1200     3
4 57.50      1000     2
5 59.75      1420     4
```

g) How to get 1st, 3rd, 6th, and 10th row and 2nd, 4th, and 5th column?

Ans:

```
> a[c(1,3,6,10),c(2,4,5)]
   FloorArea Age CentralHeating
1       1225 6.2             no
3       1200 4.2             no
6       1450 5.2             no
10       1550 9.2             no
```

h) How to get values in a specific row or a column?

Ans:

```
> a[,4]
 [1] 6.2 7.5 4.2 8.8 1.9 5.2 6.6 2.3 6.1 9.2 4.3 4.3 1.0 7.0
3.6 1.7 1.2 0.0 2.3 0.7
```

2. Calculate simple statistical measures using the values in the data file.

a) Find means, medians, standard deviations of Price, Floor Area, Rooms, and Age.

Ans:

```
> mean(a$Price)
[1] 71.5875
> mean(a$FloorArea)
[1] 1610.75
> mean(a$Rooms)
[1] 5
> mean(a$Age)
[1] 4.205
> median(a$Price)
[1] 69.875
> median(a$FloorArea)
[1] 1605
> median(a$Rooms)
[1] 5.5
> median(a$Age)
[1] 4.25
> sd(a$Price)
[1] 12.21094
> sd(a$FloorArea)
[1] 331.9649
> sd(a$Rooms)
```

```
[1] 1.65434
> sd(a$Age)
[1] 2.786523
```

b) How many houses have central heating and how many don't have?

Ans:

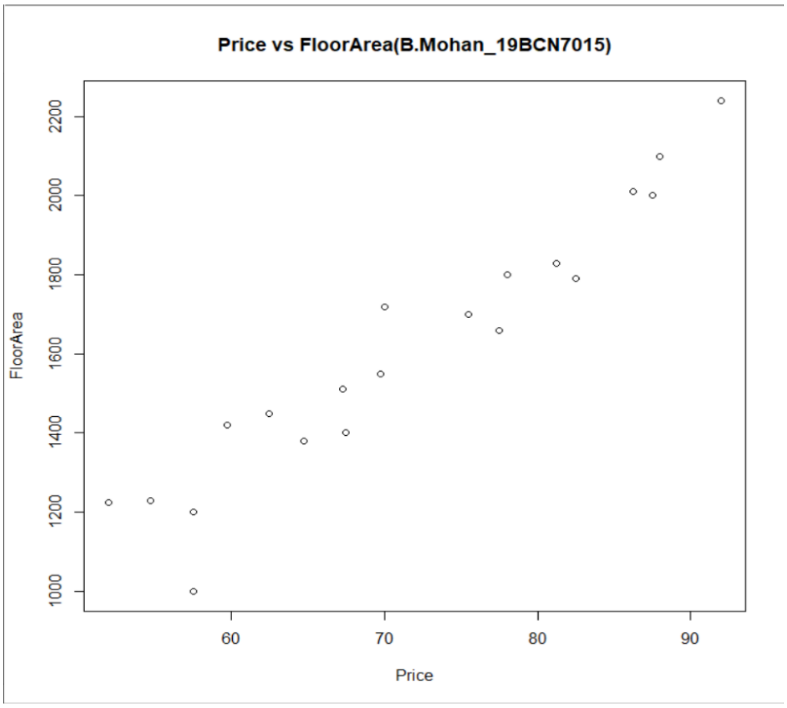
```
> sum(a$CentralHeating=='yes')
[1] 11
```

c) Plot Price vs. Floor, Price vs. Age, and Price vs. rooms, in separate graphs.

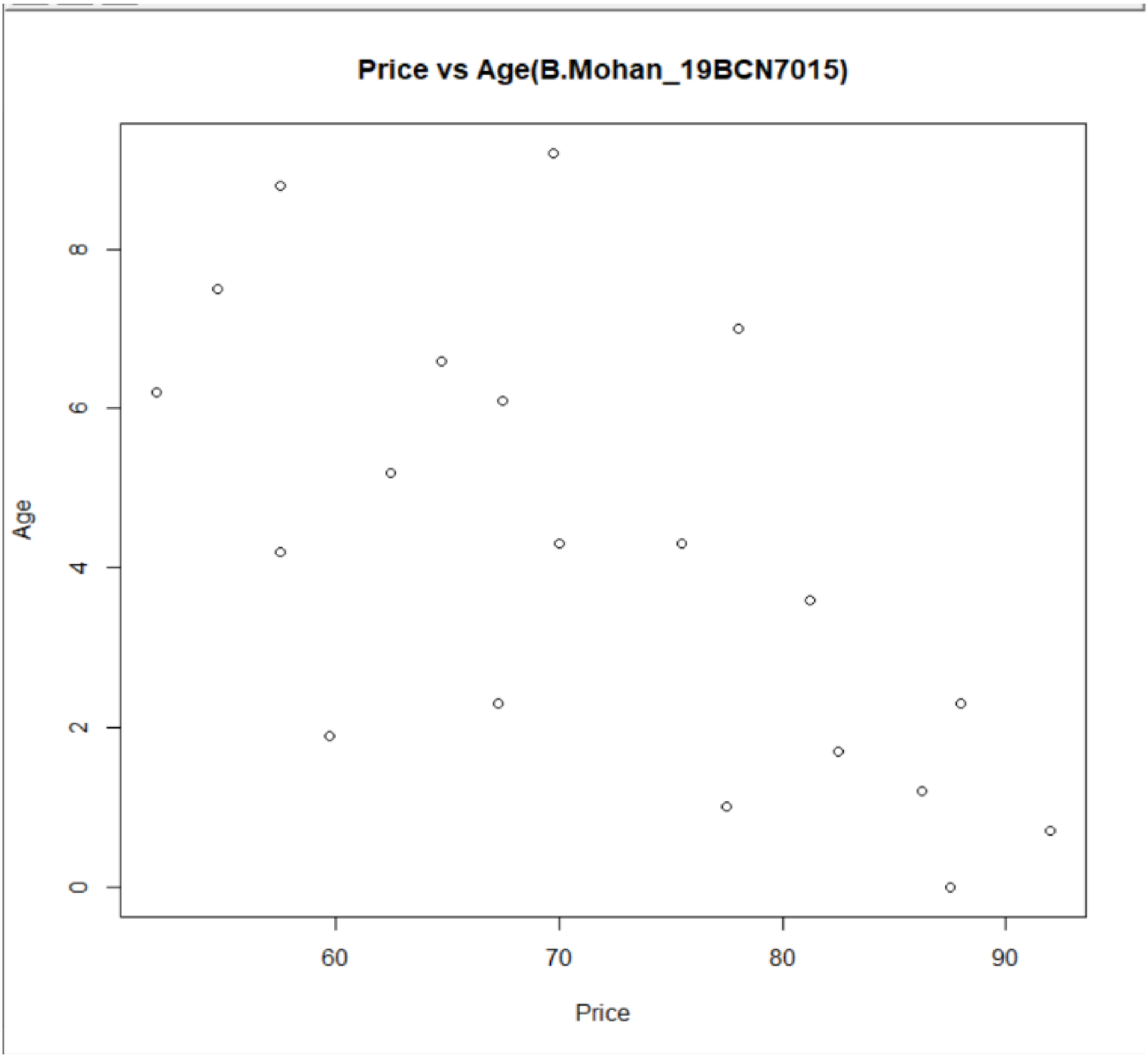
Ans:

```
>plot(a$Price,a$FloorArea,xlab="Price",ylab="FloorArea",main="Price vs FloorArea(B.Mohan_19BCN7015)")
```

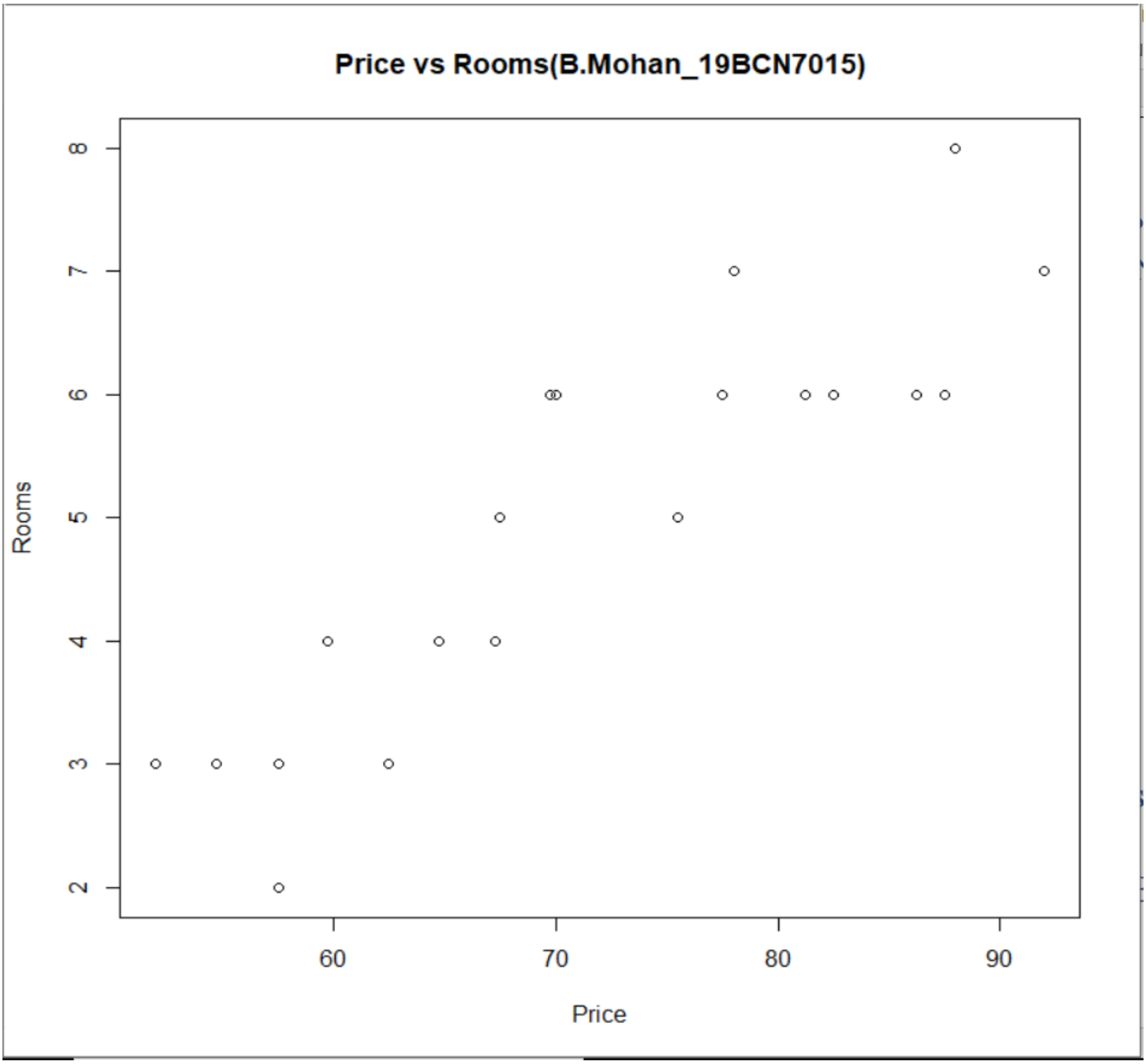
Output:



```
> plot(a$Price,a$Age,xlab="Price",ylab="Age",main="Price vs
Age(B.Mohan_19BCN7015)")
```



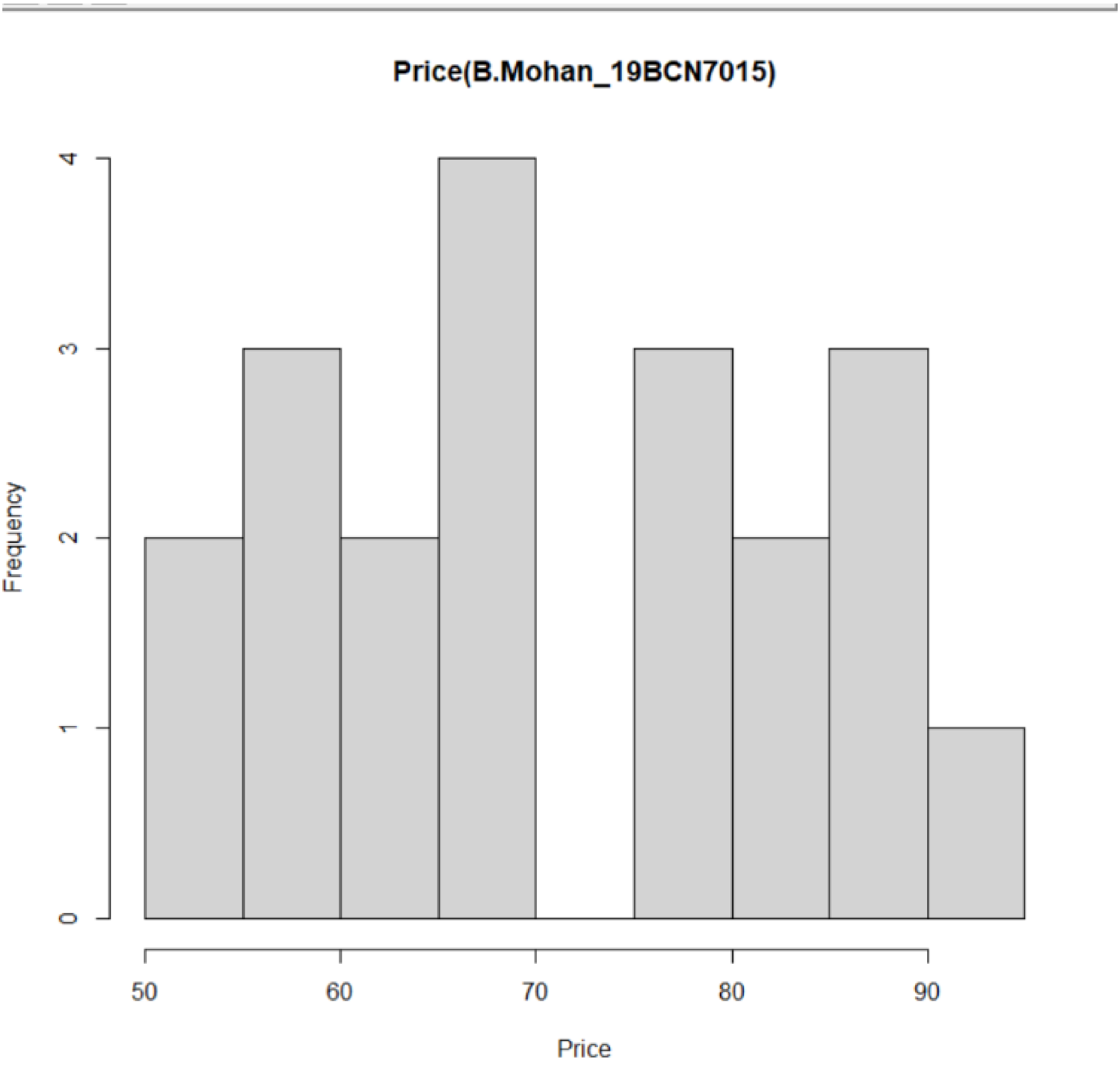
```
>plot(a$Price,a$Rooms,xlab="Price",ylab="Rooms",main="Pri
ce vs Rooms(B.Mohan_19BCN7015)")
```



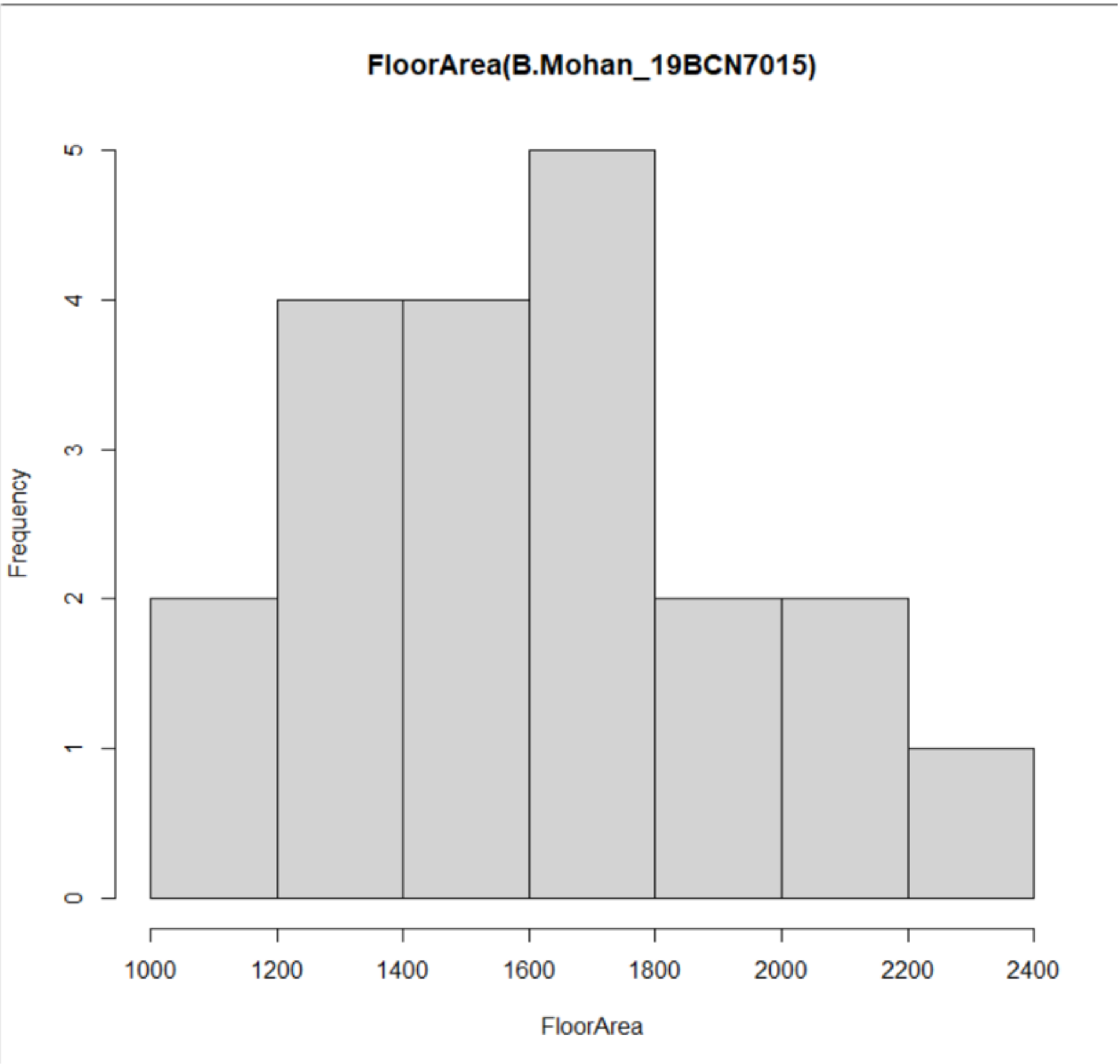
d) Draw histograms of Prices, FloorArea, and Age.

Ans:

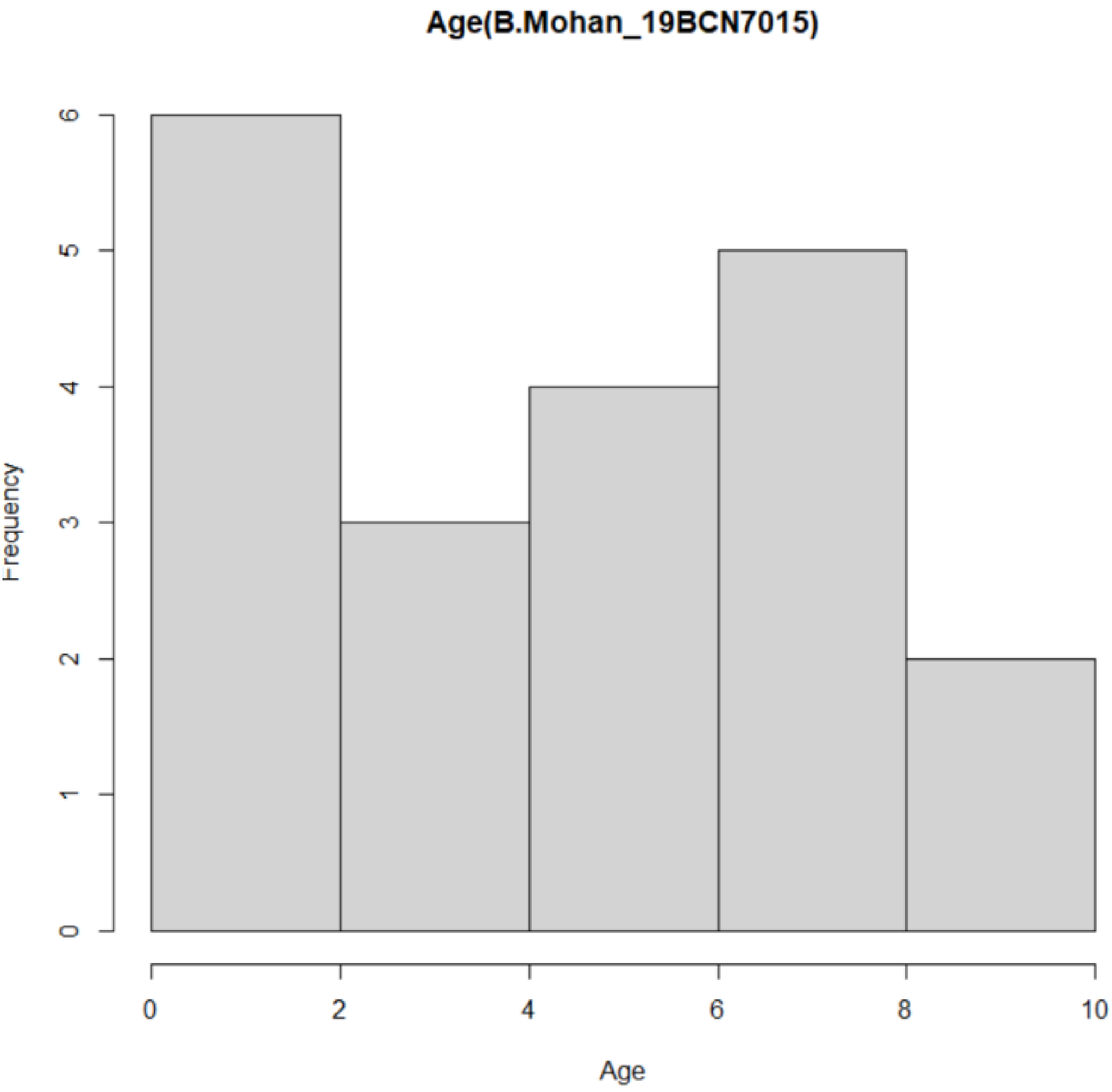
```
>hist(a$Price,xlab="Price",main="Price(B.Mohan_19BCN7015
)")
```



```
>hist(a$FloorArea,xlab="FloorArea",main="FloorArea(B.Moh
an_19BCN7015)")
```



```
>hist(a$Age,xlab="Age",main="Age(B.Mohan_19BCN7015)")
```

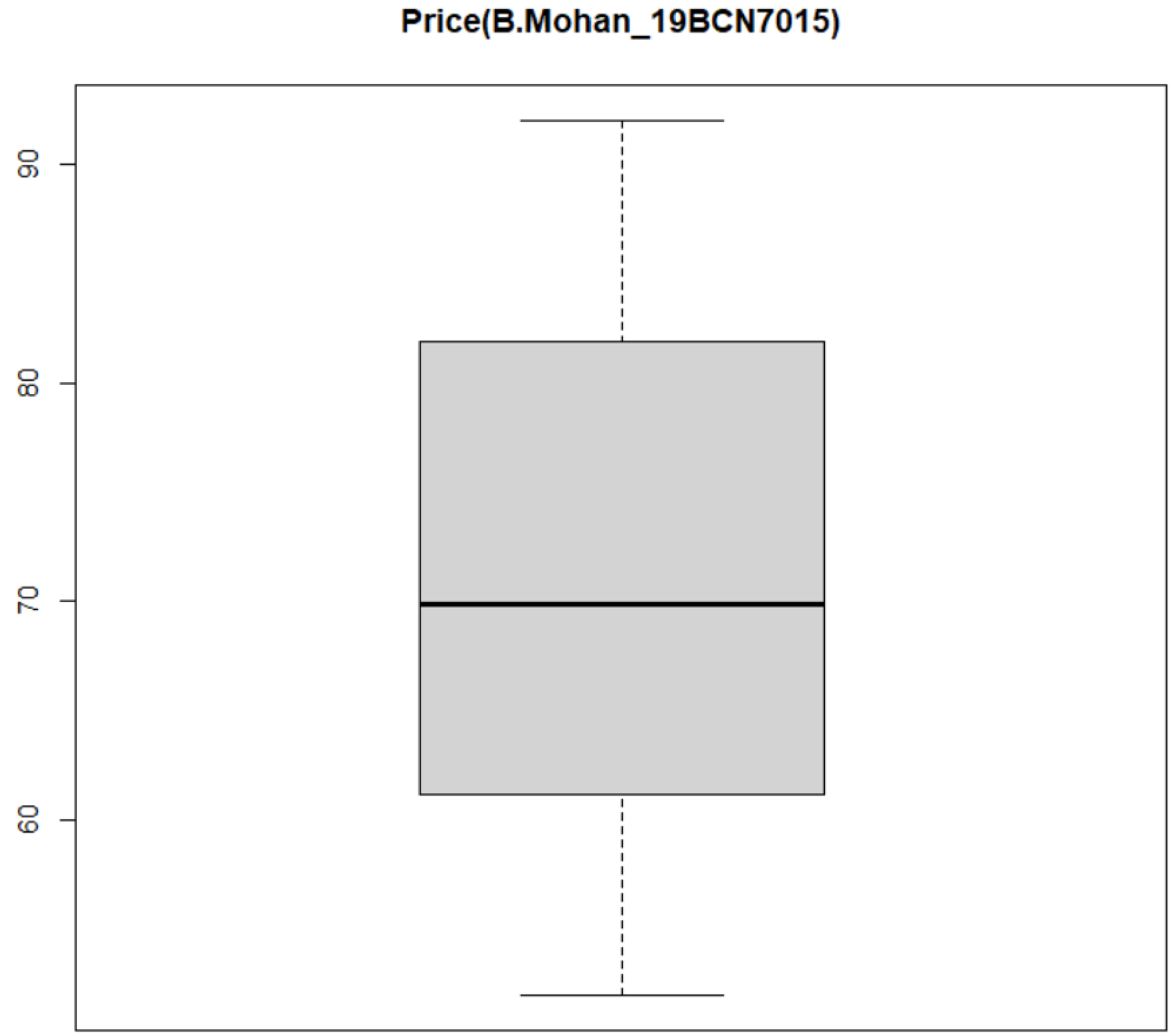


e) Draw box plots of Price, FloorArea, and Age.

Code:

```
>boxplot(a$Price,main="Price(B.Mohan_19BCN7015)")
```

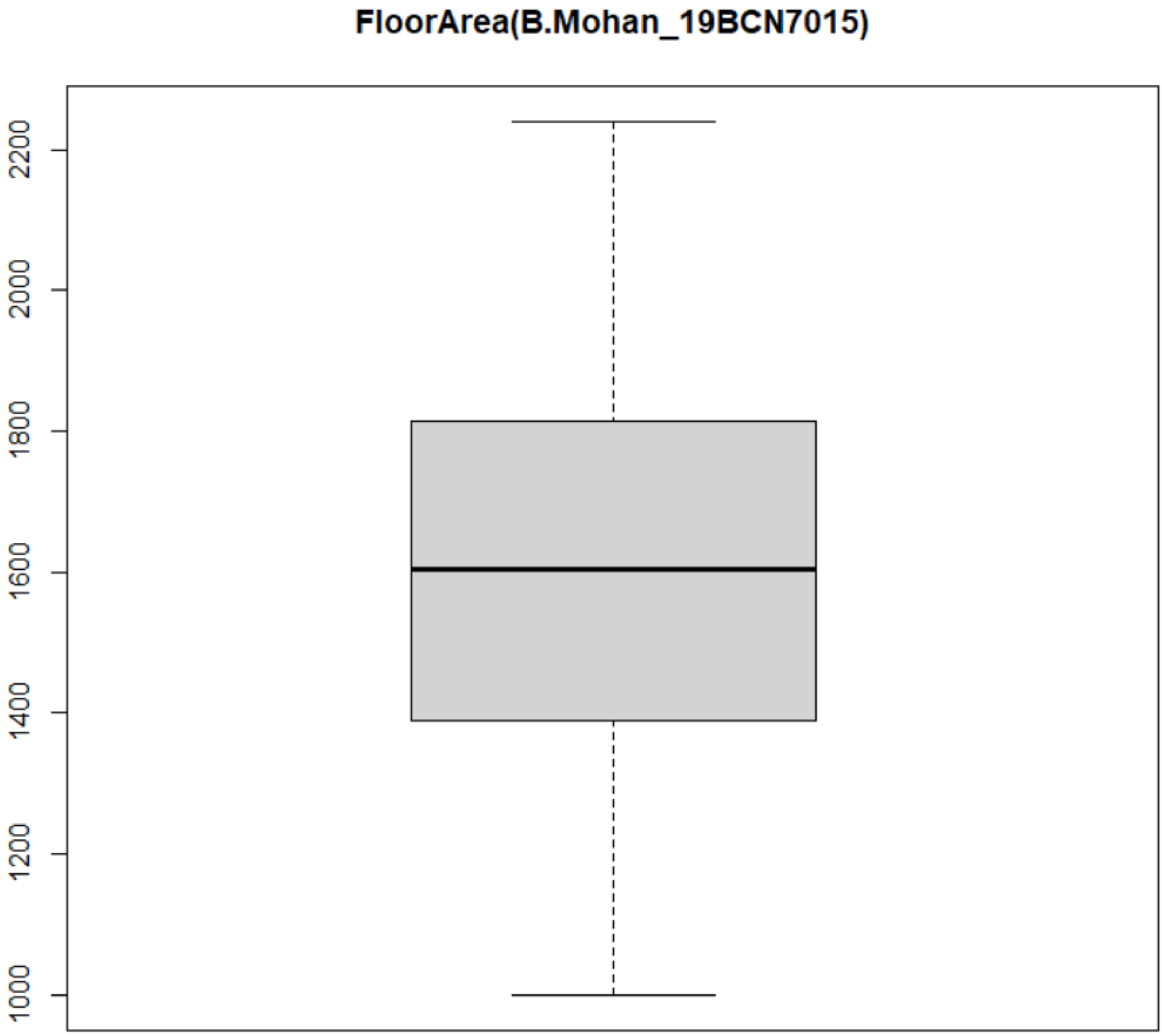
Output:



Code:

```
>boxplot(a$FloorArea,main="FloorArea(B.Mohan_19BCN701
5)")
```

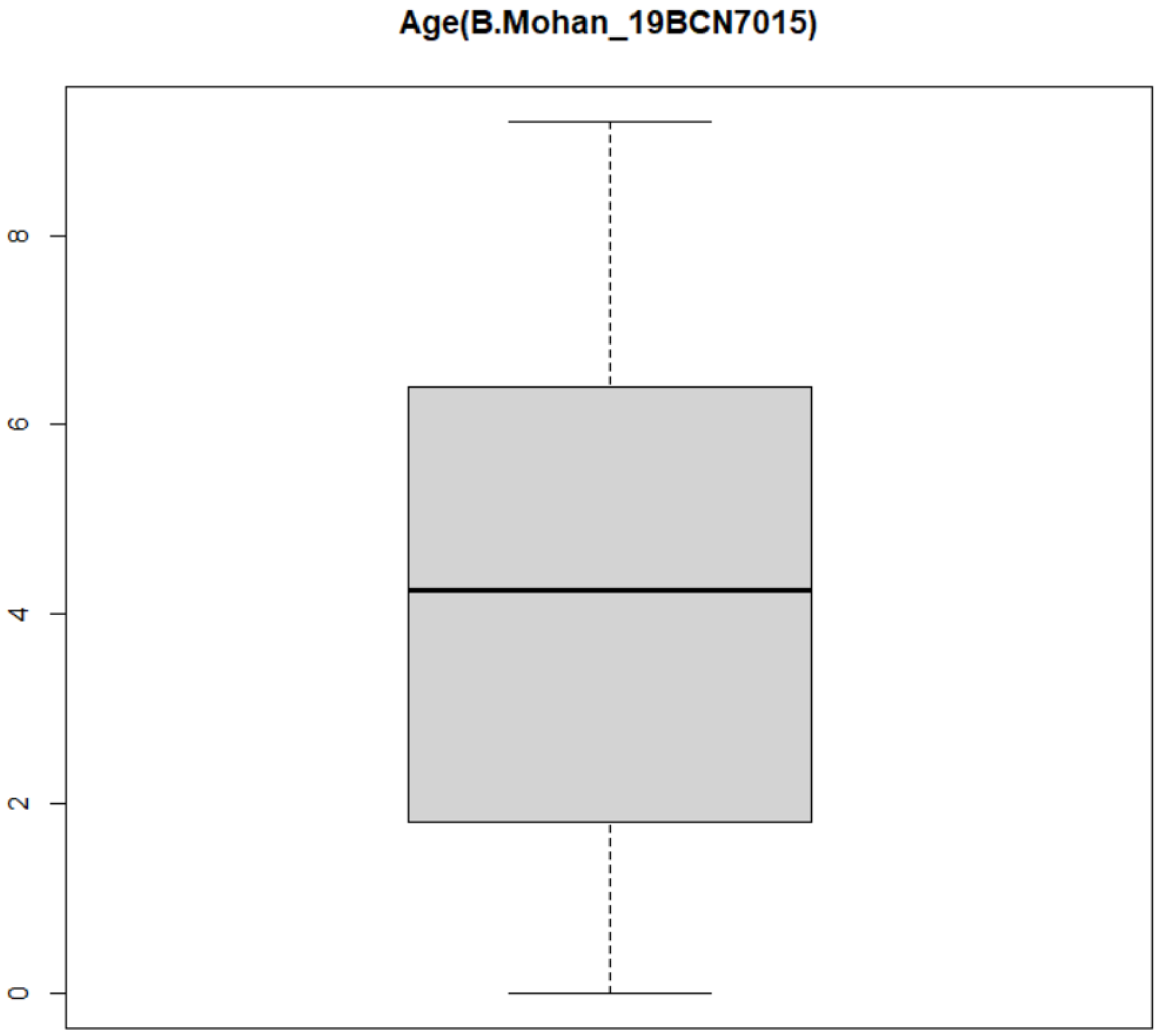
Output:



Code:

```
> boxplot(a$Age,main="Age(B.Mohan_19BCN7015)")
```

Output:

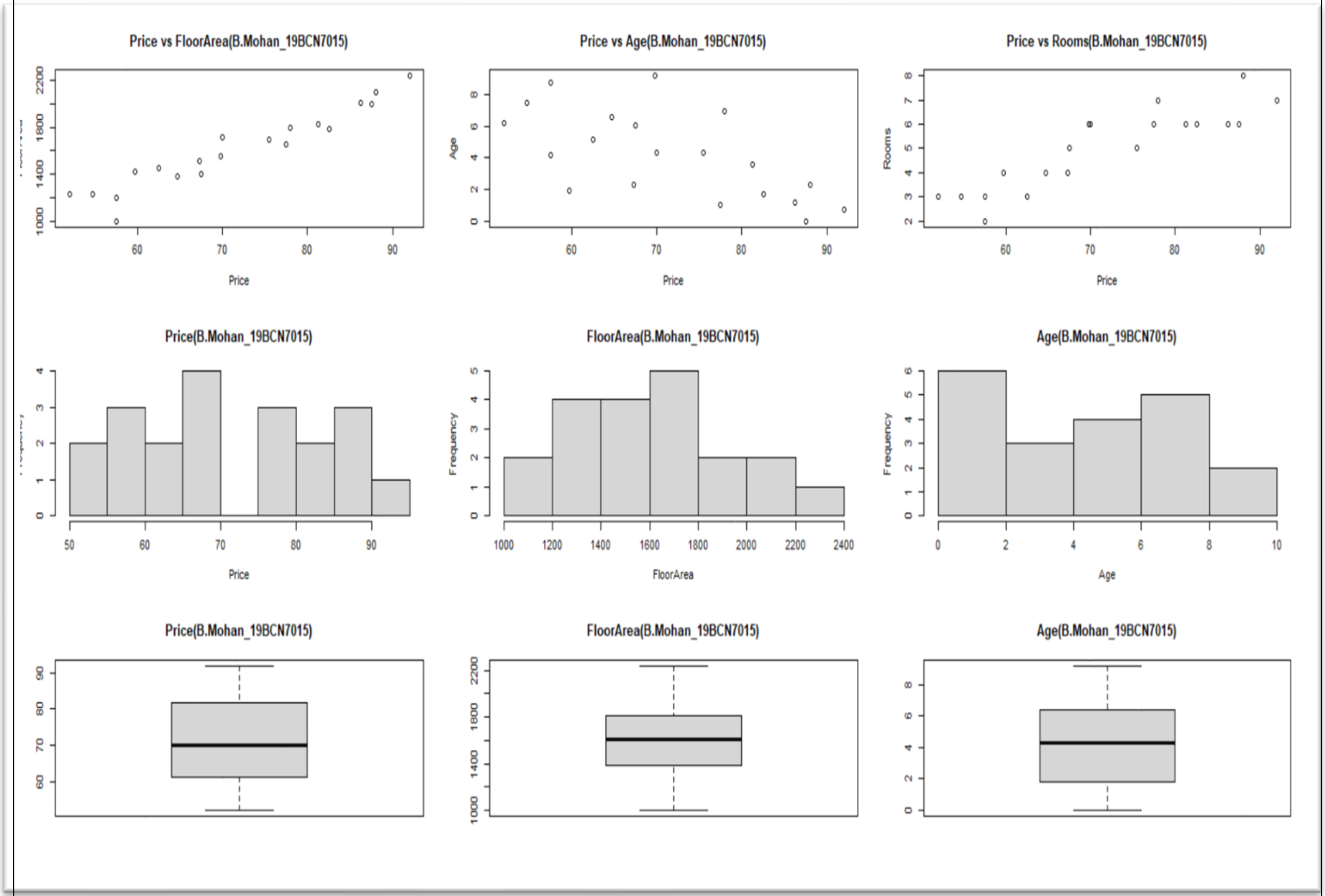


f) Draw all the graphs in (c), (d), and (e) in the same graph paper.

Ans:

```
> par(mfrow=c(3,3))

>plot(a$Price,a$FloorArea,xlab="Price",ylab="FloorArea",main="
Price vs FloorArea(B.Mohan_19BCN7015)")

> plot(a$Price,a$Age,xlab="Price",ylab="Age",main="Price vs
Age(B.Mohan_19BCN7015)")

>plot(a$Price,a$Rooms,xlab="Price",ylab="Rooms",main="Price
vs Rooms(B.Mohan_19BCN7015)")

>hist(a$Price,xlab="Price",main="Price(B.Mohan_19BCN7015)")
>hist(a$FloorArea,xlab="FloorArea",main="FloorArea(B.Mohan_
19BCN7015)")

>hist(a$Age,xlab="Age",main="Age(B.Mohan_19BCN7015)")

> boxplot(a$Price,main="Price(B.Mohan_19BCN7015)")

>boxplot(a$FloorArea,main="FloorArea(B.Mohan_19BCN7015)")

> boxplot(a$Age,main="Age(B.Mohan_19BCN7015)")
```

Plotting all Graphs:



3.Augmenting the file and saving the resultant file:

a) Calculate the value per square foot area of each apartment and store it in a vector named “PriceSqFt”.

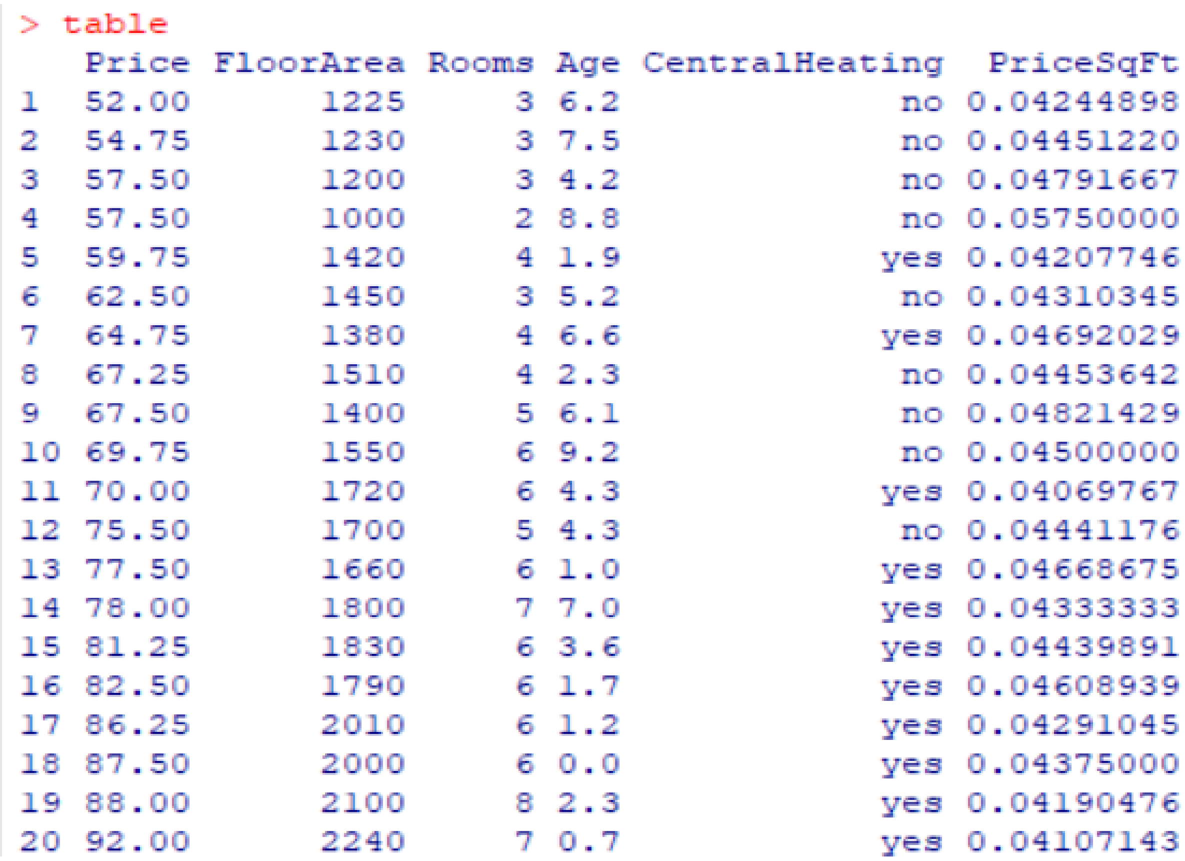
Ans:

```
> a<-read.csv(file="house_data_1.csv",header=TRUE)

> PriceSqFt<-(a$Price/a$FloorArea)

> PriceSqFt
 [1] 0.04244898 0.04451220 0.04791667 0.05750000
0.04207746 0.04310345 0.04692029 0.04453642
0.04821429 0.04500000 0.04069767 0.04441176
0.04668675 0.04333333 0.04439891 0.04608939
[17] 0.04291045 0.04375000 0.04190476 0.04107143
```

b) Place this vector after the last column in the data file.

Ans:

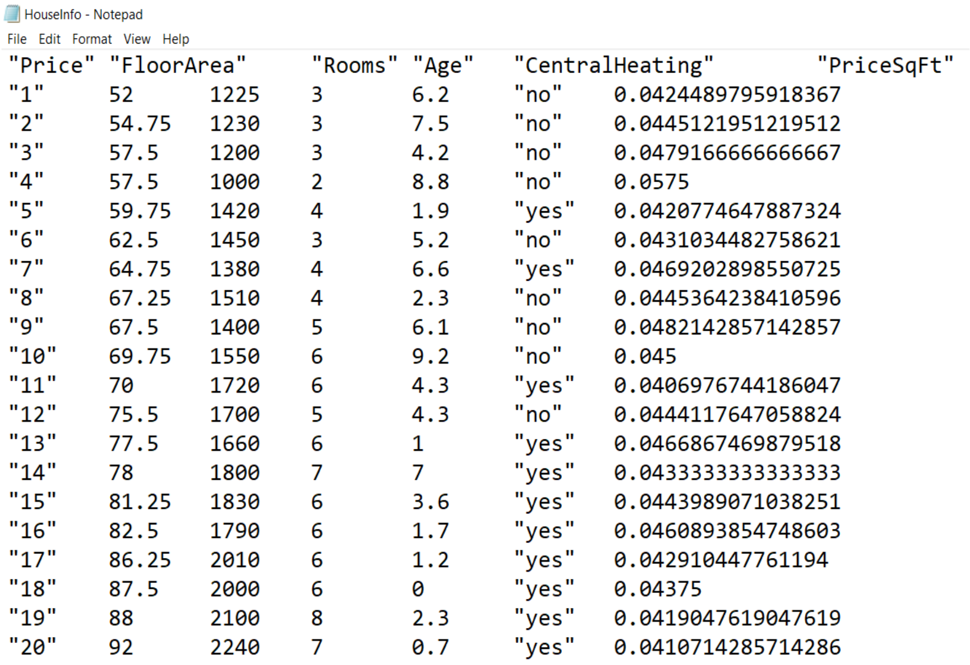
```
> table<-cbind(a,PriceSqFt)
>table
```



```
> table
   Price FloorArea Rooms Age CentralHeating  PriceSqFt
1  52.00      1225     3 6.2             no 0.04244898
2  54.75      1230     3 7.5             no 0.04451220
3  57.50      1200     3 4.2             no 0.04791667
4  57.50      1000     2 8.8             no 0.05750000
5  59.75      1420     4 1.9            yes 0.04207746
6  62.50      1450     3 5.2             no 0.04310345
7  64.75      1380     4 6.6            yes 0.04692029
8  67.25      1510     4 2.3             no 0.04453642
9  67.50      1400     5 6.1             no 0.04821429
10 69.75      1550     6 9.2             no 0.04500000
11 70.00      1720     6 4.3            yes 0.04069767
12 75.50      1700     5 4.3             no 0.04441176
13 77.50      1660     6 1.0            yes 0.04668675
14 78.00      1800     7 7.0            yes 0.04333333
15 81.25      1830     6 3.6            yes 0.04439891
16 82.50      1790     6 1.7            yes 0.04608939
17 86.25      2010     6 1.2            yes 0.04291045
18 87.50      2000     6 0.0            yes 0.04375000
19 88.00      2100     8 2.3            yes 0.04190476
20 92.00      2240     7 0.7            yes 0.04107143
```

c) Save the augmented file under name “HouseInfo.txt”.

Ans:



```
"Price" "FloorArea"     "Rooms" "Age"   "CentralHeating"        "PriceSqFt"
"1"     52      1225    3       6.2     "no"    0.0424489795918367
"2"     54.75   1230    3       7.5     "no"    0.0445121951219512
"3"     57.5    1200    3       4.2     "no"    0.0479166666666667
"4"     57.5    1000    2       8.8     "no"    0.0575
"5"     59.75   1420    4       1.9     "yes"   0.0420774647887324
"6"     62.5    1450    3       5.2     "no"    0.0431034482758621
"7"     64.75   1380    4       6.6     "yes"   0.0469202898550725
"8"     67.25   1510    4       2.3     "no"    0.0445364238410596
"9"     67.5    1400    5       6.1     "no"    0.0482142857142857
"10"    69.75   1550    6       9.2     "no"    0.045
"11"    70      1720    6       4.3     "yes"   0.0406976744186047
"12"    75.5    1700    5       4.3     "no"    0.0444117647058824
"13"    77.5    1660    6       1       "yes"   0.0466867469879518
"14"    78      1800    7       7       "yes"   0.0433333333333333
"15"    81.25   1830    6       3.6     "yes"   0.0443989071038251
"16"    82.5    1790    6       1.7     "yes"   0.0460893854748603
"17"    86.25   2010    6       1.2     "yes"   0.042910447761194
"18"    87.5    2000    6       0       "yes"   0.04375
"19"    88      2100    8       2.3     "yes"   0.0419047619047619
"20"    92      2240    7       0.7     "yes"   0.0410714285714286
```

```
> write.table(table,"E:/HouseInfo.txt",sep="\t")
```

d) Read the file "HouseInfo.txt".

Code:

```
> h<-read.delim("HouseInfo.txt")
> h
```

Output:

```
   Price FloorArea Rooms Age CentralHeating   PriceSqFt
1  52.00       1225         3 6.2                  no  0.04244898
2  54.75       1230         3 7.5                  no  0.04451220
3  57.50       1200         3 4.2                  no  0.04791667
4  57.50       1000         2 8.8                  no  0.05750000
5  59.75       1420         4 1.9                 yes  0.04207746
6  62.50       1450         3 5.2                  no  0.04310345
7  64.75       1380         4 6.6                 yes  0.04692029
8  67.25       1510         4 2.3                  no  0.04453642
9  67.50       1400         5 6.1                  no  0.04821429
10 69.75       1550         6 9.2                  no  0.04500000
11 70.00       1720         6 4.3                 yes  0.04069767
12 75.50       1700         5 4.3                  no  0.04441176
13 77.50       1660         6 1.0                 yes  0.04668675
14 78.00       1800         7 7.0                 yes  0.04333333
15 81.25       1830         6 3.6                 yes  0.04439891
16 82.50       1790         6 1.7                 yes  0.04608939
17 86.25       2010         6 1.2                 yes  0.04291045
18 87.50       2000         6 0.0                 yes  0.04375000
19 88.00       2100         8 2.3                 yes  0.04190476
20 92.00       2240         7 0.7                 yes  0.04107143
```

Day 3: Matrix operations Random Sampling and Probability

Matrices and arrays

a) Matrices and arrays are represented as vectors with dimensions: Create one matrix x with 1 to 12 numbers with 3X4 order.

Ans:

```
> x<-rbind(c(1:4),c(5:8),c(9:12))
> x
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

b) Create same matrix with *matrix* function.

Code:

```
> x<-matrix(1:12,nrow=3,byrow=FALSE)
> x
Output:
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

c) Give name of rows of this matrix with A,B,C.

Ans:

```
> x<-
matrix(1:12,nrow=3,byrow=FALSE,dimname=list(c("A","B","C
"),c("W","X","Y","Z")))
> x
  W X Y  Z
A 1 4 7 10
B 2 5 8 11
C 3 6 9 12
```

d) Transpose of the matrix.

Ans:

```
> t(x)
   A  B  C
W  1  2  3
X  4  5  6
Y  7  8  9
Z 10 11 12
```

e) Use functions *cbind* and *rbind* separately to create different matrices.

Ans:

```
> x<-matrix(c(6:1,5,12,23),nrow=3,byrow=TRUE)
> x
     [,1] [,2] [,3]
[1,]    6    5    4
[2,]    3    2    1
```

```
[3,]    5   12   23
> x<-rbind(x,c(12:14))
> x
     [,1] [,2] [,3]
[1,]    6    5    4
[2,]    3    2    1
[3,]    5   12   23
[4,]   12   13   14
```

```
> x<-matrix(c(1:5,2,1,11,19),nrow=3,byrow=FALSE)
> x
     [,1] [,2] [,3]
[1,]    1    4    1
[2,]    2    5   11
[3,]    3    2   19
> x<-cbind(x,c(14,23,15))
> x
     [,1] [,2] [,3] [,4]
[1,]    1    4    1   14
[2,]    2    5   11   23
[3,]    3    2   19   15
```

f) Use arbitrary numbers to create matrix.

Ans:

```
> x<-matrix(c(1,4,2,5,2,5,6,5,2),nrow=3,byrow=TRUE)
> x
     [,1] [,2] [,3]
[1,]    1    4    2
[2,]    5    2    5
[3,]    6    5    2
```

g) Verify matrix multiplication.

Ans:

```
> m<-matrix(1:9,nrow=3,byrow=FALSE)

#3 rows 3 columns

> n<-matrix(1:6,nrow=3,byrow=TRUE)

#3 rows 2 columns
```

Number of columns of m=number of rows of n

```
> m%*%n

     [,1] [,2]

[1,]   48   60

[2,]   57   72

[3,]   66   84

#not possible as number of columns of n is not equal to
#number of rows of m

> n%*%m

Error in n %*% m : non-conformable arguments
```

Random sampling

a) In R, you can simulate these situations with the *sample* function. Pick five numbers at random from the set 1:40.

Ans:

```
> random<-sample(1:40,5)
> random
[1] 27 21 40 26 28
> random<-sample(1:40,5)
> random
[1] 28 11 10 38  7
```

b) Notice that the default behaviour of *sample* is *sampling without replacement*. That is, the samples will not contain the same number twice, and size obviously cannot be bigger than the length of the vector to be sampled. If you want sampling with replacement, then you need to add the argument *replace=TRUE*. Sampling with replacement is suitable for modelling coin tosses or throws of a die. So, for instance, simulate 10 coin tosses.

Ans:

```
> a<-sample(c("T","H"),10,replace=TRUE)
> a
 [1] "H" "T" "T" "H" "H" "H" "H" "T" "T" "H"
> a<-sample(c("T","H"),10,replace=TRUE)
> a
 [1] "T" "T" "T" "T" "H" "H" "T" "H" "H" "T"
```

c) In fair coin-tossing, the probability of heads should equal the probability of tails, but the idea of a random event is not restricted to symmetric cases. It could be equally well applied to other cases, such as the successful outcome of a surgical procedure. Hopefully, there would be a better than 50% chance of this. Simulate data with nonequal probabilities for the outcomes (say, a 90% chance of success) by using the *prob* argument to sample.

Ans:

```
>
sample(c("Success","Failure"),10,replace=TRUE,prob=c(0.9,
0.1))
 [1] "Success" "Success" "Success" "Success" "Success"
"Success" "Success" "Success"
 [9] "Failure" "Success"
>
sample(c("Success","Failure"),10,replace=TRUE,prob=c(0.9,
0.1))
 [1] "Success" "Success" "Success" "Success" "Failure"
"Success" "Success" "Success"
 [9] "Success" "Success"
>
sample(c("Success","Failure"),10,replace=TRUE,prob=c(0.9,
0.1))
 [1] "Success" "Failure" "Success" "Success" "Success"
"Success" "Success" "Success"
 [9] "Success" "Success"
```

d) The *choose* function can be used to calculate the following express.

$$\binom{40}{5} = \frac{40!}{5!35!}$$

Ans:

```
> choose(40,5)
[1] 658008
#or
> factorial(40)/(factorial(5)*factorial(35))
[1] 658008
```

e) Find 5!

Ans:

```
> factorial(5)
[1] 120
# or
> prod(1:5)
[1] 120
```

Day 4: Binomial Distribution

Ex 1: Five terminals on an online computer system are attached to a communication line to the central computer system. The probability that any terminal is ready to transmit is 0.95.

Let X denote the number of ready terminals

Ex 2: A fair coin is tossed 10 times; success and failure are “heads” and “tails” respectively, each with probability 0.5.

Let X denote the number of heads (Success obtained).

Ex 3: It is known that 20% of integrated circuit chips on a production line are defective. To maintain and monitor the quality of the chips, a sample of twenty chips is selected at regular intervals for inspection.

Let X denote the number of defectives found in the sample.

Ex 4:

It is known that 1% of bits transmitted through a digital transmission are received in error. One hundred bits are transmitted each day.

Let X denote number of bits found in error each day.

1. Find the probability that the third terminals from Ex1 works also Find the individual probabilities for all Terminals and Use round Function to round off to four decimals.

Code:

```
> q1<-round(dbinom(3,size=5,prob=0.95),digits=4)

> q1
```

Output:

```
[1] 0.0214
```

Code:

```
> q2<-round(dbinom(0:5,size=5,prob=0.95),digits=4)

> q2
```

Output:

```
[1] 0.0000 0.0000 0.0011 0.0214 0.2036 0.7738
```

2. Find the probabilities for all examples and use round function to round off to 4 decimals.

Code for Ex 1:

```
> q1<-round(dbinom(0:5,size=5,prob=0.95),digits=4)
> q1
```

Output:

```
[1] 0.0000 0.0000 0.0011 0.0214 0.2036 0.7738
```

Code for Ex 2:

```
> q2<-round(dbinom(0:10,size=10,prob=0.5),digits=4)
> q2
```

Output:

```
 [1] 0.0010 0.0098 0.0439 0.1172 0.2051 0.2461 0.2051 0.1172 0.0439 0.0098
[11] 0.0010
```

Code for Ex 3:

```
> q3<-round(dbinom(0:20,size=20,prob=0.2),digits=4)
> q3
```

Output:

```
 [1] 0.0115 0.0576 0.1369 0.2054 0.2182 0.1746 0.1091
0.0545 0.0222 0.0074
[11] 0.0020 0.0005 0.0001 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
[21] 0.0000
```

Code for Ex 4:

```
>q4<-
round(dbinom(0:100,size=100,prob=0.01),digits=4)
> q4
```

Output:

```
  [1] 0.3660 0.3697 0.1849 0.0610 0.0149 0.0029 0.0005
0.0001 0.0000 0.0000
 [11] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
 [21] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
 [31] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
 [41] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000
```

```
 [51] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000

 [61] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000

 [71] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000

 [81] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000

 [91] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000

[101] 0.0000
```

3. Find the cumulative probabilities of all examples

Code for Ex 1:

```
> q1<-pbinom(0:5,size=5,prob=0.95)

> q1
```

Output:

```
[1] 0.0000003125 0.0000300000 0.0011581250 0.0225925000
0.2262190625 1.0000000000
```

Code for Ex 2:

```
> q2<-pbinom(0:10,size=10,prob=0.5)

> q2
```

Output:

```
 [1] 0.0009765625 0.0107421875 0.0546875000 0.1718750000
0.3769531250 0.6230468750 0.8281250000 0.9453125000
0.9892578125 0.9990234375 1.0000000000
```

Code for Ex 3:

```
> q3<-pbinom(0:20,size=20,prob=0.2)

> q3
```

Output:

```
 [1] 0.01152922 0.06917529 0.20608472 0.41144886
0.62964826 0.80420779 0.91330749 0.96785734 0.99001821
0.99740517 0.99943659 0.99989827 0.99998484 0.99999815
0.99999982 0.99999999

[17] 1.00000000 1.00000000 1.00000000 1.00000000
1.00000000
```

Code for Ex 4:

```
> q4<-pbinom(0:100,size=100,prob=0.01)

> q4
```

Output:

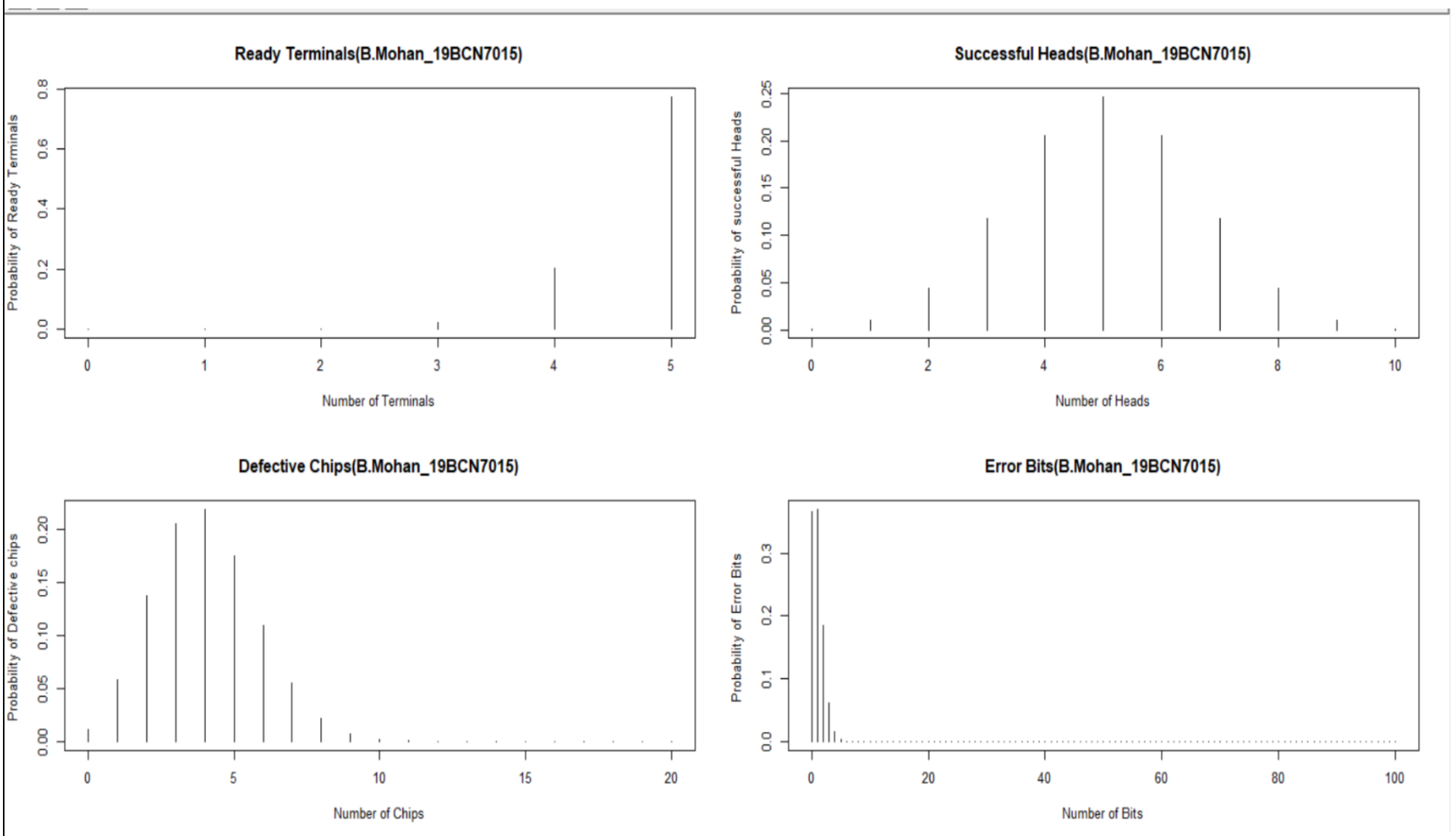
```
  [1] 0.3660323 0.7357620 0.9206268 0.9816260 0.9965677
0.9994655 0.9999289 0.9999918 0.9999992 0.9999999
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000

 [19] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000

 [37] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000

 [55] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000

 [73] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000

 [91] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000
```

4. Plot the Binomial distribution for all examples in one window

Code for plotting:

```
> q1<-dbinom(0:5,size=5,prob=0.95)

> q2<-dbinom(0:10,size=10,prob=0.5)

> q3<-dbinom(0:20,size=20,prob=0.2)

> q4<-dbinom(0:100,size=100,prob=0.01)

> par(mfrow=c(2,2))

> plot(0:5,q1,xlab="Number of Terminals",ylab="Probability of Ready Terminals",main="Ready Terminals(B.Mohan_19BCN7015)",type="h")

> plot(0:10,q2,xlab="Number of Heads",ylab="Probability of successful Heads",main="Successful Heads(B.Mohan_19BCN7015)",type="h")

> plot(0:20,q3,xlab="Number of Chips",ylab="Probability of Defective chips ",main="Defective Chips(B.Mohan_19BCN7015)",type="h")

> plot(0:100,q4,xlab="Number of Bits",ylab="Probability of Error Bits ",main="Error Bits(B.Mohan_19BCN7015)",type="h")
```

Output:

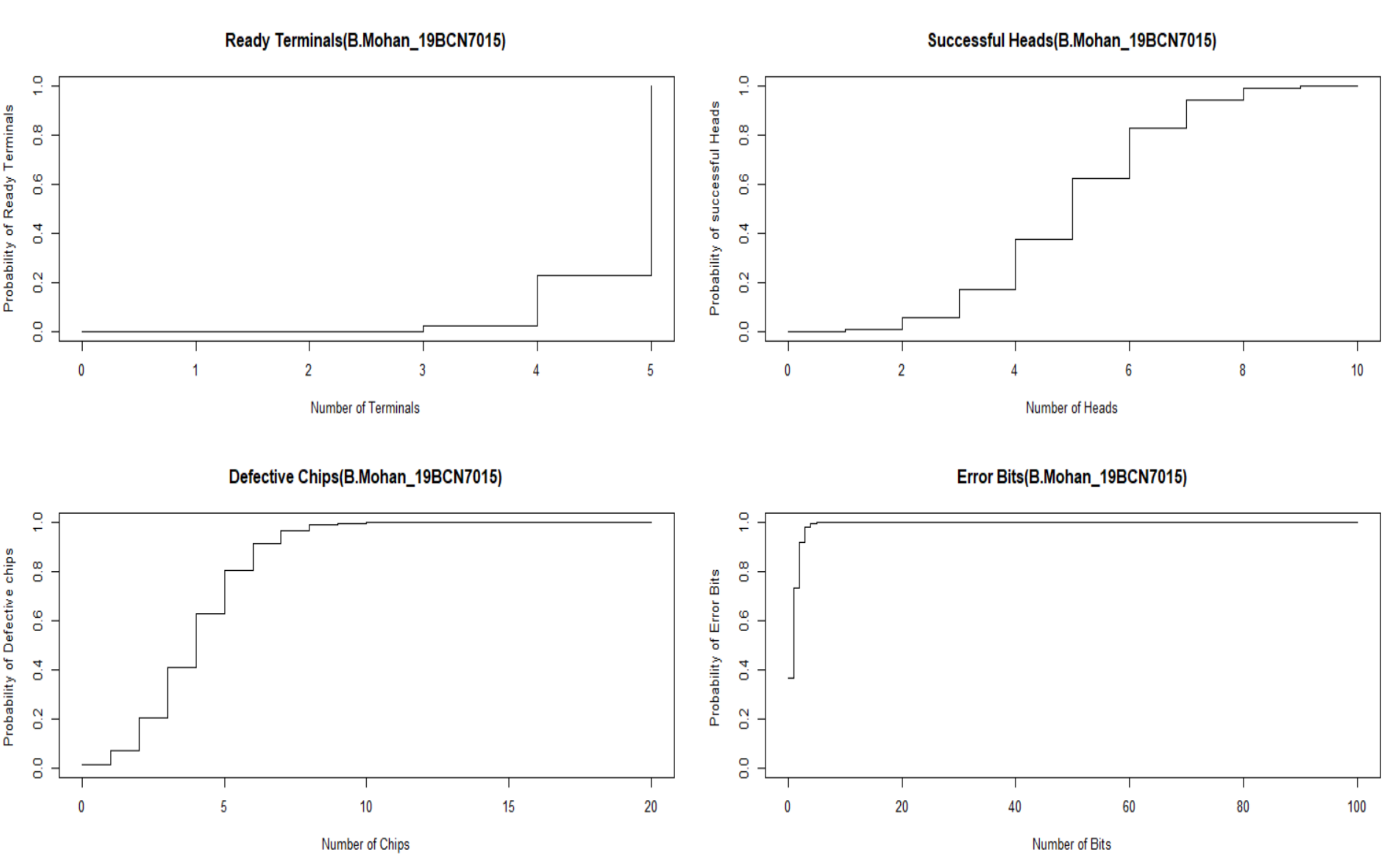


5. Plot the cumulative binomial distribution for all examples as a step function within a bound of 0 to 1

Code for plotting:

```
> q1<-pbinom(0:5,size=5,prob=0.95)

> q2<-pbinom(0:10,size=10,prob=0.5)

> q3<-pbinom(0:20,size=20,prob=0.2)

> q4<-pbinom(0:100,size=100,prob=0.01)
> plot(0:5,q1,xlab="Number of Terminals",ylab="Probability of Ready Terminals",main="Ready Terminals(B.Mohan_19BCN7015)",type="s",ylim=c(0,1))

> plot(0:10,q2,xlab="Number of Heads",ylab="Probability of successful Heads",main="Successful Heads(B.Mohan_19BCN7015)",type="s",ylim=c(0,1))

> plot(0:20,q3,xlab="Number of Chips",ylab="Probability of Defective chips ",main="Defective Chips(B.Mohan_19BCN7015)",type="s",ylim=c(0,1))

> plot(0:100,q4,xlab="Number of Bits",ylab="Probability of Error Bits ",main="Error Bits(B.Mohan_19BCN7015)",type="s",ylim=c(0,1))
```

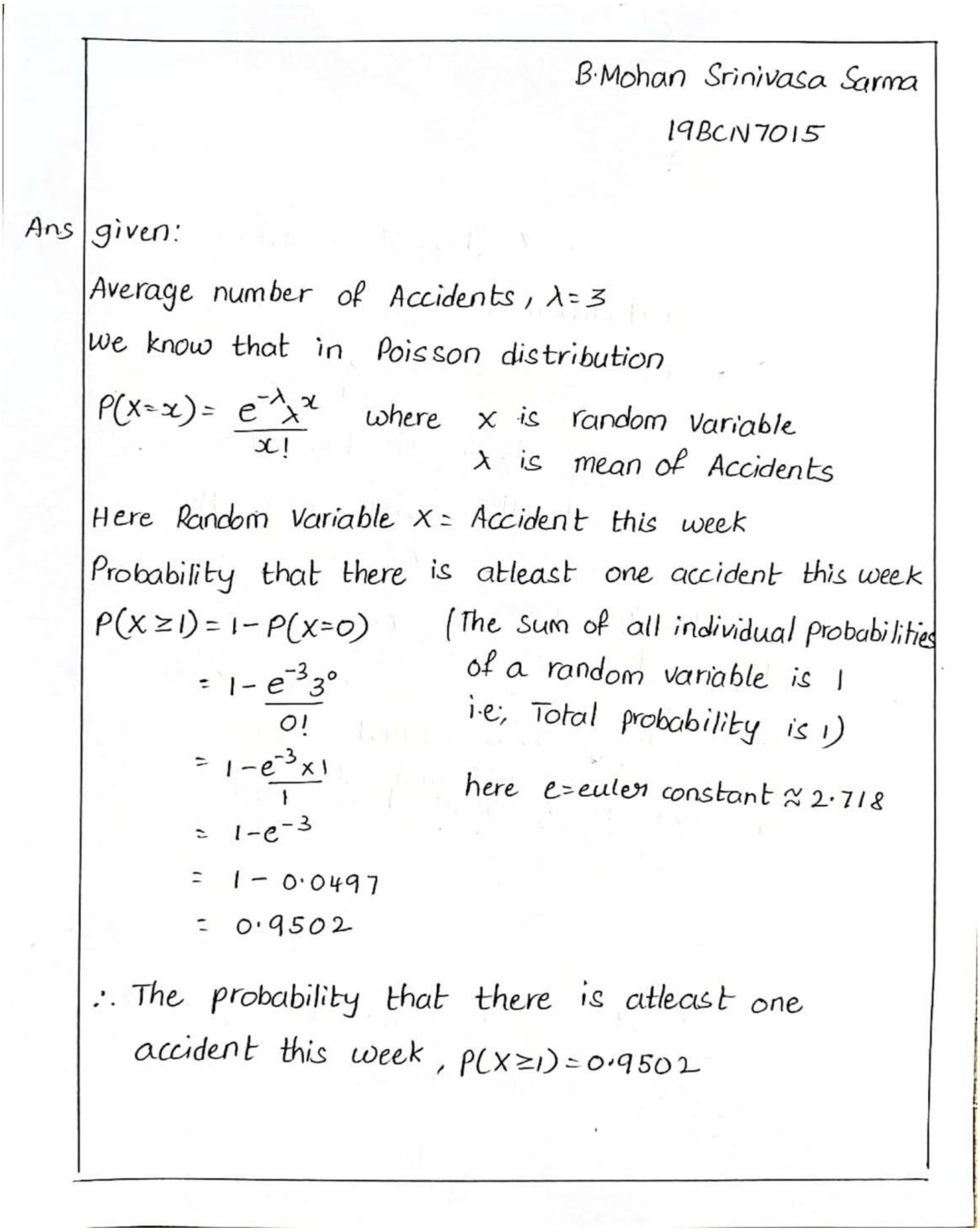
Output:



Day 5: Poisson Distribution

a. Suppose that the average number of accidents occurring weekly on a particular stretch of a highway equals 3. Calculate the probability that there is at least one accident this week.

Ans:



B·Mohan Srinivasa Sarma

19BCN7015

Ans given:

Average number of Accidents, $\lambda = 3$

We know that in Poisson distribution

$P(X=x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$ where x is random variable

λ is mean of Accidents

Here Random Variable $X =$ Accident this week

Probability that there is atleast one accident this week

$P(X \geq 1) = 1 - P(X=0)$ (The sum of all individual probabilities of a random variable is 1 i.e; Total probability is 1)

$= 1 - \frac{e^{-3}3^{0}}{0!}$

$= 1 - \frac{e^{-3} \times 1}{1}$

$= 1 - e^{-3}$

$= 1 - 0{\cdot}0497$

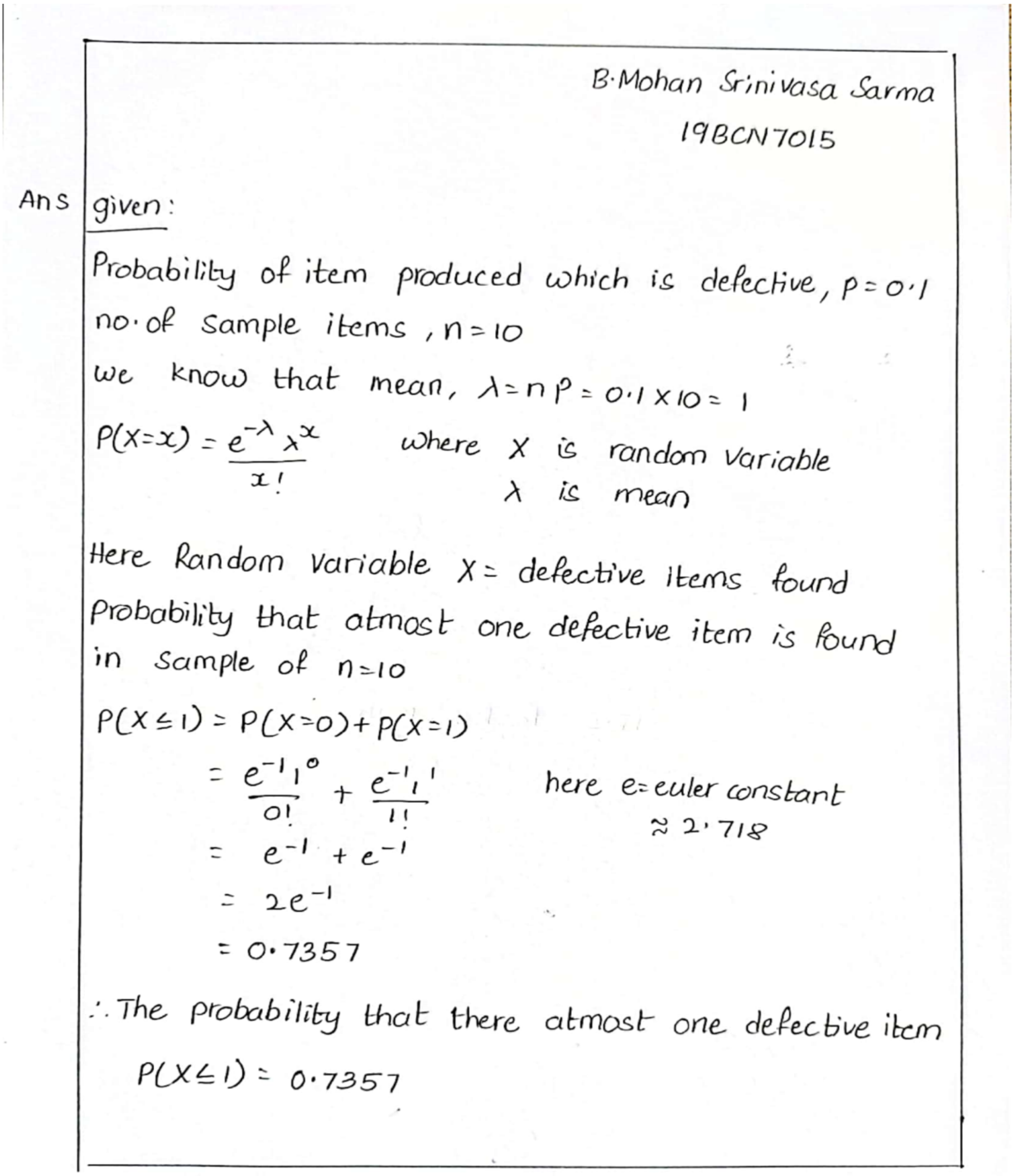
$= 0{\cdot}9502$

here $e =$ euler constant $\approx 2{\cdot}718$

$\therefore$ The probability that there is atleast one accident this week, $P(X \geq 1) = 0{\cdot}9502$

```
> ppois(0,lambda=3,lower=FALSE)
[1] 0.9502129
```

b. Suppose the probability that an item produced by certain machine will be defective is .1. Find the probability that a sample of 10 items will contain at most one defective item. Assume that the quality of successive items is independent.



B·Mohan Srinivasa Sarma
19BCN7015

Ans given:

Probability of item produced which is defective, $p = 0{\cdot}1$

no·of sample items, $n = 10$

we know that mean, $\lambda = np = 0{\cdot}1 \times 10 = 1$

$P(X=x) = \dfrac{e^{-\lambda}\lambda^{x}}{x!}$ where X is random variable
λ is mean

Here Random variable X = defective items found

Probability that atmost one defective item is found in sample of $n=10$

$P(X \leq 1) = P(X=0) + P(X=1)$

$= \dfrac{e^{-1}1^{0}}{0!} + \dfrac{e^{-1}1^{1}}{1!}$ here e = euler constant $\approx 2{\cdot}718$

$= e^{-1} + e^{-1}$

$= 2e^{-1}$

$= 0{\cdot}7357$

∴ The probability that there atmost one defective item

$P(X \leq 1) = 0{\cdot}7357$

```
> ppois(1,lambda=1,lower=TRUE)
[1] 0.7357589
```

c. If the average number of claims handled daily by an insurance company is 5, what proportion of days have less than 3 claims? What is the probability that there will be 4 claims in exactly 3 of next 5 days? Assume that the number of claims on different days is independent.

Ans:

```
> ppois(3,lambda=5,lower=TRUE)-dpois(3,lambda=5)

[1] 0.124652

#or

> ppois(2,lambda=5,lower=TRUE)

[1] 0.124652

> p<-dpois(4,lambda=5)

> p

[1] 0.1754674

> dbinom(3,size=5,prob=p)

[1] 0.03672864
```

Day 6: Normal Distribution-I

Suppose IQ’S are normally distributed with a mean of 100 and a standard deviation of 15.

1. What percentage of people have an IQ less than 125?

Code:

```
> q1<-pnorm(125,mean=100,sd=15,lower.tail=TRUE)
> q1
```

Output:

```
[1] 0.9522096
```

2. What Percentage of people have an IQ greater than 110?

Code:

```
> q2<-pnorm(110,mean=100,sd=15,lower.tail=FALSE)
> q2
```

Output:

```
[1] 0.2524925
```

3. What Percentage of people have an IQ between 110 and 125?

Code:

```
>q3<-pnorm(125,mean=100,sd=15,lower.tail=TRUE)-pnorm(110,mean=100,sd=15,lower.tail=TRUE)
> q3
```

Output:

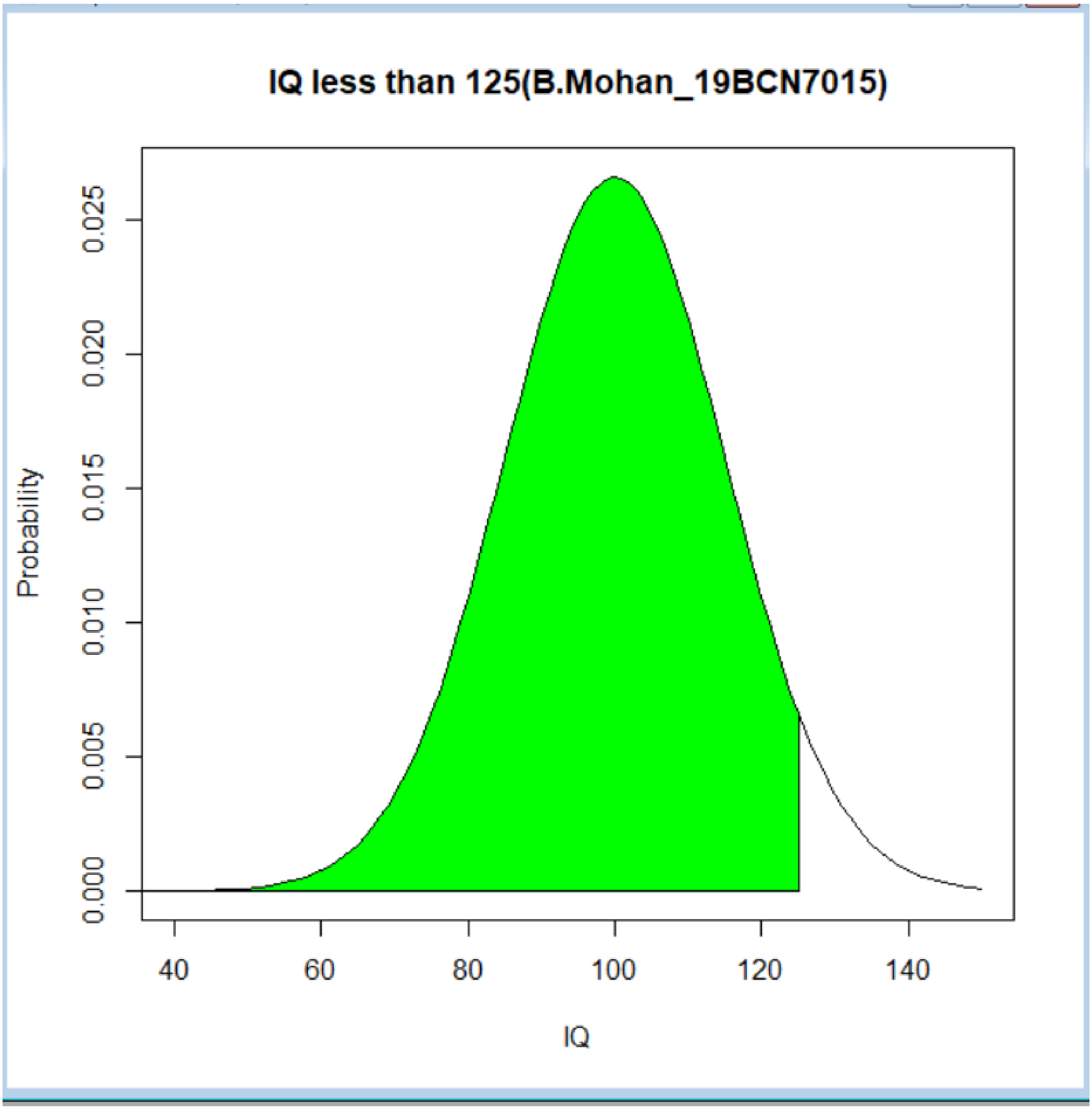
```
[1] 0.2047022
```

4. Draw the curves for above Cases

Code for Plotting for question 1:

```
> x<-seq(40,150)
> y<-dnorm(z,100,15)
> plot(x,y,type='n',xlab="IQ",ylab="Probability",main="IQ less than 125(B.Mohan_19BCN7015)")
> i<-x>=0&x<=125
> lines(x,y)
> polygon(c(0,x[i],125),c(0,y[i],0),col="green")
```

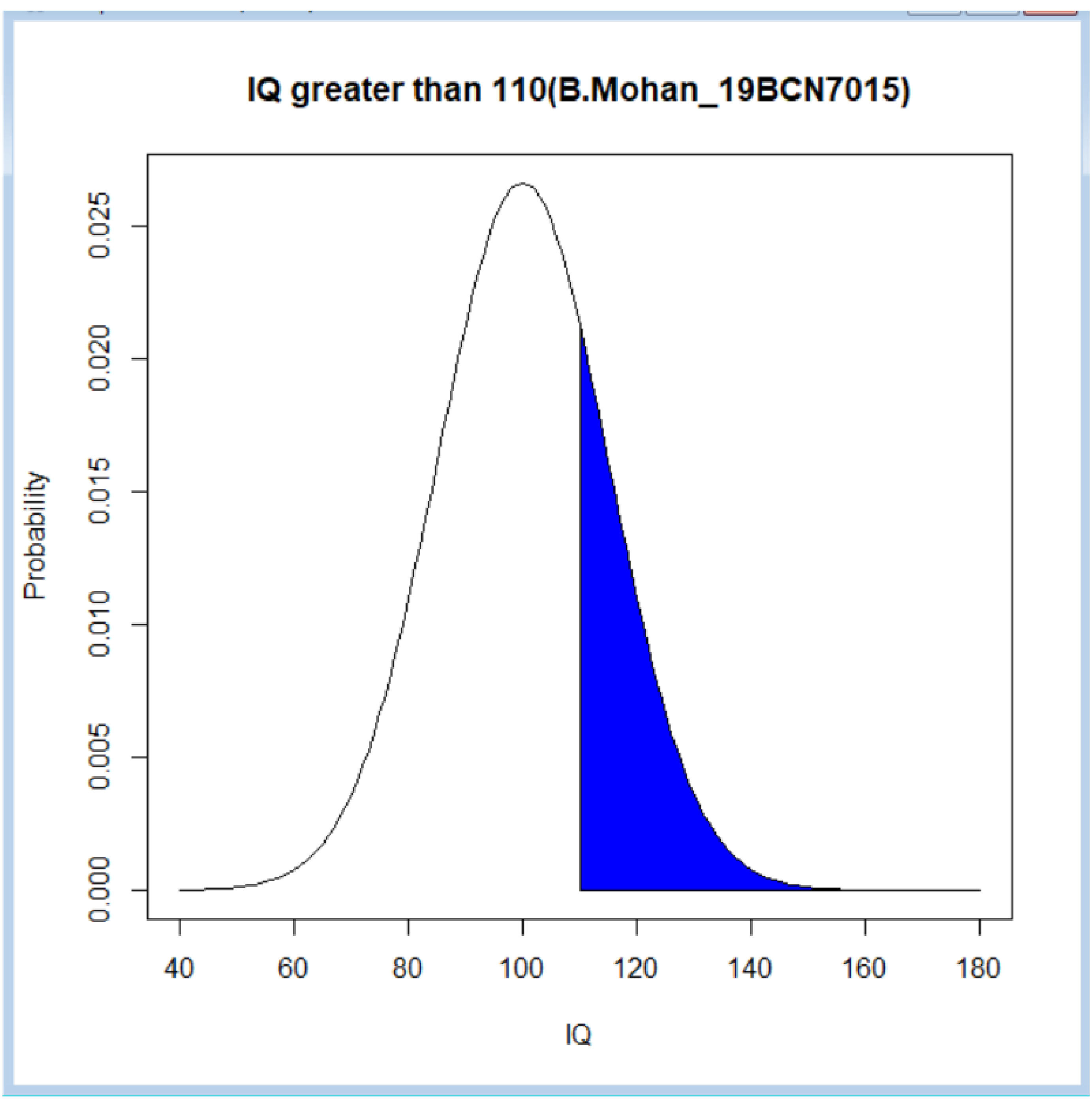
Plot:



Code for plotting for question 2:

```
> x<-seq(40,180)
> y<-dnorm(x,100,15)
> plot(x,y,type='n',xlab="IQ",ylab="Probability",main="IQ
greater than 110(B.Mohan_19BCN7015)")
> i<-x>=110&y<=180
> lines(x,y)
> polygon(c(110,x[i],180),c(0,y[i],0),col="blue")
```

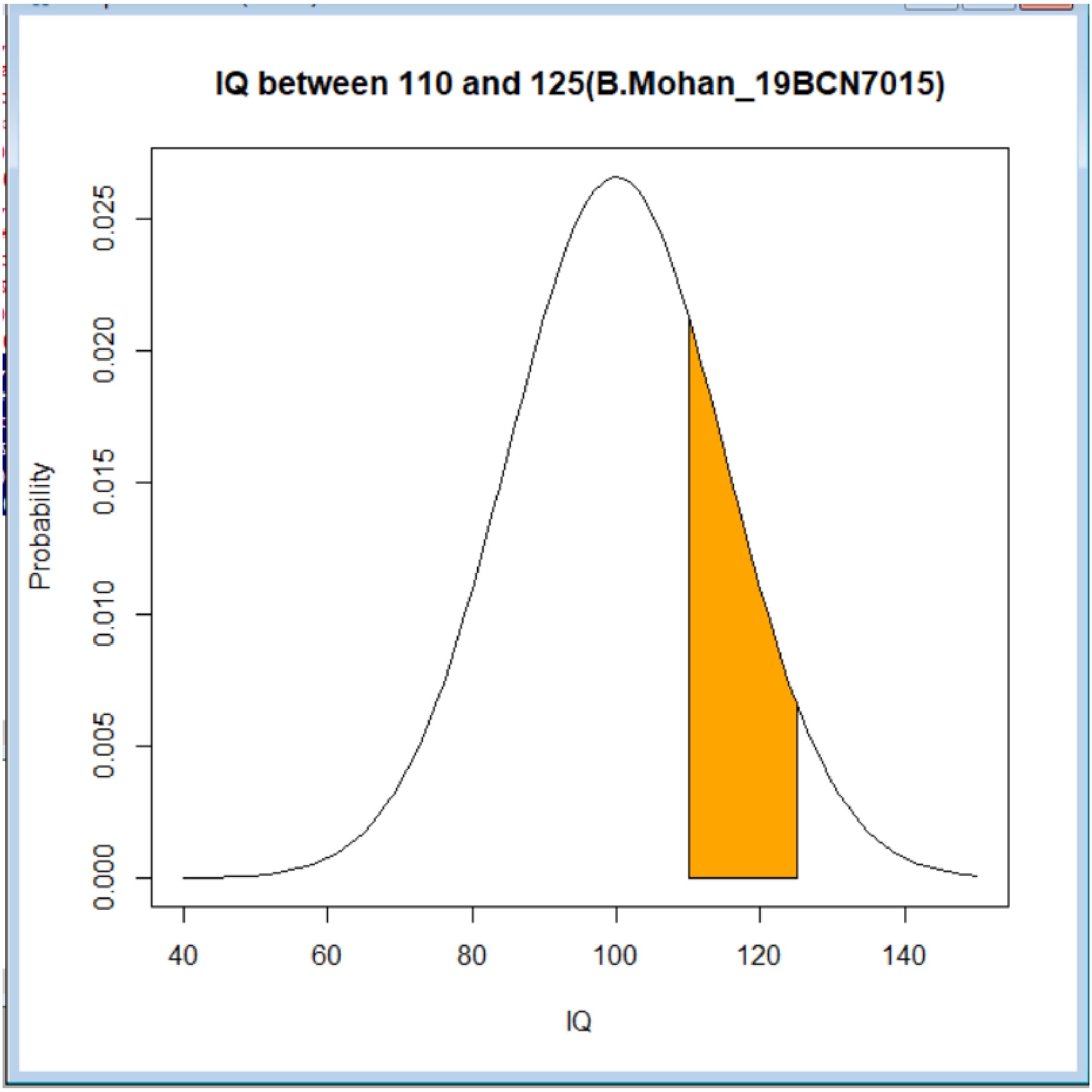
Plot:



Code for plotting for Question 3:

```
> x<-seq(40,150)
> y<-dnorm(x,100,15)
> plot(x,y,type='n',xlab="IQ",ylab="Probability",main="IQ
between 110 and 125(B.Mohan_19BCN7015)")
> i<-x>=110&x<=125
> lines(x,y)
> polygon(c(110,x[i],125),c(0,y[i],0),col="orange")
```

Plot:



5. What IQ separates the lower 25% from the others?

Code:

```
>qnorm(0.25,mean=100,sd=15,lower.tail=TRUE)
```

Output:

```
[1] 89.88265
```

6. What IQ separates the top 10% from the others? (Find P90)

Code:

```
>qnorm(0.90,mean=100,sd=15,lower.tail=TRUE)
```

Output:

```
[1] 119.2233
```

Day 7:Normal Distribution-II

A wall Street analyst estimates that the annual return from the stock of company A can be considered to be an observation from normal distribution with mean = 8.0% and standard deviation (sigma) =1.5%. The analyst's investment choices are based upon considerations that any return greater than 5% is "Satisfactory" and a return greater than 10% is "Excellent".

Find the probability that company A's stock will prove to be "Unsatisfactory".

Code:

```
> q1<-pnorm(5,8,1.5,lower.tail=TRUE)
> q1
```

Output:

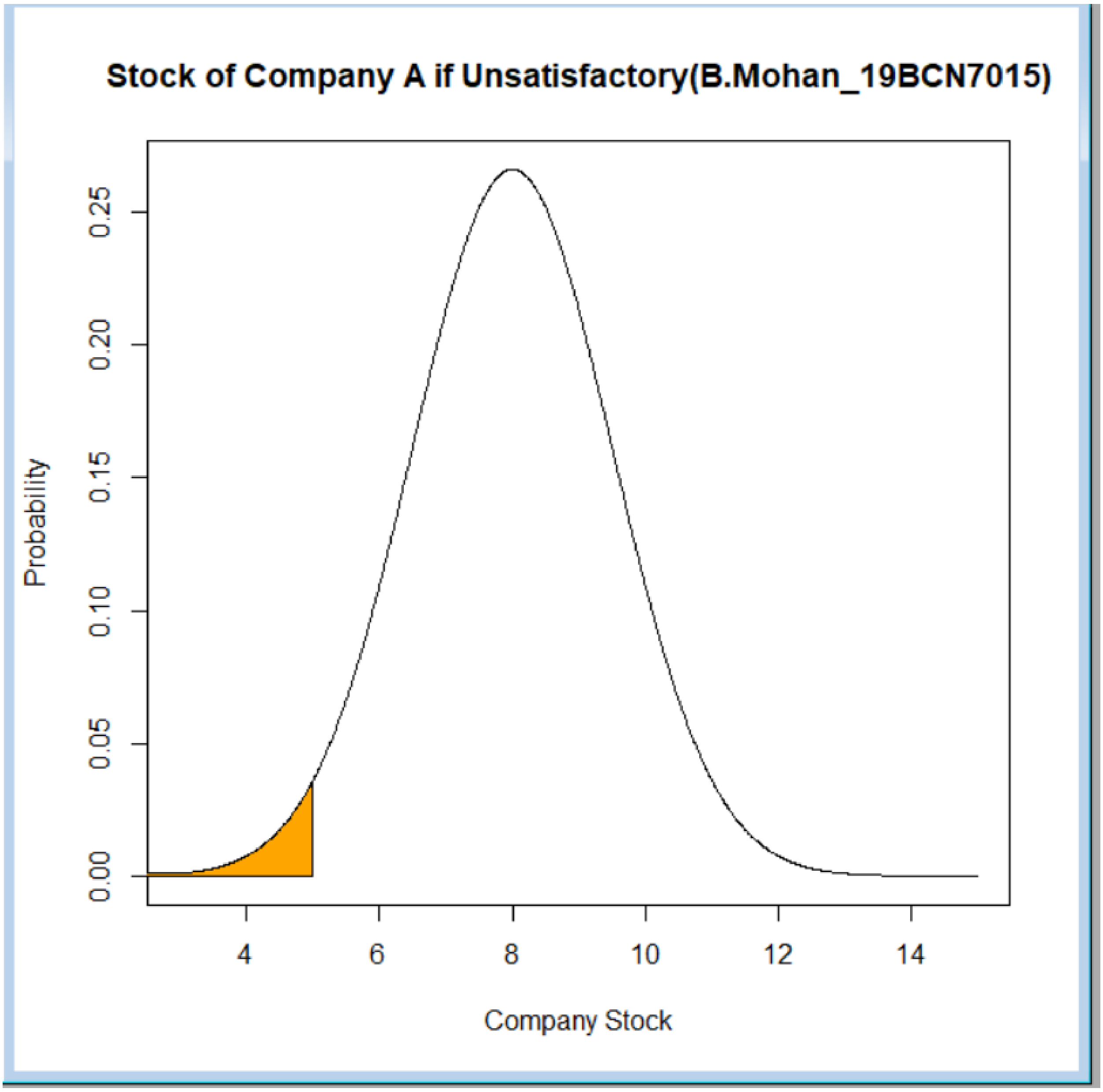
```
[1] 0.02275013
```

Plot:

```
> x<-seq(3,15,0.01)
> y<-dnorm(x,8,1.5)
```

```
>plot(x,y,type='n',xlab="Company Stock",ylab="Probability",main="Stock of Company A is Unsatisfactory(B.Mohan_19BCN7015)")
> i<-x>=0&x<=5
> lines(x,y)
> polygon(c(0,x[i],5),c(0,y[i],0),col="orange")
```

Output:



Find the probability that Company A’s stock will prove to be excellent

Code:

```
> q2<-pnorm(10,8,1.5,lower.tail=FALSE)

> q2
```

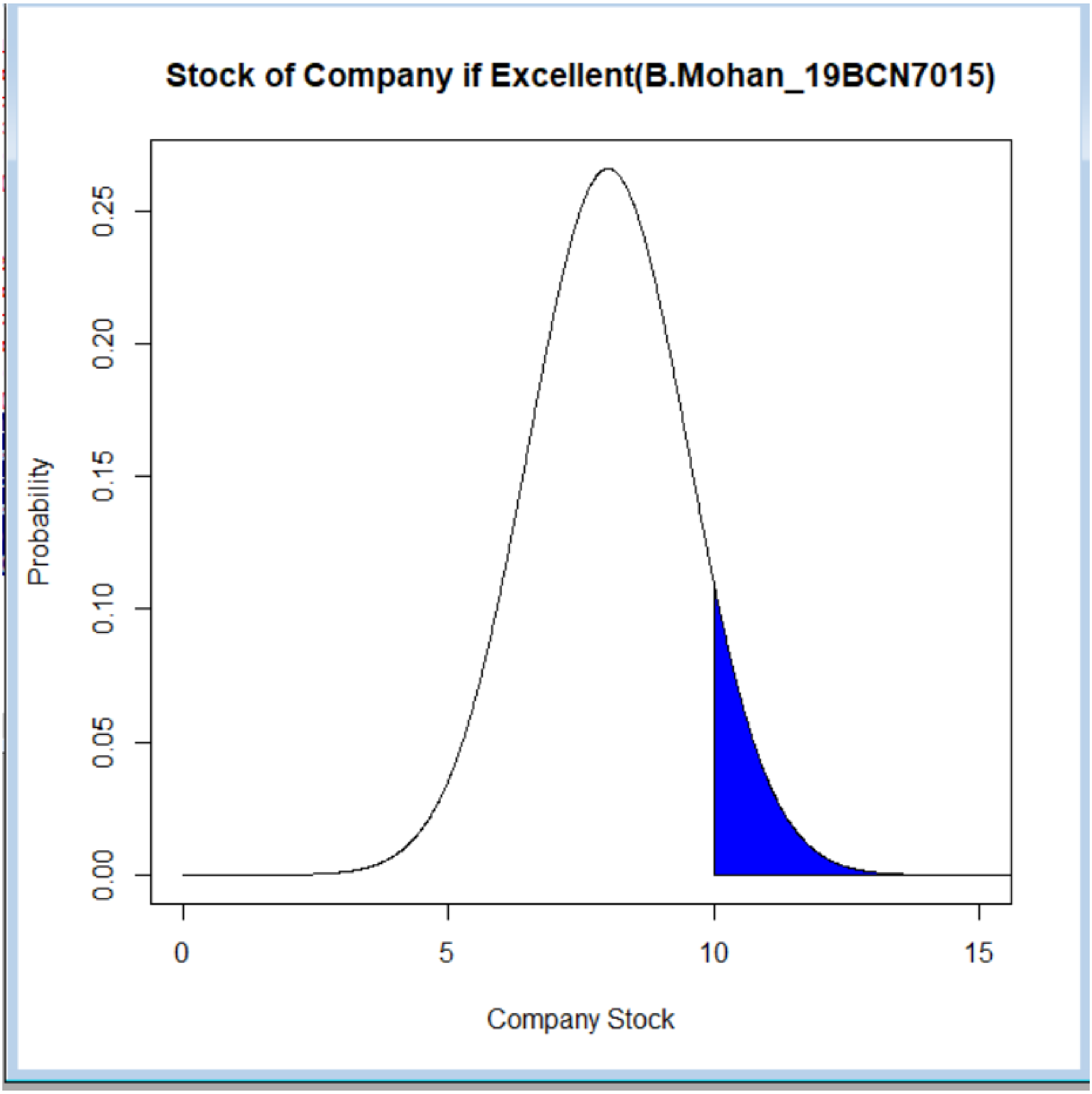
Output:

```
[1] 0.09121122
```

Code for plotting:

```
> x<-seq(0,15,0.01)

> y<-dnorm(x,8,1.5)

> plot(x,y,type='n',xlab="Company
Stock",ylab="Probability",main="Stock of Company if
Excellent(B.Mohan_19BCN7015)")

> i<-x>=10&x<=15

> lines(x,y)

> polygon(c(10,x[i],20),c(0,y[i],0),col="blue")
```

Output:



Stock of Company if Excellent(B.Mohan_19BCN7015)
Probability
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
15
Company Stock

Day 8: Test of Hypothesis: Z-Test

#Difference of means using Z test

We decide to run a test using an experimental evaporation pan and a standard evaporation pan over ten successive days. The two types are set up side by side so that atmospheric conditions should be the same. A coin is tossed to decide which operation pan is on the left hand side and which on the right hand side on any particular day. The measure daily evaporation are as follows:

Pair or Day no. 1 2 3 4 5 6 7 8 9 10

Evaporation mm:

Pan A 9.1 4.6 14.0 16.9 11.4 10.7 27.4 22.8 42.8 29.4

Pan B 6.7 3.1 13.8 16.6 12.3 6.5 24.2 20.1 41.9 27.7

Does the experimental pan A give significantly higher evaporation than the standard Pan B at 1% level of Significance.

Code:

```
> pana<-c(9.1,4.6,14.0,16.9,11.4,10.7,27.4,22.8,42.8,29.4)

> panb<-c(6.7,3.1,13.8,16.6,12.3,6.5,24.2,20.1,41.9,27.7)

> mua<-mean(pana)

> mub<-mean(panb)
> #h0:mua-mun=0
> #h0:mua-mub=0
> #h1:mua-mub>0
> n=10
> zstatistic=(mua-mub)/sqrt((sd(pana)^2+sd(panb)^2)/n)
> mua
[1] 18.91
> mub
[1] 17.29
> zstatistic
[1] 0.3099821
> #value of z statistic is 0.3099821
> p<-pnorm(zstatistic,lower.tail=FALSE)
> p
[1] 0.3782873
#or
> 1-pnorm(zstatistic,lower.tail=TRUE)
[1] 0.3782873
> #probability is 0.3782
> #As p value is greater than 1% level of significane h0 null
hypothesis is accepted
>#There is no evidence that Experimental Pan A  may give more
heat than Pan B
```

Day 9: Test of Hypothesis: t-Test

#Difference of means using t test

Arsenic concentration in public drinking water supplies is a potential health risk. An article in the Arizona Republic (Sunday, May 27, 2001 reported drinking water arsenic concentrations in parts per billion(ppb) for 10 metropolitan Phoenix communities and 10 communities in rural Arizona. The data follow

Metro Phoenix	Rural Arizona
Phoenix,3	Rimrock,48
Chandler,7	GoodYear,44
Gilbert,25	New River,40
Glendale,10	Apachie Junction,38
Mesa,15	Buckeye,33
Paradise Valley,6	Nogales,21
Scottsdale,25	sedona,12
Tempe,15	payson,1
Sun city,7	Casa Grande,18

We wish to determine it there is any difference in mean arsenic concentrations between metropolitan Phoenix communities and communities in rural Arizona

Code:

```
> x<-c(3,7,25,10,15,6,12,25,15,7)
> y<-c(48,44,40,38,33,21,20,12,1,18)
> #H0=mu1-mu2=0
>#H1=mu1-mu2!=0 (H1 is not equal to zero)
> #For 5% Significance Level( or 95% Confidence level)
> t.test(x,y,mu=0,alt="two.sided",conf=0.95,var.eq=F)
```

Output:

```

        Welch Two Sample t-test


data:  x and y
t = -2.7669, df = 13.196, p-value = 0.01583
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -26.694067  -3.305933
sample estimates:
mean of x mean of y
     12.5      27.5


>#t-statistic value=-2.7669
```

```
>#degree of freedom=13.196(approx. 13)

>#p-value=0.01583

> #According to the test at 1% significance level the null hypothesis is rejected as true difference means is not equal to zero.

> #Conclusion: There may be some be difference in mean of arsenic concentrations between metropolitan phoenix communities and communities in rural Arizona


> #For 1% Significance Level( or 99% Confidence level)

> t.test(x,y,mu=0,alt="two.sided",conf=0.99,var.eq=F)
```

Output:

```
	Welch Two Sample t-test


data:  x and y

t = -2.7669, df = 13.196, p-value = 0.01583

alternative hypothesis: true difference in means is not equal to 0

99 percent confidence interval:

 -31.289827   1.289827

sample estimates:

mean of x mean of y

     12.5      27.5
```

```
>#t-statistic value=-2.7669
>#degree of freedom=13.196(approx. 13)
>#p-value=0.01583

> #According to the test at 5% significance level the null
hypothesis is rejected as true difference means is not equal to
zero.

> #Conclusion: There may be some be difference in mean of
arsenic concentrations between metropolitan phoenix
communities and communities in rural Arizona

>#Hence at both 1% and 5% significant levels null hypothesis is
rejected. Alternative hypothesis is accepted. In both cases of
significance, the confidence intervals are changing. There may be
some be difference in mean of arsenic concentrations between
metropolitan phoenix communities and communities in rural
Arizona
```

Day 10: Regression-I

The cetane number is a critical property in specifying the ignition quality of a fuel used in diesel engine. Determination of this number for a biodiesel fuel is expensive and time consuming. The article “Relating the cetane Number of biodiesel Fuels to their Fatty Acid Composition: A Critical Study” (J. of Automobile Engr.,2009: 565-583) included the following data on x =iodine value(g) and y=cetane number of 14 biofuels. The iodine value is the amount of iodine necessary to saturate a sample of 100g of oil. The article’s authors fit the simple linear regression model to this data, So let’s follow their lead.

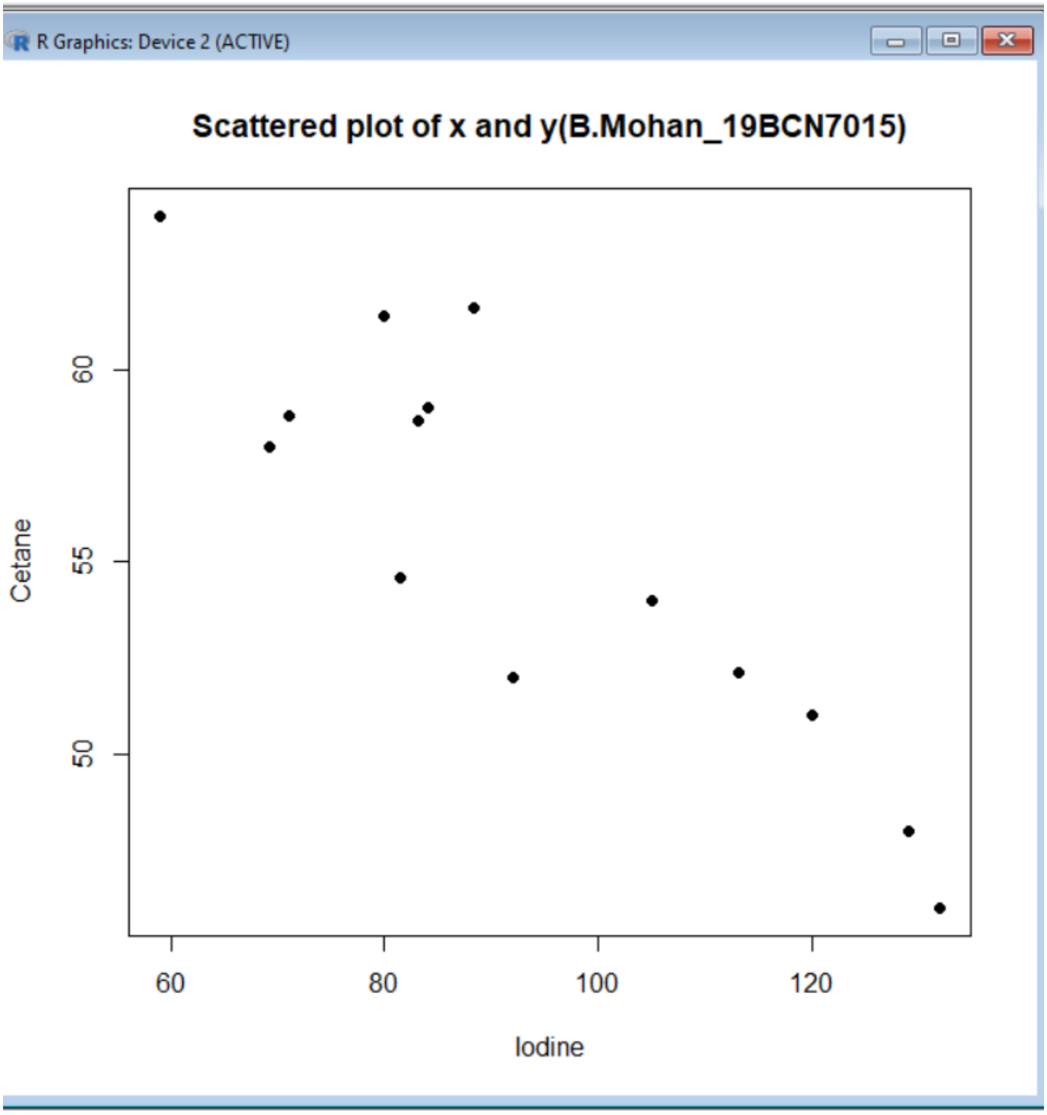
x	*132*	*129*	*120*	*113.2*	*105*	*92*	*84*	*83.2*	*88.4*	*59*	*80*	*81.5*	*71*	*69.2*
y	46	48	51	52.1	54	52	59	58.7	61.6	64	61.4	54.6	58.8	58

1. Do the Scatter plot for the given data.

Code:

```
> x<-
c(132,129,120,113.2,105,92,84,83.2,88.4,59,80,81.5,71,69.2)

> y<-c(46,48,51,52.1,54,52,59,58.7,61.6,64,61.4,54.6,58.8,58)

> plot(x,y,pch=16,xlab="Iodine",ylab="Cetane",main="Scattered
plot of x and y(B.Mohan_19BCN7015)")
```

Output:



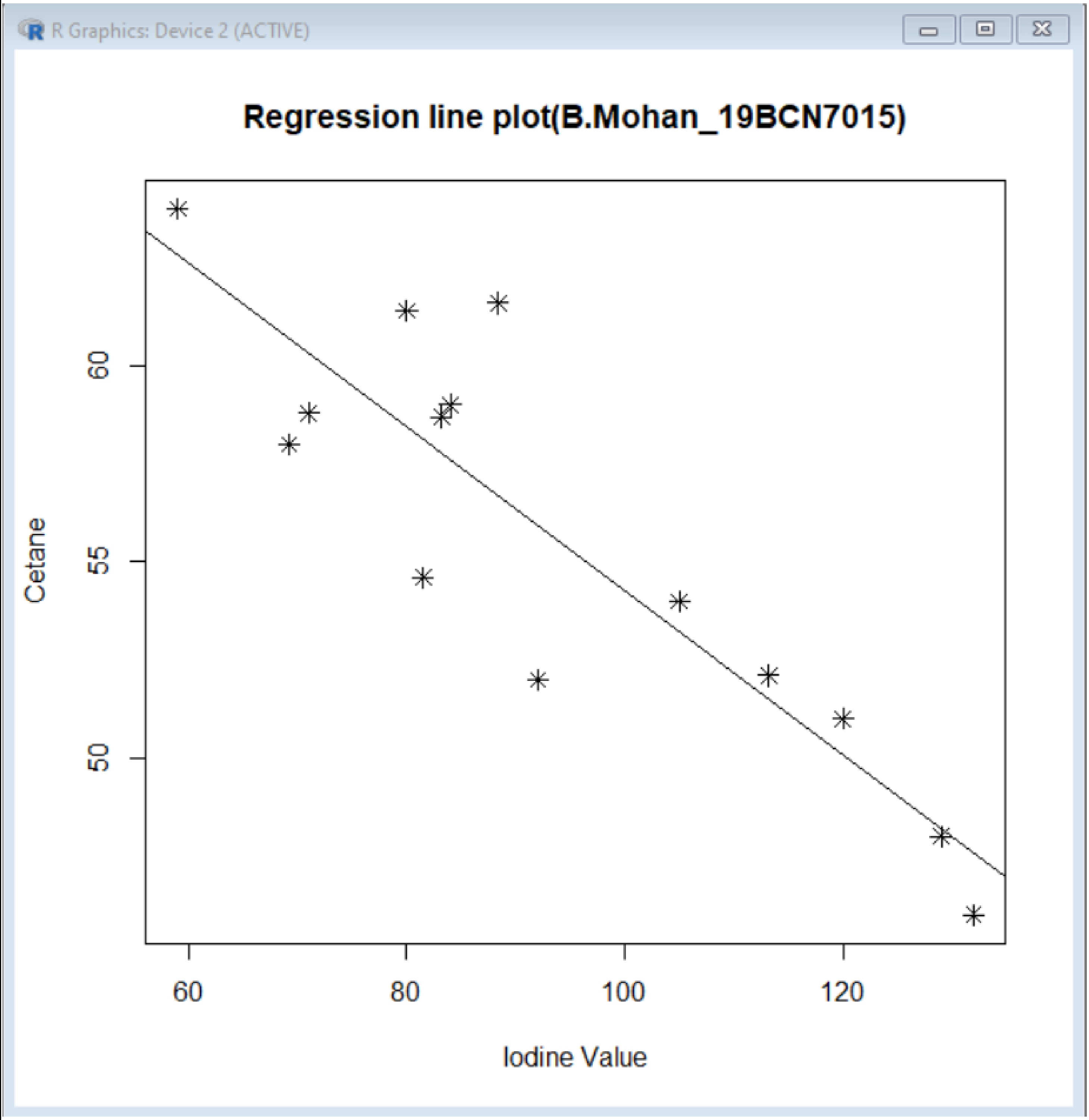
2. Based on the Given data find the true regression line equation and plot accordingly.

Code:

```
> x<-c(132,129,120,113.2,105,92,84,83.2,88.4,59,80,81.5,71,69.2)

> y<-c(46,48,51,52.1,54,52,59,58.7,61.6,64,61.4,54.6,58.8,58)

> meanx<-mean(x)

> meanx

[1] 93.39286

> #Mean of x is 93.39286

> meany<-mean(y)

> meany

[1] 55.65714

> #Mean of Y is 55.65714

> x1<-x-meanx

> y1<-y-meany

> sxy<-sum(x1*y1)

> sxy

[1] -1424.414

> #Value of sxy is -1424.414

> sxx<-sum(x1^2)

> sxx

[1] 6802.769
```

```
> #Value of sxx is 6802.769

> beta1<-sxy/sxx

> beta1

[1] -0.2093874

> #Value of beta1 is -0.2093874

> beta0<-muy-beta1*mux

> beta0

[1] 75.21243

> #Value of beta0 is 75.21243

> plot(x,y,pch=8,xlab="Iodine
Value",ylab="Cetane",main="Regression line
plot(B.Mohan_19BCN7015)",abline(lm(y~x)),cex=1.2)
```

Output:


R Graphics: Device 2 (ACTIVE)
Regression line plot(B.Mohan_19BCN7015)
Cetane
50
55
60
60
80
100
120
Iodine Value

3. Prediction anaylsis of requirement of cetane for 100g of Iodine

Code:

```
> res<-beta0+beta1*100

> res
```

Output:

```
[1] 54.27369

>#The value of y when x=100 is 54.27
```

4.Find the Correlation Coefficient

Code:

```
> syy<-sum(y1^2)

> res<-sxy/(sqrt(sxx)*sqrt(syy))

> res
```

Output:

```
[1] -0.889247
```

Day 11: Regression-II

.Question by Industry Expert

Rishav, an intelligent guy from Delhi, grew up seeing the festival of Holi in its various colours. His father was a daily labour and his earning was not that much. However, Rishavhad business in his blood. He thought to start his venture with opening a little shop at the time of Holi with all the pichkaris, gulals and colours.

He wanted to earn some specific amount of money which would help him pay his education bill for that year. He also observed that there is a kind of linear relationship between the size of the shop and sales (in rupees) from that shop.

Could you please help Rishav know that how much money he will probably earn, if he opens a shop of 350 sq ft.?

For your reference, Rishav has given the list of shops (size and sales figure) he collected from his locality.

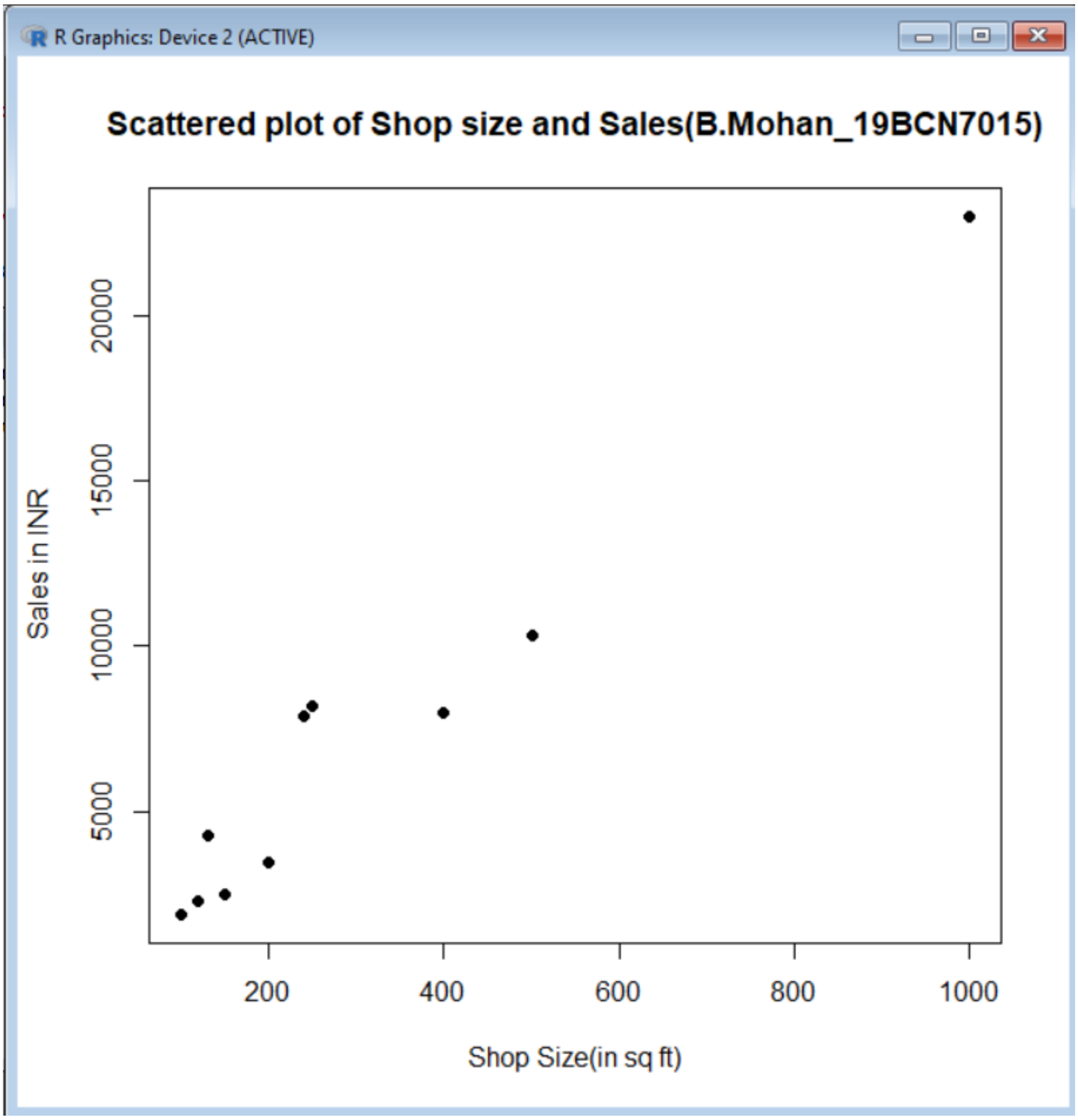
Shop size (in sq ft)	Sales (In INR)
100	1900
130	4300
200	3450
240	7890
250	8210
500	10324
1000	22980
400	8000
150	2505
120	2300

1. Do the Scatter plot for the given data.

Code:

```
> shopsize<-c(100,130,200,240,250,500,1000,400,150,120)

> sales<-c(1900,4300,3450,7890,8210,10324,22980,8000,2505,2300)

> plot(shopsize,sales,pch=16,xlab="Shop Size(in sq ft)",ylab="Sales in INR",main="Scattered plot of Shop size and Sales(B.Mohan_19BCN7015)")
```

Output:


R Graphics: Device 2 (ACTIVE)
Scattered plot of Shop size and Sales(B.Mohan_19BCN7015)
Sales in INR
5000
10000
15000
20000
200
400
600
800
1000
Shop Size(in sq ft)

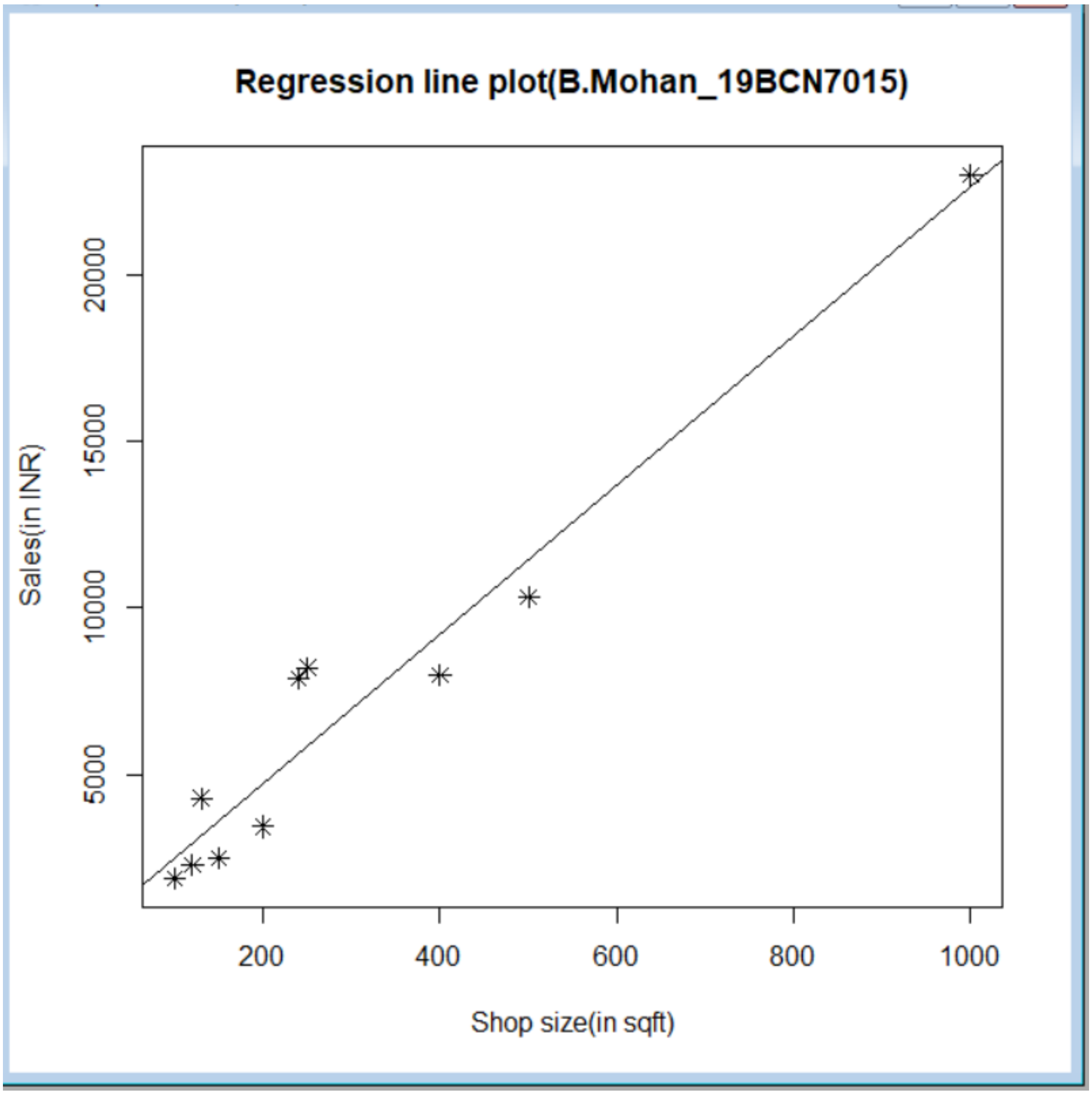
2. Based on the Given data find the true regression line equation and plot accordingly.

Code:

```
> meanx<-mean(shopsize)

> meany<-mean(sales)

> meanx

[1] 309

> #Mean of Shop Size is 309

> meany

[1] 7185.9

> #Mean of Sales is 7185.9

> x1<-shopsize-meanx

> y1<-sales-meany

> sxy<-sum(x1*y1)

> sxy

[1] 15174419

> #Value of sxy is 15174419

> sxx<-sum(x1^2)

> sxx

[1] 679090

> #Value of sxx is  679090

> beta1<-sxy/sxx

> beta1

[1] 22.34523
```

```
> #Value of Beta1 is 22.34523

> beta0<-meany-beta1*meanx

> beta0

[1] 281.2254

> #Value of beta0 is 281.2254

> #Equation of regresssion line is y=281.2254+22.34523*x

>plot(shopsize,sales,pch=8,xlab="Shop size(in sqft)",ylab="Sales(in
INR)",main="Regression line
plot(B.Mohan_19BCN7015)",abline(lm(sales~shopsize)),cex=1.2)
```

Plot of Regression Line:



3. Prediction anaylsis of how much money will Rishav earn from 350 sq ft shop?

Code:

```
> #When rishey opens shop at 350 sqft
> res<-beta0+beta1*350
> res
```

Output:

```
[1] 8102.054
> #Therefore sales of rishey in his 350sqft shop is 8102.054
```

4. Find the Correlation Coefficient

Code:

```
> syy<-sum(y1^2)
> res<-sxy/(sqrt(sxx)*sqrt(syy))
> res
```

Output:

```
[1] 0.9738672
>#Therefore correlation Coefficient is 0.9738
```

Day 12: Case Study with a Real life problem

Question:

Price is US $ of the First Six Products out of 29 totals

Product	Wegmans Price	Publix Price
skim milk, gallon	1.89	3.55
Activa yogurt, plain, large	2.69	2.39
eggs, large, Grade A, dozen	1.29	2.69
Jif creamy peanut butter, 40 oz.	4.39	5.87
Diet Coke, 12 pack	3.33	4.99
Bread, 10 pack	2.66	2.29

Note: these prices are not fictional, they are actual prices taken on March 6th, 2011 from the Wegmans in Webster, NY, and subsequently March 8th, 2011 from the Publix in Windermere, FL.

Usually all people like to buy the products if they are cheap to buy. Above the data collected from internet of two Super markets which provides staple ingredients of daily life. Above the prices of the staple ingredients in both markets.

Find if there is any difference in means of prices in markets so that customers may choose from which market.

Code:

```
> #Here for this data we can use t test as the sample size is small.
> #t-test for above data
> #H0<-mu1-mu2=0
> #H1<-mu1-mu2!=0(mu1-mu2 is not equal to zero)
> #Testing at 1% Significance level(99% Confidence level)
> t.test(wp,pp,mu=0,alt="two.sided",var.eq=F,conf=0.99)
```

Output:

```
        Welch Two Sample t-test


data:  wp and pp
t = 0.79698, df = 5.0919, p-value = 0.461
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 -20.37751  30.53417
sample estimates:
mean of x mean of y 
 8.708333  3.630000
```

```
> #At 1% significance level we get
>#t-statistic=0.7968
>#Degrees of freedom=5.0919
>#p-Value=0.461
```

Hypothesis: We can see that at 1% significance level the null hypothesis is rejected. The alternate hypothesis that difference in means is not equal to zero.

```
> #Testing at 5% Significance level(95% Confidence level)
> t.test(wp,pp,mu=0,alt="two.sided",var.eq=F,conf=0.95)
```

Output:

```
        Welch Two Sample t-test


data:  wp and pp
t = 0.79698, df = 5.0919, p-value = 0.461
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -11.21290  21.36956
sample estimates:
mean of x mean of y
 8.708333  3.630000
```

```
> #At 5% significance level we get
>#t-statistic=0.7968
>#Degrees of freedom=5.0919
>#p-Value=0.461
```

Hypothesis: We can see that at 5% significance level the null hypothesis is rejected. The alternate hypothesis that difference in means is not equal to zero.

Conclusion:

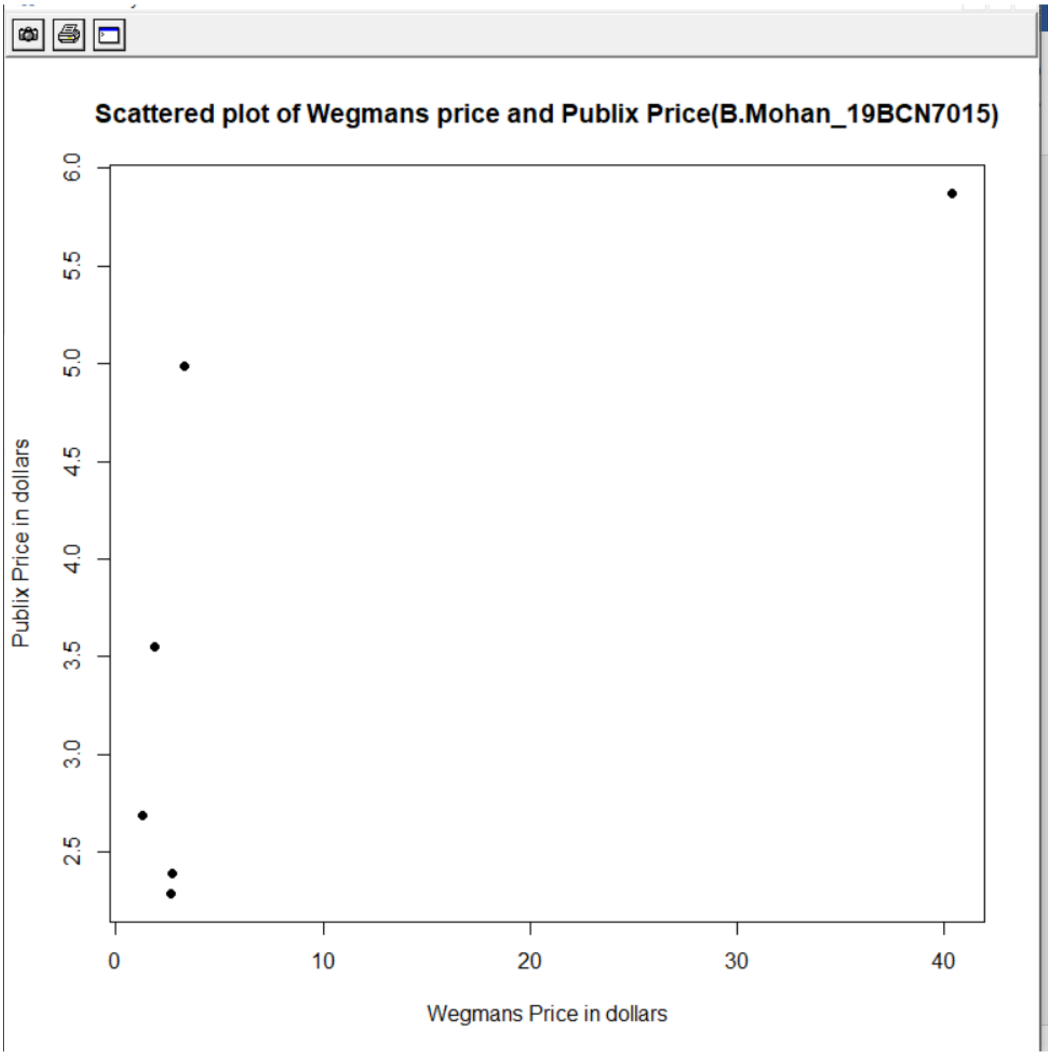
We can conclude that at both levels of significance tested above the null hypothesis (H0) is rejected. We can see that in each case the confidence interval is changing So there may be some difference in the means of prices in both markets. We may tell that the customer might purchase good at market which has less average of price according to given sample of data.

Also find the estimated cost of certain item in Publix Price if the item costs 3$ in Wegmans Market.

1. Do the Scatter plot for the given data.

```
>#To find the above query we may use Regression analysis
> wp<-c(1.89,2.69,1.29,40.39,3.33,2.66)
> pp<-c(3.55,2.39,2.69,5.87,4.99,2.29)
> plot(wp,pp,pch=16,xlab="Wegmans Price in dollars",ylab="Publix Price in dollars",main="Scattered plot of Wegmans price and Publix Price(B.Mohan_19BCN7015)")
```

Output:

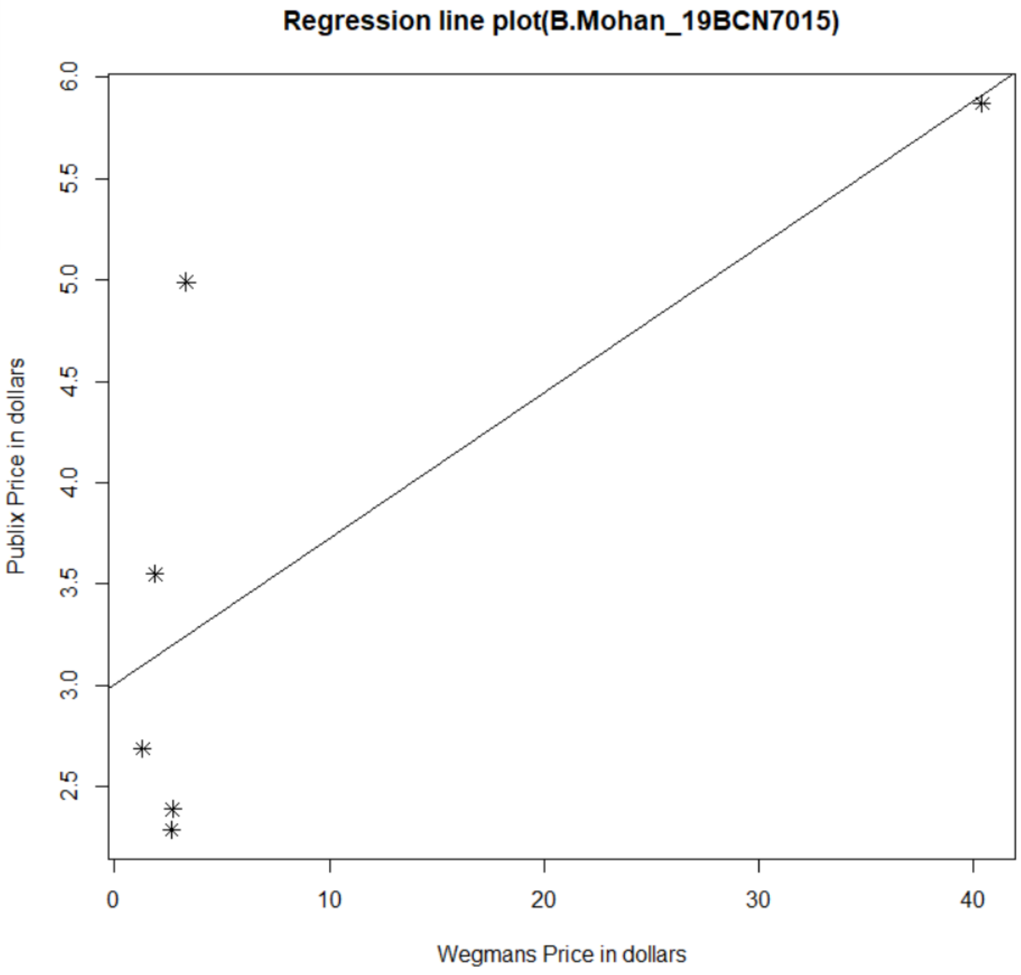


2. Based on the Given data find the true regression line equation and plot accordingly.

Code:

```
> wp<-c(1.89,2.69,1.29,40.39,3.33,2.66)

> pp<-c(3.55,2.39,2.69,5.87,4.99,2.29)

> meanx<-mean(wp)

> meany<-mean(pp)

> meanx

[1] 8.708333

>#Mean of Wegmans price is 8.708333

> meany

[1] 3.63

>#Mean of Publix price is 8.708333

> x1<-wp-meanx

> y1<-pp-meany

> sxy<-sum(x1*y1)

> sxy

[1] 86.7386

>#Value of sxy is 86.7386

> sxx<-sum(x1^2)

> sxx

[1] 1206.978

>#Value of sxx is 1206.978

> beta1<-sxy/sxx
> beta1
[1] 0.07186425
> beta0<-meany-beta1*meanx
> beta0
[1] 3.004182
>#Value of beta0 is 3.004182
> #Equation of regresssion line is y=3.004182+0.07186425*x
> plot(wp,pp,pch=8,xlab="Wegmans Price in dollars",ylab="Publix Price in dollars",main="Regression line plot(B.Mohan_19BCN7015)",abline(lm(pp~wp)),cex=1.2)
```

Plot of regression line:



3. Prediction anaylsis of how much it costs of certain item in Publix Price if the item costs 3$ in Wegmans Market.

Code:

```
>#Price of item in Publix market
> res<-beta0+beta1*3
> res
```

Output:

```
[1] 3.219775
>#Therefore Price of same item at Publix market may be 3.21 dollars .Hence it is preferable to buy in wegmans market
```

4. Find the Correlation Coefficient

Code:

```
> syy<-sum(y1^2)
> res<-sxy/(sqrt(sxx)*sqrt(syy))
> res
```

Output:

```
[1] 0.7497026
>#Hence Correlation Coefficient is 0.749706
```