Hands on exercise
R Programming

Submitted by – Gudi Varaprasad

Registration Number – 19BCE7048

Submitted to - Dr. Tanuj Kumar sir

Submitted on – 08th December 2020

Semester - Fall Semester (2020-2021)

Course Name – Applied Statistics

Course code - MAT1011

Course type – Embedded Lab

Slot – L1

Hands on exercise
R Programming

VIT AP

Table of Contents

■ *Indicates question*

■ *Indicates R code, command*

■ *Indicates result, answer*

Day 1 : Basics about R

Date : 9-11-2020

1. Simple Operations

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

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

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

```
> length(x)
[1] 7
```

c) Find the last element of x

```
> print(x[length(x)])
[1] 6
```

d) Find the minimum element of x

```
> print(min(x))
[1] 1
```

e) Find the maximum element of x.

```
> print(max(x))
[1] 9
```

2. Enter the data {1, 2, ... ,19,20} in a variable x.
a) Find the 3rd element in the data list.

```
> x <- (1:20)
> print(x)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

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

```
> print(x[3:5])
[1] 3 4 5
```

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

```
> print(x[c(2,5,6,12)])
[1] 2 5 6 12
```


Submitted by Gudi Varaprasad – 19BCE7048

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

```
> print(rev(x))
 [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.

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

```
> print(rep(c(4, 6, 3),10))
 [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.

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

```
> l1 = list(2, 1, 3, 4)
> l2 = list(5, 7, 12, 6, -8)
> l3 = append(l1, l2)
> # print(l3)
> length(l3)==11
[1] FALSE
```

4. (a) Print all numbers starting with 3 and ending with 7 with an increment of 0:0.5. Store these numbers in x.

```
> x = seq(from=3, to=7, by=0.5)
> print(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)

```
> l = seq(from=2, to=14, by=2)
> print(l)
[1]  2  4  6  8 10 12 14
```

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

```
> print(2*x)
[1]  6  7  8  9 10 11 12 13 14
```

Day 2 : Data Analysis using R

Date : 10-11-2020

1. Few Simple Statistical measures :

(a) Enter data as 1,2,…,10.

```
> x = (1:10)
> print(x)
 [1]  1  2  3  4  5  6  7  8  9 10
```

(b) Find sum of numbers

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

(c) Find Mean and Median

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

(d) Find sum of squares of these values

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

(e) Find the value of Mean deviation about Mean (M.D.x)

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

(f) Check whether M.D.x is < = Standard deviation

```
> sd.result = sqrt(var(x))
> sd.result
[1] 3.02765
```

2. Create a file as follows and store as a :-

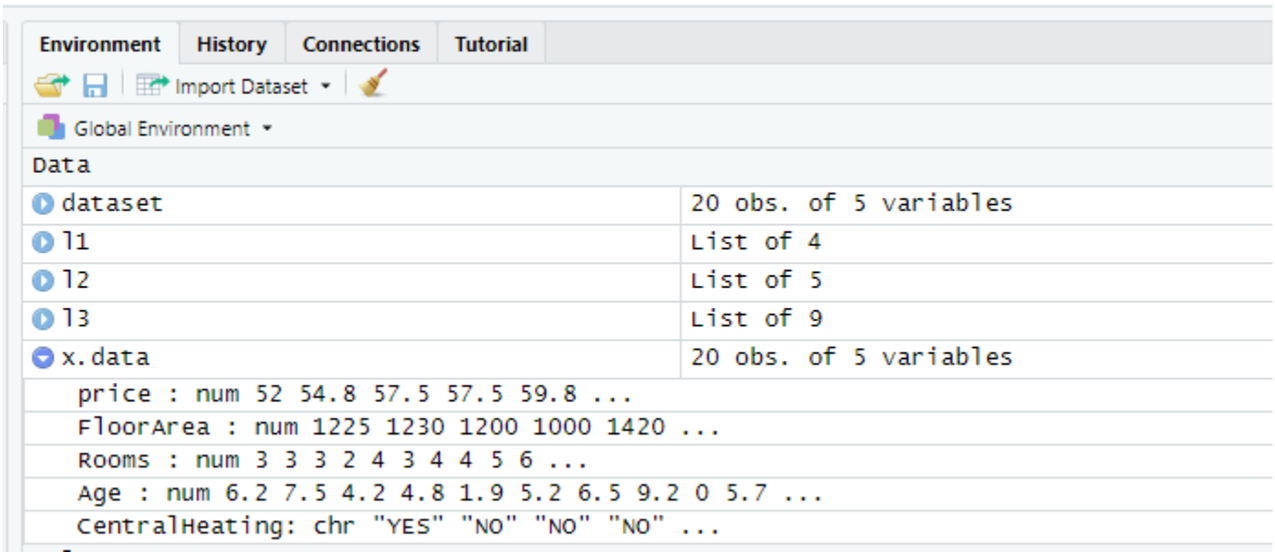
	price	FloorArea	Rooms	Age	CentralHeating
1	52.00	1225	3	6.2	YES
2	54.75	1230	3	7.5	NO
3	57.50	1200	3	4.2	NO
4	57.50	1000	2	4.8	NO
5	59.75	1420	4	1.9	YES
6	62.50	1450	3	5.2	YES
7	64.75	1380	4	6.5	NO
8	67.25	1510	4	9.2	NO
9	67.50	1400	5	0.0	NO
10	69.75	1550	6	5.7	NO
11	70.00	1720	6	7.3	YES
12	75.50	1700	5	4.5	NO
13	77.50	1660	6	6.8	YES
14	78.00	1800	7	0.7	YES
15	81.25	1830	6	5.6	YES
16	82.50	1790	6	2.3	NO
17	86.25	2010	6	6.7	YES
18	87.50	2000	6	3.4	NO
19	88.00	2100	8	5.6	YES
20	92.00	2240	7	3.4	YES

Submitted by Gudi Varaprasad – 19BCE7048

> # getwd()

> setwd("E:\\VITAP\\19BCE7048\\Semester_3\\Applied Statistics\\R programming Lab\\Lab Materials\\Lab 2 - Data Analysis Using R")

> x.data=read.csv('data.csv', sep=',', header = TRUE)



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

> length(x.data)

[1] 5

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

> NROW(x.data)

[1] 20

> NCOL(x.data)

[1] 5

c) What are the variables in the data file?

> names(x.data)

[1] "price" "FloorArea" "Rooms" "Age" "CentralHeating"

d) 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?

> head(x.data)

	price	FloorArea	Rooms	Age	CentralHeating
1	52.00	1225	3	6.2	YES
2	54.75	1230	3	7.5	NO
3	57.50	1200	3	4.2	NO
4	57.50	1000	2	4.8	NO
5	59.75	1420	4	1.9	YES
6	62.50	1450	3	5.2	YES

> tail(x.data)

	price	FloorArea	Rooms	Age	CentralHeating
15	81.25	1830	6	5.6	YES
16	82.50	1790	6	2.3	NO
17	86.25	2010	6	6.7	YES
18	87.50	2000	6	3.4	NO
19	88.00	2100	8	5.6	YES
20	92.00	2240	7	3.4	YES

e) 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?

> x.data[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

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

> x.data[c(1,3,6,10),c(2,4,5)]

	FloorArea	Age	CentralHeating
1	1225	6.2	YES
3	1200	4.2	NO
6	1450	5.2	YES
10	1550	5.7	NO

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

> # Element at second row, third column

> x.data[2,3]

[1] 3

3. Calculate simple statistical measures using the values in the data file.
a) Find means, medians, standard deviations of Price, Floor Area, Rooms, and Age.

> mean(x.data$price)

[1] 71.5875

> median(x.data$price)

[1] 69.875

> sd(x.data$price)

[1] 12.21094

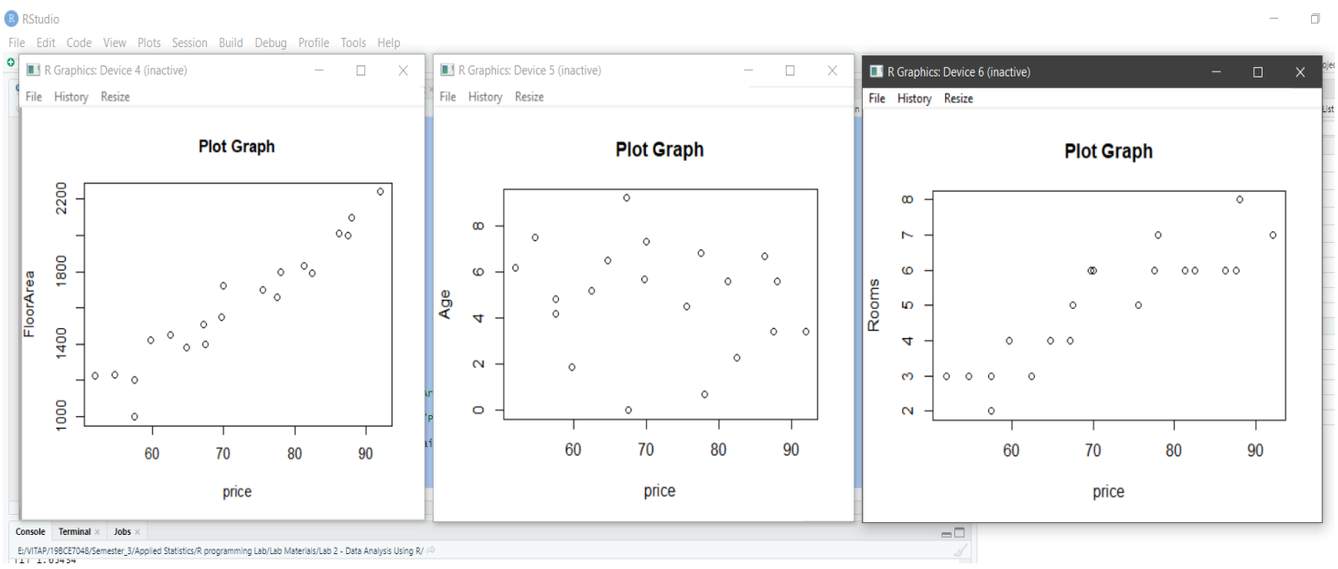
```
> mean(x.data$FloorArea)
[1] 1610.75
> median(x.data$FloorArea)
[1] 1605

> sd(x.data$FloorArea)
[1] 331.9649
>
> mean(x.data$Rooms)
[1] 5
> median(x.data$Rooms)
[1] 5.5
> sd(x.data$Rooms)
[1] 1.65434
>
> mean(x.data$Age)
[1] 4.875
> median(x.data$Age)
[1] 5.4
> sd(x.data$Age)
[1] 2.366182
```

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

```
> print(nrow(subset(x.data,x.data$CentralHeating=='YES')))
[1] 10
> print(nrow(subset(x.data,x.data$CentralHeating=='NO')))
[1] 10
```

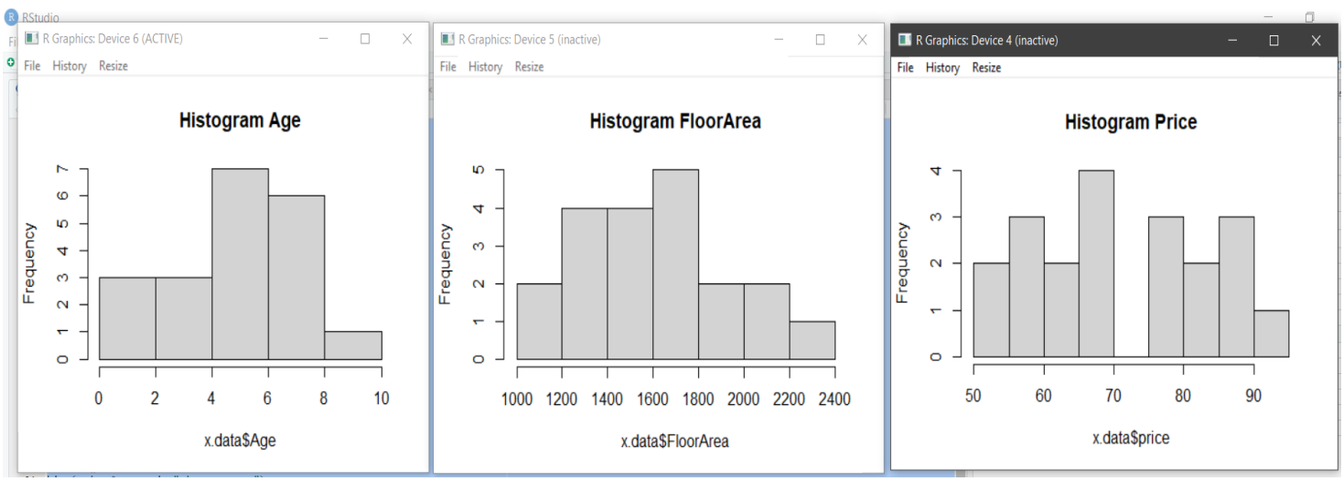
c) Plot Price vs. Floor, Price vs. Age, and Price vs. rooms, in separate graphs.

```
> windows()
> plot(x.data$price,x.data$FloorArea, xlab = "price", ylab = "FloorArea", main="Plot Graph")
> windows()
> plot(x.data$price,x.data$Age, xlab = "price", ylab = "Age", main="Plot Graph")
> windows()
> plot(x.data$price,x.data$Rooms, xlab = "price", ylab = "Rooms", main="Plot Graph")
```



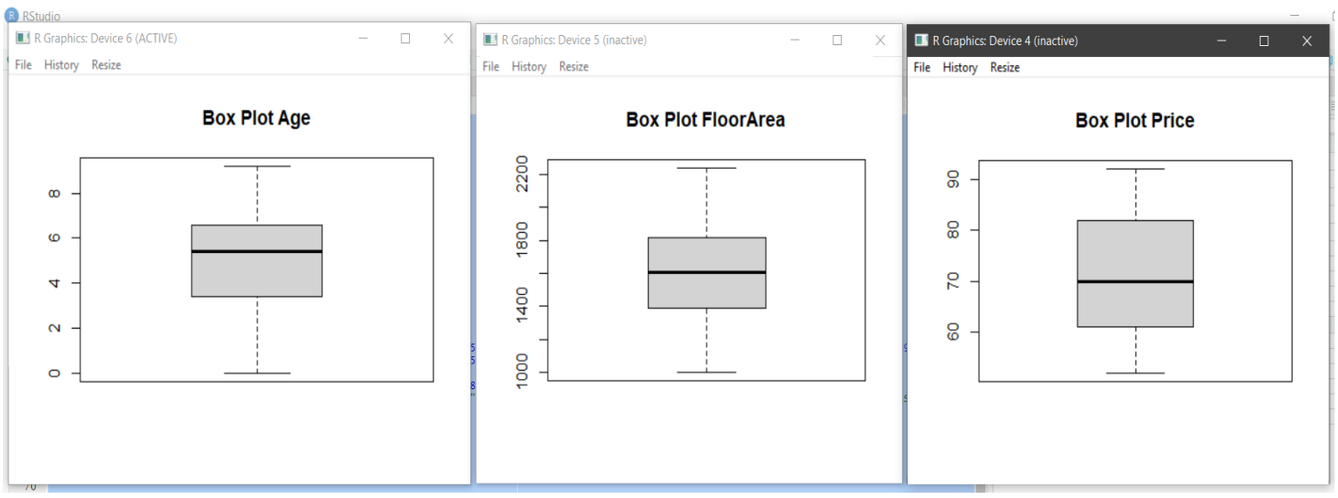
d) Draw histograms of Prices, FloorArea, and Age.

```
> windows()
> hist(x.data$price, main="Histogram Price")
> windows()
> hist(x.data$FloorArea, main="Histogram FloorArea")
> windows()
> hist(x.data$Age, main="Histogram Age")
```



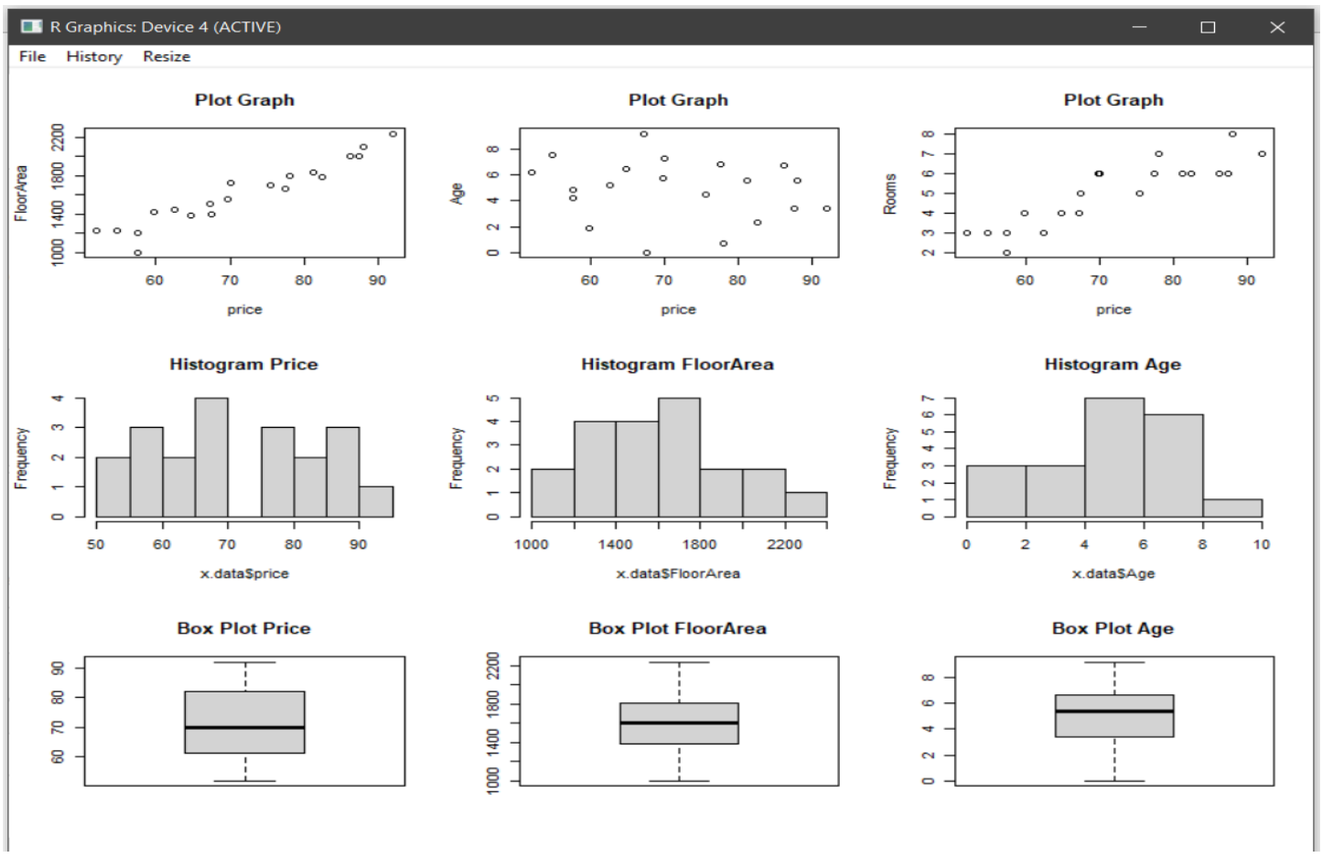
e) Draw box plots of Price, FloorArea, and Age.

```
> windows()
> boxplot(x.data$price, main="Box Plot Price")
> windows()
> boxplot(x.data$FloorArea, main="Box Plot FloorArea")
> windows()
> boxplot(x.data$Age, main="Box Plot Age")
```



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

```
> windows()
> par(mfrow=c(3,3))
>
> plot(x.data$price,x.data$FloorArea, xlab = "price", ylab = "FloorArea", main="Plot Graph")
> plot(x.data$price,x.data$Age, xlab = "price", ylab = "Age", main="Plot Graph")
> plot(x.data$price,x.data$Rooms, xlab = "price", ylab = "Rooms", main="Plot Graph")
> hist(x.data$price, main="Histogram Price")
> hist(x.data$FloorArea, main="Histogram FloorArea")
> hist(x.data$Age, main="Histogram Age")
> boxplot(x.data$price, main="Box Plot Price")
> boxplot(x.data$FloorArea, main="Box Plot FloorArea")
> boxplot(x.data$Age, main="Box Plot Age")
```



Day 3 : Descriptive Statistics

Date : 16-11-2020

Experiment:-

Collect at least 60 students and analyse the data by using descriptive statistics and Interpret the results.

The (data.csv) .csv file of 60 students data is uploaded → HERE ← you can download.

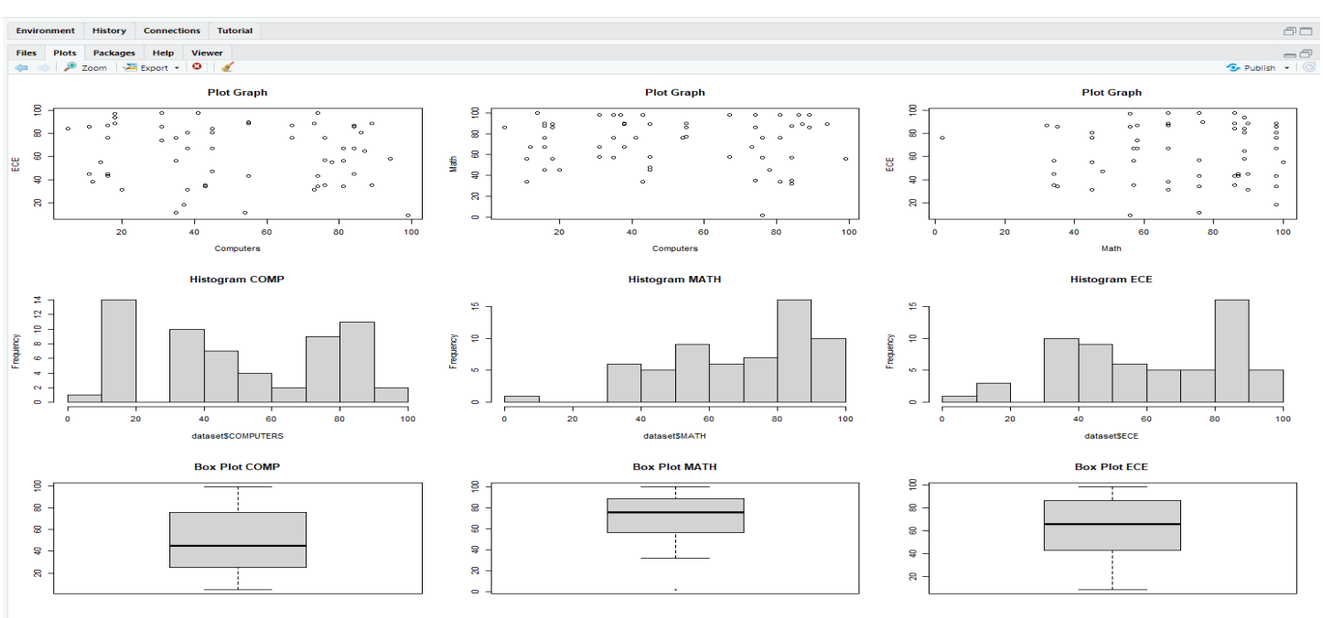
```
>
> #  getwd()
> setwd("E:\\VITAP\\19BCE7048\\Semester_3\\Applied Statistics\\R programming Lab\\Lab Materials\\Lab 3 - Descriptive Statistics")
>
> dataset=read.csv('data.csv',sep=',', header = TRUE)
> class(dataset)
[1] "data.frame"
> str(dataset)
'data.frame':   60 obs. of  4 variables:
 $ NAMES    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ COMPUTERS: int  81 35 76 84 74 87 67 45 31 43 ...
 $ MATH     : int  76 57 2 32 98 89 58 48 98 34 ...
 $ ECE      : int  34 56 76 87 43 65 87 47 86 35 ...
> mat <- as.matrix(dataset)
> class(mat)
[1] "matrix" "array"
> nrow(mat)
[1] 60
> ncol(mat)
[1] 4
>
> # first value in mat
> mat[1, 1]
NAMES
   1
>
> # a middle value in mat
> mat[4, 2]
COMPUTERS
     84
>
```


Submitted by Gudi Varaprasad – 19BCE7048

```
> mat[1:4, 1:2]
  NAMES COMPUTERS
[1,]    1      81
[2,]    2      35
[3,]    3      76
[4,]    4      84
>
> mat[5:8, 1:2]
  NAMES COMPUTERS
[1,]    5      74
[2,]    6      87
[3,]    7      67
[4,]    8      45
>
> mat[c(1,3,5), c(1,3)]
  NAMES MATH
[1,]    1   76
[2,]    3    2
[3,]    5   98
>
> # All columns from row 5
> mat[5, ]
  NAMES COMPUTERS     MATH      ECE
     5      74      98      43
>
> # first row, all of the columns
> col_1 <- mat[1, ]
>
> # max particle size for col_1
> max(col_1)
[1] 81
>
> # max particle size for col_2
> max(mat[2, ])
[1] 57
>
> # minimum particle size for operator 3
```

```
> min(mat[, 3])
[1] 2
>
> # mean for operator 3
> mean(mat[, 3])
[1] 71.16667
>
> # median for operator 3
> median(mat[, 3])
[1] 76
>
> # standard deviation for operator 3
> sd(mat[, 3])
[1] 22.64776
>
> #
> dataset=read.csv('data.csv',sep=',',header = TRUE)
> class(dataset)
[1] "data.frame"
> str(dataset)
'data.frame':   60 obs. of  4 variables:
 $ NAMES    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ COMPUTERS: int  81 35 76 84 74 87 67 45 31 43 ...
 $ MATH     : int  76 57 2 32 98 89 58 48 98 34 ...
 $ ECE      : int  34 56 76 87 43 65 87 47 86 35 ...
>
> mean(dataset$COMPUTERS)
[1] 50.48333
> median(dataset$COMPUTERS)
[1] 45
> sd(dataset$COMPUTERS)
[1] 27.66537
>
> mean(dataset$MATH)
[1] 71.16667
> median(dataset$MATH)
[1] 76
```

```
> sd(dataset$MATH)
[1] 22.64776
>
> mean(dataset$ECE)
[1] 61.58333
> median(dataset$ECE)
[1] 66
> sd(dataset$ECE)
[1] 25.2105
>
>
> #
> dataset=read.csv('data.csv',sep=',', header = TRUE)
> # stats graphs
> #
> par(mfrow=c(3,3))
> plot(dataset$COMPUTERS,dataset$ECE, xlab = "Computers", ylab = "ECE", main="Plot Graph")
> plot(dataset$COMPUTERS,dataset$MATH, xlab = "Computers", ylab = "Math", main="Plot Graph")
> plot(dataset$MATH,dataset$ECE, xlab = "Math", ylab = "ECE", main="Plot Graph")
>
> hist(dataset$COMPUTERS, main="Histogram COMP")
> hist(dataset$MATH, main="Histogram MATH")
> hist(dataset$ECE, main="Histogram ECE")
>
> boxplot(dataset$COMP, main="Box Plot COMP")
> boxplot(dataset$MATH, main="Box Plot MATH")
> boxplot(dataset$ECE, main="Box Plot ECE")
```



Day 4 : Random Sampling and Probability

Date : 21-11-2020

1. 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.

```
> x = (1:12)
> y = matrix(x,nrow = 3, ncol = 4, byrow = "TRUE")
> print(y)
   [,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.

```
> y = matrix(c(1:12),nrow = 3, ncol = 4, byrow = "TRUE")
> print(y)
   [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

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

```
> rownames(y) <- c("A","B","C")
> print(y)
  [,1] [,2] [,3] [,4]
A    1    2    3    4
B    5    6    7    8
C    9   10   11   12
```

d) Transpose of the matrix.

```
> trans = t(y)
> print(trans)
   A B  C
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
```


Submitted by Gudi Varaprasad – 19BCE7048

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

```
> #Creating a Matrix
> MatrixA <- matrix(data = 1:9, nrow = 3, ncol = 3)
> #Printing Matrix
> MatrixA
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> #Creating a new Matrix using rbind()
> MatrixB <- rbind(MatrixA, c(10,11,12))
> #Printing that Matrix
> MatrixB
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
[4,]   10   11   12
>
> #Creating a new Matrix using cbind()
> MatrixC <- cbind(MatrixA, c(10, 11, 12))
> #Printing Matrix
> MatrixC
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

f) Use arbitrary numbers to create matrix.

```
> matrix(sample(1:20), nrow=4, ncol=4)
     [,1] [,2] [,3] [,4]
[1,]   16    1   19    6
[2,]    2   13   10   12
[3,]    5    9   20   14
[4,]    3   11   15    8
```

g) Verify matrix multiplication.

```
>
> # matrix multiplication
> matrix(c(1,5,3,8), ncol=2, nrow=2)
     [,1] [,2]
[1,]    1    3
[2,]    5    8
>
> # matrix multiplication in R - element by element
> a = matrix(c(1,3,5,7), ncol=2, nrow=2)
> b = matrix(c(2,4,6,8), ncol=2, nrow=2)
> print(a*b)
     [,1] [,2]
[1,]    2   30
[2,]   12   56
>
> # matrix multiplication in R - algebraic
> m1 = a %*% b
> print(m1)
     [,1] [,2]
[1,]   22   46
[2,]   34   74
> m2 = b %*% a
> print(m2)
     [,1] [,2]
[1,]   20   52
[2,]   28   76
> identical(a, b)
[1] FALSE
```

2. Random sampling

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

```
> arb = sample(1:40, 5, replace=T)
> print(arb)
[1] 24  3  7 29 20
```

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.

```
> sample(c("H", "T"), 10, replace=T)
 [1] "H" "H" "H" "T" "T" "H" "H" "T" "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.

```
> sample(c("success", "failure"), 10, replace=T, prob=c(90,10))
 [1] "success" "success" "success" "success" "success" "success" "success" "success" "success" "success"
```

d) The *choose* function can be used to calculate the following express 40 C 5 :

```
> choose(40, 5)
[1] 658008
```

e) Find 5!

```
>
> factorial <- function(n) {
+   if(n <= 1) {
+     return(1)
+   } else {
+     return(n * factorial(n-1))
+   }
+ }
> factorial(5)
[1] 120
```

Day 5 : Binomial Distribution

Date : 23-11-2020

1. Five terminals on an on-line 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.

a) Find the probability of getting exactly 3 ready terminals.

```
> dbinom(x = 3, size =5, prob = 0.95)
```

```
[1] 0.02143438
```

b) Find all the probabilities.

```
> x<- 0:5
```

```
> dbinom(x, size= 5, prob =0.95)
```

```
[1] 0.0000003125 0.0000296875 0.0011281250 0.0214343750 0.2036265625 0.7737809375
```

2. 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.
Find the probability of different number of defective found in the sample?

```
> x<- 0:20
```

```
> dbinom(x, 20, 0.2)
```

```
 [1] 1.152922e-02 5.764608e-02 1.369094e-01 2.053641e-01 2.181994e-01 1.745595e-01
1.090997e-01 5.454985e-02 2.216088e-02 7.386959e-03 2.031414e-03

[12] 4.616849e-04 8.656592e-05 1.331783e-05 1.664729e-06 1.664729e-07 1.300570e-08
7.650410e-10 3.187671e-11 8.388608e-13 1.048576e-14
```

3. It is known that 1% of bits transmitted through a digital transmission are received in error. One hundred bits are transmitted each day. Find the probability of different number of bits found in error each day.?

```
> x<- 0:100
```

```
> dbinom(x,100, 0.01)
```

```
 [1] 3.660323e-01 3.697296e-01 1.848648e-01 6.099917e-02 1.494171e-02 2.897787e-03
4.634508e-04 6.286346e-05 7.381694e-06 7.621951e-07

[11] 7.006036e-08 5.790112e-09 4.337710e-10 2.965956e-11 1.861747e-12 1.078184e-13
5.785707e-15 2.887697e-16 1.344999e-17 5.863367e-19

[21] 2.398650e-20 9.230014e-22 3.347893e-23 1.146841e-24 3.716614e-26 1.141263e-27
3.325359e-29 9.206008e-31 2.424382e-32 6.079954e-34

[31] 1.453457e-35 3.315151e-37 7.220499e-39 1.502889e-40 2.991491e-42 5.698078e-44
1.039212e-45 1.815713e-47 3.040667e-49 4.882708e-51

[41] 7.521343e-53 1.111802e-54 1.577594e-56 2.149411e-58 2.812590e-60 3.535467e-62
4.269888e-64 4.955382e-66 5.526836e-68 5.924459e-70

[51] 6.103988e-72 6.044749e-74 5.753549e-76 5.263395e-78 4.627377e-80 3.909263e-82
3.173103e-84 2.474154e-86 1.852815e-88 1.332276e-90
```


Submitted by Gudi Varaprasad – 19BCE7048

[61] 9.195842e-93 6.090970e-95 3.870118e-97 2.357936e-99 1.376951e-101 7.703225e-104 4.126306e-106 2.115098e-108 1.036813e-110 4.856976e-113

[71] 2.172673e-115 9.273039e-118 3.772701e-120 1.461680e-122 5.387028e-125 1.886367e-127 6.267832e-130 1.973343e-132 5.877609e-135 1.653336e-137

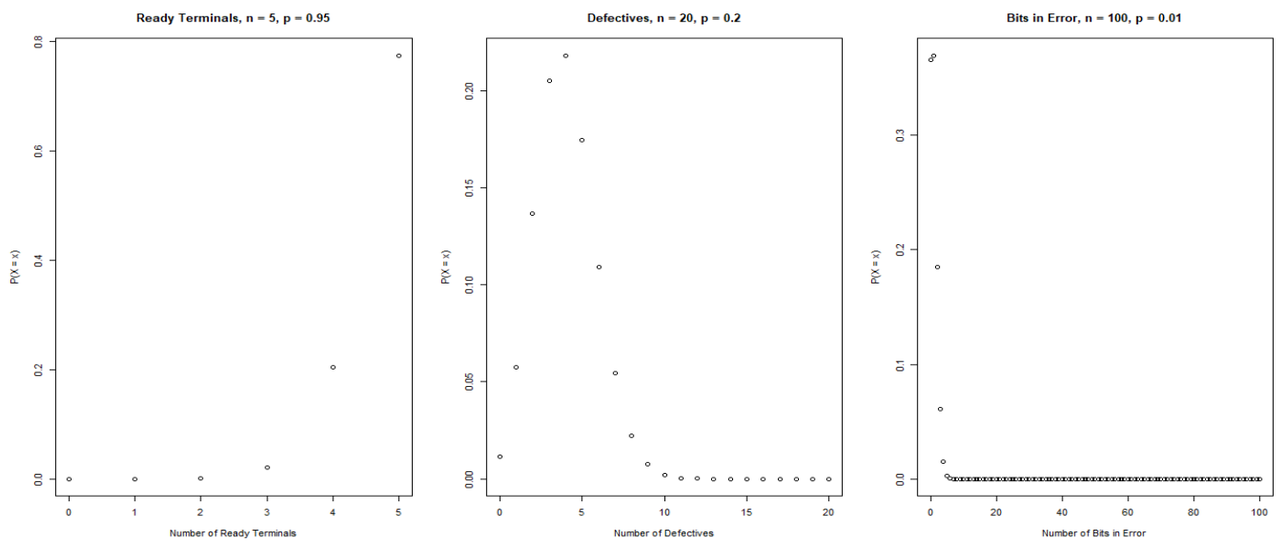
[81] 4.383845e-140 1.093365e-142 2.558996e-145 5.605686e-148 1.145943e-150 2.178859e-153 3.838722e-156 6.239651e-159 9.310773e-162 1.268066e-164

[91] 1.565513e-167 1.737722e-170 1.717116e-173 1.492009e-176 1.122294e-179 7.159768e-183 3.766713e-186 1.568973e-189 4.851495e-193 9.900000e-197

[101] 1.000000e-200

4. Plot all of the above problems in a single window for random variable and respective Probability distribution.

```
> par(mfrow = c(1,3)) # single window
>
> x<-0:5 #question 1
> plot(x, dbinom(x, size = 5, prob = 0.95),
+     xlab = "Number of Ready Terminals",
+     ylab = "P(X = x)",
+     main = "Ready Terminals, n = 5, p = 0.95")
>
> x<-0:20 #question 2
> plot(x, dbinom(x, size = 20, prob = 0.2),
+     xlab = "Number of Defectives",
+     ylab = "P(X = x)",
+     main = "Defectives, n = 20, p = 0.2")
> x<-0:100 #question 3
> plot(x, dbinom(x, size = 100, prob = 0.01),
+     xlab = "Number of Bits in Error",
+     ylab = "P(X = x)",
+     main = "Bits in Error, n = 100, p = 0.01")
```



5. For Q.No. 1 Find P(X ≤ 3) and P(X > 3). For Q. No. 2 Find P(X ≤ 4) and P(X > 4). Find all the cumulative probabilities and round to 4 decimal places.

```
> # P(X <= 3): n = 5, p = 0.95
>   pbinom(3, 5, 0.95)
[1] 0.0225925
>
> # P(X > 3):
>   1-pbinom(3, 5, 0.95)
[1] 0.9774075
>
>
> # P(X <= 4): n = 20, p = 0.2
>   pbinom (4, size = 20, prob =0.2)
[1] 0.6296483
>
> # P(X > 4) = 1- P(X <= 4 )
>   1- pbinom(4, size = 20, prob = 0.2)
[1] 0.3703517
>
>   x<-0:20
>   prob<- pbinom(x, size = 20, prob=0.2)
> # round to 4 decimal places:
>   round(prob, 4)
 [1] 0.0115 0.0692 0.2061 0.4114 0.6296 0.8042 0.9133 0.9679 0.9900 0.9974 0.9994 0.9999 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
```

6. The probability that a patient recover from a rare blood disease is 0.4. If 15 people are known to have contracted this disease, what is the probability that

(a) at least10 survive,

```
>   1-pbinom(10,15,0.4) + dbinom(x = 10, size =15, prob = 0.4)
[1] 0.0338333
```

(b) from 3 to 8 survive, and

```
>   pbinom(8,15,0.4) - pbinom(3,15,0.4) + dbinom(3,15,0.4)
[1] 0.8778386
```

(c) exactly 5 survive?

```
>   dbinom(x = 5, size =15, prob = 0.4)
[1] 0.1859378
```

7. Write your own function for Binomial Distribution and cumulative binomial distribution.

```
> # for Binomial Distribution
> binomialfunc = function(x,n,p) {
+    bprob = choose(n,x)*(p^x)*((1-p)^(n-x))
+    return(bprob)
+ }
> binomialfunc(7,17,0.3)  #example
[1] 0.1201446
>
> # cumulative binomial distribution function
> cbinomialfunc = function(x,n,p) {
+   cbprob = sum(choose(n,x)*(p^x)*((1-p)^(n-x)))
+   return(cbprob)
+ }
> cbinomialfunc(1:7,17,0.55)  #example
[1] 0.1834065
>
```

Day 6 : Poisson Distribution

Date : 24-11-2020

1. During a laboratory experiment, the average number of radioactive particles passing through a counter in 1 millisecond is 4. What is the probability that 6 particles enter the counter in a given millisecond?

> # lambda = 4, x=6

> # dpois(x, lambda, log = FALSE)

> dpois(6, 4, log = FALSE)

[1] 0.1041956

2. In a certain industrial facility, accidents occur infrequently. It is known that the probability of an accident on any given day is 0.05 and accidents are independent of each other.
(a) What is the probability that in any given period of 4000 days there will be an accident on one day?

> # lambda = 4000*0.05 = 200

> dpois(1, 200, log = FALSE)

[1] 2.767793e-85

(b) What is the probability that there are at most three days with an accident?

> # P(x<=3) ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)

> ppois(3, 200, lower.tail = TRUE, log.p = FALSE)

[1] 1.873151e-81

3. In a manufacturing process where glass products are made, defects or bubbles occur, occasionally rendering the piece undesirable for marketing. It is known that, on average, 1 in every 1000 of these items produced has one or more bubbles. What is the probability that a random sample of 8000 will yield fewer than 7 items possessing bubbles?

> # p = 1/1000 = 0.001 , n=8000 , mean = 8 , find P(X < 7) = P(X <= 6)

> ppois(6, 8, lower.tail = TRUE, log.p = FALSE)

[1] 0.3133743

4. On average, 3 traffic accidents per month occur at a certain intersection. What is the probability that in any given month at this intersection
(a) exactly 5 accidents will occur?

> # mean = 3

> dpois(5, 3, log = FALSE)

[1] 0.1008188

(b) fewer than 3 accidents will occur?

> # find P(X < 3) = P(X <= 2)

> ppois(2, 3, lower.tail = TRUE, log.p = FALSE)

[1] 0.4231901

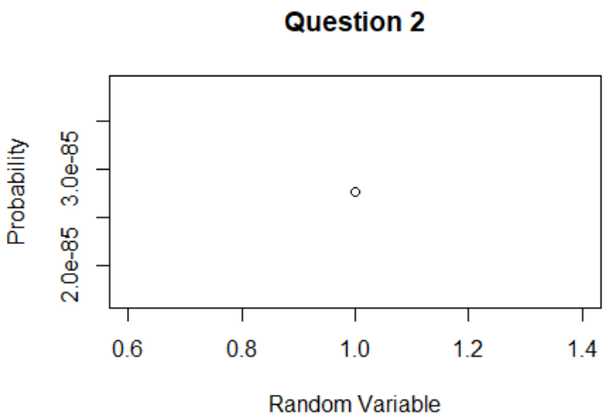
Submitted by Gudi Varaprasad – 19BCE7048

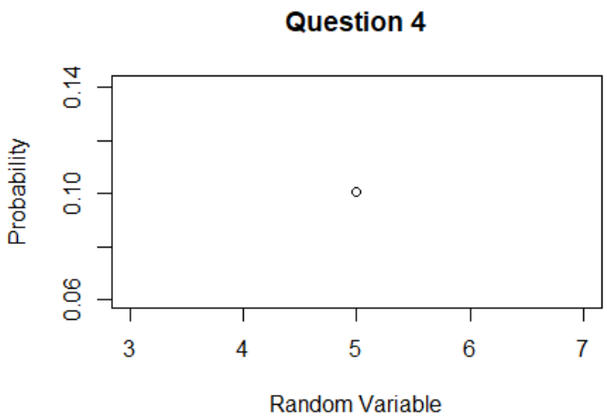
(c) at least 2 accidents will occur?

```
> # P(X >= 2) = 1 - P(X < 2) = 1 - P(X <= 1)
> 1 - ppois(1, 3, lower.tail = TRUE, log.p = FALSE)
[1] 0.8008517
```

5. Plot the graph for Q. No. 2 and 4 for Random Variable against Probability Distribution function.

```
> windows()
> par(mfrow = c(1,2)) # single window
> q2 = 2.767793e-85 #question 2
> plot(1,q2,xlab="Random Variable",ylab="Probability", main="Question 2")
>
> q4 = 0.1008188 #question 4
> plot(5,q4,xlab="Random Variable",ylab="Probability", main="Question 4")
```





6. A company makes electric motors. The probability an electric motor is defective is 0.01. What is the probability that a sample of 300 electric motors will contain x defective motors, where $5 \leq x < 8$?

```
> # lambda = 300*0.01 = 3
> sum(dpois(5:7, 3, log = FALSE))
[1] 0.1728323
```

Day 7 : Normal Distribution

Date : 28-11-2020

1) IQ is a normal distribution of mean of 100 and standard deviation of 15
a) What percentage of people have an IQ<125?

```
> # pnorm(x, mean, sd)
> pnorm(125, 100, 15) * 100
[1] 95.22096
```

b) What percentage of people have IQ>110?

```
> (1 - pnorm(110, 100, 15))* 100
[1] 25.24925
```

c) What percentage of people have 110<IQ<125?

```
> (pnorm(125,100,15,lower.tail=TRUE) - pnorm(110,100,15,lower.tail=TRUE)) * 100
[1] 20.47022
```

d) Find 25% for standard normal distribution.

```
> qnorm(0.25)
[1] -0.6744898
```

e) Find 25% normal distribution with mean and standard deviation 2& 3.

```
> qnorm(0.25,2,3,lower.tail=TRUE)
[1]  -0.02346925
```

f) What IQ separates the lower 25% from the others.

```
> qnorm(0.25,100,15,lower.tail=TRUE)
[1] 89.88265
```

g) What IQ separates the top 25% from the others.

```
> qnorm(0.25,100,15,lower.tail=FALSE)
[1] 110.1173
> # or same as
> qnorm(0.75,100,15,lower.tail=TRUE)
[1] 110.1173
```

h) Find 25 percentile for mean 100 and SD 15.

```
> qnorm(0.25,100,15)
[1] 89.88265
```

2) Generate the 20 random number for a normal distribution with mean 572 and SD is 51. Calculate mean and SD of data set.

Submitted by Gudi Varaprasad – 19BCE7048

> r = rnorm(20,572,51)

> mean(r)

[1] 571.1462

> sd(r)

[1] 48.73847

3) Make appropriate histogram of data in above question and visually assume if normal density curve & histogram density estimates are similar.

> # red for histogram || blue for normal curve

> d = rnorm(20,572,51)

> meann = mean(d)

> stdv = sd(d)

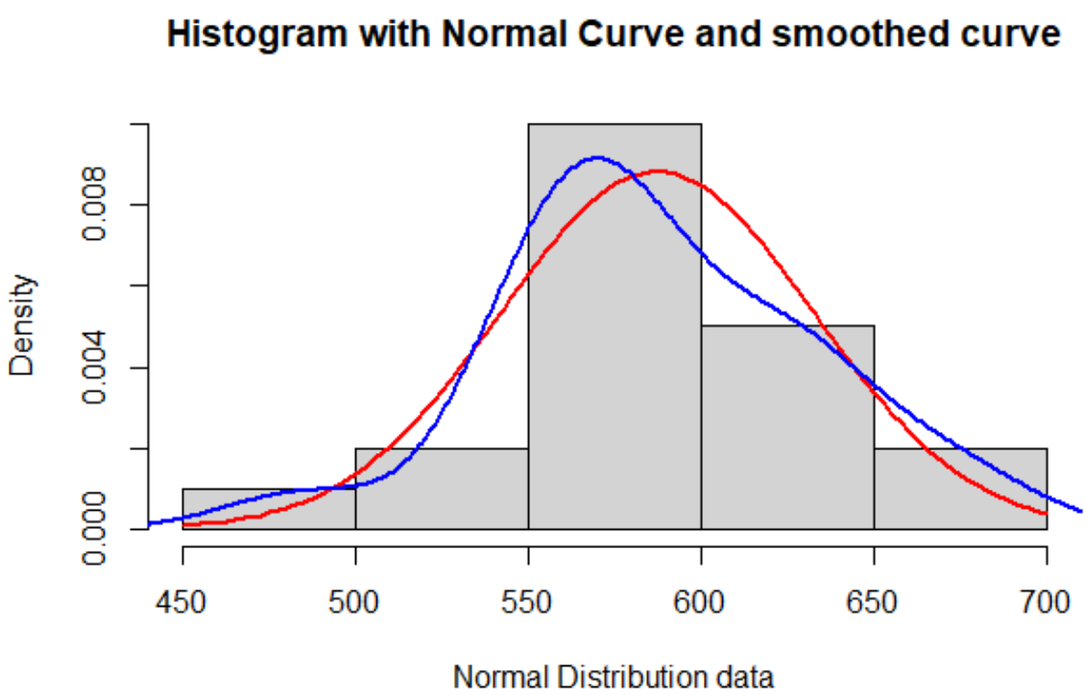
> windows()

> hist(d, xlab="Normal Distribution data", freq = FALSE,

+ main = "Histogram with Normal Curve and smoothed curve")

> curve(dnorm(x, mean=meann, sd=stdv), col="red", lwd=2, add=TRUE)

> lines(density(d, adjust = 1), col="blue", lwd=2)



Day 8 : Z Test

Date : 30-11-2020

1. Test the hypothesis that the mean systolic blood pressure in a certain population equals 140 mmHg. The standard deviation has a known value of 20 and a data set of 55 patients is available.

120, 115, 94, 118, 111, 102, 102, 131, 105, 107, 115, 139, 115, 113, 114, 105, 115, 134, 109, 109, 93, 118, 109, 106, 125, 150, 142, 119, 127, 141, 149, 144, 142, 149, 161, 143, 140, 148, 149, 141, 146, 159, 152, 135, 134, 161, 130, 125, 141, 148, 153, 145, 137, 147, 175

```
> # 1 H0: Mean=140 H1: Mean!=140(two tailed)  here alpha=0.05
>
> Mean=140
> stdv=20
> n=55
> alpha=0.05
>
data=c(120,115,94,118,111,102,102,131,105,107,115,139,115,113,114,105,115,134,109,109,93,118,109,106,125,150,142,119,127,141,149,144,142,149,161,143,140,148,149,141,146,159,152,135,134,161,130,125,141,148,153,145,137,147,175)
> X=mean(data)
>
> Z=sqrt(n)*(X-Mean)/stdv
> print(Z)
[1] -3.660905
>
> pValue=2*(1-pnorm(abs(Z))) # two tailed
> print(pValue)
[1] 0.0002513257
>
> if(pValue<=alpha)
+ {
+   print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "Ho is rejected and H1 is accepted"
```

2. A coin is tossed 100 times and turns up head 43 times Test the claim that this is a fair coin. Use 5% level of significance to test the claim.

```
> #2  H0:coin is fair, Mean=0.5  H1: Mean!=0.5 (two tailed) here alpha=0.05
>
> Mean=0.5
> n=100
> stdv=sqrt(n*0.5*0.5)
> X=0.43
> alpha=0.05
>
> Z=sqrt(n)*(X-Mean)/stdv
> print(Z)
[1] -0.14
>
> pValue=2*(1-pnorm(abs(Z))) # two tailed
> print(pValue)
[1] 0.88866
>
> if(pValue<=alpha)
+ {
+   print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "H0 is accepted and H1 is rejected"
```

3. A manufacturer of sports equipment has developed a new synthetic fishing line that the company claims has a mean breaking strength of 8 kilograms with a standard deviation of 0.5 kilogram. Test the hypothesis that μ = 8 kilograms against the alternative that μ is not equal to 8 kilograms if a random sample of 50 lines is tested and found to have a mean breaking strength of 7.8 kilograms. Use a 0.01 level of significance.

```
> #3 HO:Mean=8  H1: Mean!=8 (two tailed) here alpha=0.01
>
> Mean=8
> n=50
> stdv=0.5
> X=7.8
> alpha=0.01
>
> Z=sqrt(n)*(X-Mean)/stdv
```

```
> print(Z)
[1] -2.828427
>
> pValue=2*(1-pnorm(abs(Z))) # two tailed
> print(pValue)
[1] 0.004677735
>
> if(pValue<=alpha)
+ {
+   print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "Ho is rejected and H1 is accepted"
```

Day 9 : T Test

Date : 01-12-2020

1. An outbreak of salmonella-related illness was attributed to ice produced at a certain factory. Scientists measured the level of Salmonella in 9 randomly sampled batches ice cream. The levels(in MPN/g) were:

0.593 0.142 0.329 0.691 0.231 0.793 0.519 0.392 0.418

Is there evidence that the mean level pf Salmonella in ice cream greater than 0.3 MPN/g?

```
> # 1 H0: Mean=0.3 H1: Mean>0.3(right tailed)  here alpha=0.05
>
> x<-c(0.593, 0.142, 0.329, 0.691, 0.231, 0.793, 0.519, 0.392, 0.418)
>
> # t.test(x,y=NULL,alternative=c("two.sided","less","greater"), mu=0, paired=FALSE,
var.equal=FALSE, conf.level=0.95)
>
> t.test(x,alternative="greater",mu = 0.3, paired = FALSE, var.equal =FALSE,conf.level = 0.95)

        One Sample t-test

data:  x
t = 2.2051, df = 8, p-value = 0.02927
alternative hypothesis: true mean is greater than 0.3
95 percent confidence interval:
 0.3245133      Inf
sample estimates:
mean of x
0.4564444

>
> p=0.02927 #obtained from t test
>
> if(p<=0.05)
+ {
+  print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "Ho is rejected and H1 is accepted"
```

2. Suppose that 10 volunteers have taken an intelligence test; here are the results obtained. The average score of the entire population is 75 in the same test. Is there any significant difference (with a significance level of 95%) between the sample and population means, assuming that the variance of the population is not known? **Scores:** 65, 78, 88, 55, 48, 95, 66, 57, 79, 81

```
> # 2  H0: Mean=75 H1: Mean!=75 (two tailed)  here alpha=0.95
>
> x<-c(65, 78, 88, 55, 48, 95, 66, 57, 79, 81)
> t.test(x,alternative="two.sided",mu =75, paired = FALSE, var.equal = FALSE,conf.level = 0.05)

        One Sample t-test

data:  x
t = -0.78303, df = 9, p-value = 0.4537
alternative hypothesis: true mean is not equal to 75
5 percent confidence interval:
 70.8871 71.5129
sample estimates:
mean of x
    71.2

>
> p=0.4537  #obtained from t test
>
> if(p<=0.95)
+ {
+   print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "Ho is rejected and H1 is accepted"
```

3. Comparing two independent sample means, taken from two populations with unknown variance. The following data shows the heights of individuals of two different countries with unknown population variances. Is there any significant difference b/n the average heights of two groups?

A: 175 168 168 190 156 181 182 175 174 179

B: 185 169 173 173 188 186 175 174 179 180

```
> # 3 H0: MeanA-MeanB=0 H1: MeanA-MeanB!=0 (two tailed)  here alpha=0.05
>
```

```
> A<-c(175, 168, 168, 190, 156, 181, 182, 175, 174, 179)
> B<-c(185, 169, 173, 173, 188, 186, 175, 174, 179, 180)
> t.test(A,B,alternative = "two.sided",mu = 0, paired = FALSE, var.equal= FALSE,conf.level =0.95)

        Welch Two Sample t-test

data:  A and B
t = -0.94737, df = 15.981, p-value = 0.3576
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -11.008795   4.208795
sample estimates:
mean of x mean of y
   174.8     178.2

>
> p=0.3576  #obtained from t test
>
> if(p<=0.05)
+ {
+   print("Ho is rejected and H1 is accepted")
+ }else {
+   print("H0 is accepted and H1 is rejected")
+ }
[1] "H0 is accepted and H1 is rejected"
```

Day 10 : Correlation

Date : 05-12-2020

Q.1) It is important that scientific researchers in the area of forest products be able to study correlation among the anatomy and mechanical properties of trees. For the study Quantitative Anatomical Characteristics of Plantation Grown Loblolly Pine (Pinus Taeda L.) and Cottonwood (Populus deltoides Bart. Ex Marsh.) and Their Relationships to Mechanical Properties, conducted by the Department of Forestry and Forest Products at Virginia Tech, 29 loblolly pines were randomly selected for investigation. Table shows the resulting data on the specific gravity in grams/cm3 and the modulus of rupture in kilopascals (kPa). Compute and interpret the sample correlation coefficient.

Specific Gravity, x (g/cm^3)	Modulus of Rupture, y (kPa)	Specific Gravity, x (g/cm^3)	Modulus of Rupture, y (kPa)
0.414	29,186	0.581	85,156
0.383	29,266	0.557	69,571
0.399	26,215	0.550	84,160
0.402	30,162	0.531	73,466
0.442	38,867	0.550	78,610
0.422	37,831	0.556	67,657
0.466	44,576	0.523	74,017
0.500	46,097	0.602	87,291
0.514	59,698	0.569	86,836
0.530	67,705	0.544	82,540
0.569	66,088	0.557	81,699
0.558	78,486	0.530	82,096
0.577	89,869	0.547	75,657
0.572	77,369	0.585	80,490
0.548	67,095		

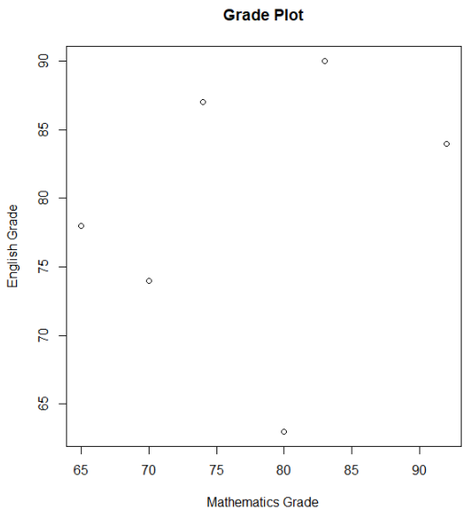
```
> #getwd()
> setwd("E:\\VITAP\\19BCE7048\\Semester_3\\Applied Statistics\\R programming Lab\\Lab Materials\\Lab 10 - Correlation")
> #getwd()
>
> dataset=read.csv('Question1.csv',sep=',',header = TRUE)
>
> x = dataset$ï..SG
> y = dataset$MR
>
> meanx = mean(x)
> meany = mean(y)
>
> Sxx = sum((x - meanx) * (x - meanx))
> Syy = sum((y - meany) * (y - meany))
> Sxy = sum((x - meanx) * (y - meany))
```

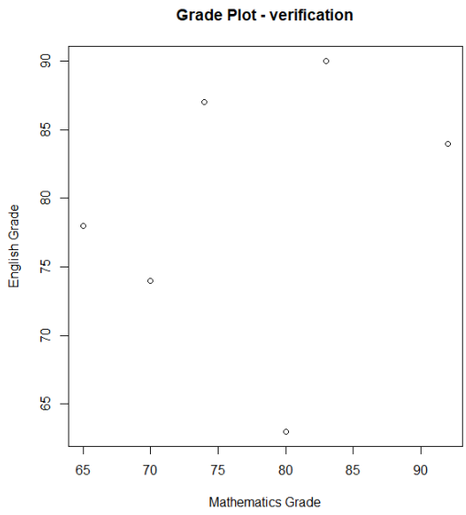
```
>
> r = Sxy / (sqrt(Sxx*Syy))
> print(r)
[1] 0.9434969
>
> # Verification:
> cor(x, y, method="pearson")
[1] 0.9434969
```

Q.2) Compute and interpret the correlation coefficient for the following grades of 6 students selected at random:

Mathematics grade	70	92	80	74	65	83
English grade	74	84	63	87	78	90

```
> x = c(70,92,80,74,65,83)
> y = c(74,84,63,87,78,90)
>
> meanx = mean(x)
> meany = mean(y)
>
> Sxx = sum((x - meanx) * (x - meanx))
> Syy = sum((y - meany) * (y - meany))
> Sxy = sum((x - meanx) * (y - meany))
>
> r = Sxy / (sqrt(Sxx*Syy))
> print(r)
[1] 0.2396639
> windows()
> par(mfrow=c(1,2))
> plot(x,y, xlab = "Mathematics Grade", ylab = "English Grade", main = "Grade Plot")
>
> # Verification:
> cor(x, y, method="pearson")
[1] 0.2396639
> plot(x,y, xlab = "Mathematics Grade", ylab = "English Grade", main = "Grade Plot - verification")
```





Q.3) Assume that x and y are random variables with a bivariate normal distribution. Calculate r.

Individual	Arm Strength, x	Dynamic Lift, y
1	17.3	71.7
2	19.3	48.3
3	19.5	88.3
4	19.7	75.0
5	22.9	91.7
6	23.1	100.0
7	26.4	73.3
8	26.8	65.0
9	27.6	75.0
10	28.1	88.3
11	28.2	68.3
12	28.7	96.7
13	29.0	76.7
14	29.6	78.3
15	29.9	60.0
16	29.9	71.7
17	30.3	85.0
18	31.3	85.0
19	36.0	88.3
20	39.5	100.0
21	40.4	100.0
22	44.3	100.0
23	44.6	91.7
24	50.4	100.0
25	55.9	71.7

```
> #getwd()
> setwd("E:\\VITAP\\19BCE7048\\Semester_3\\Applied Statistics\\R programming Lab\\Lab Materials\\Lab 10 - Correlation")
> #getwd()
>
> dataset=read.csv('Question3.csv',sep=',',header = TRUE)
>
> x = dataset$AS
> y = dataset$DL
>
> meanx = mean(x)
> meany = mean(y)
>
```

```
> Sxx = sum((x - meanx) * (x - meanx))
> Syy = sum((y - meany) * (y - meany))
> Sxy = sum((x - meanx) * (y - meany))
>
> r = Sxy / (sqrt(Sxx*Syy))
> print(r)
[1] 0.3916965
>
> # Verification:
> cor(x, y, method="pearson")
[1] 0.3916965
```

Day 11 : Regression

Date : 07-12-2020

1) In a certain type of metal test specimen, the normal stress on a specimen is known to be functionally related to the shear resistance. The following is a set of coded experimental data on the two variables:

Normal Stress	Shear Resistance
26.8	26.5
25.4	27.3
28.9	24.2
23.6	27.1
27.7	23.6
33.9	25.9
24.7	26.3
28.1	22.5
26.9	21.7
27.4	21.4
26.6	28.8
25.6	24.9

```
> dataset=read.csv('dataset.csv',sep=',',header = TRUE)
>
> x = dataset$ï..NS
> y = dataset$SR
>
> meanx=mean(x)
> meany=mean(y)
>
> beta1 = sum((x-meanx)*(y-meany)) / (sum((x-meanx)^2))
> print(" Slope = ")
[1] " Slope = "
> print(beta1)
[1] -0.2104415
>
> beta0 = meany-(beta1*meanx)
> print(" Intercept = ")
[1] " Intercept = "
> print(beta0)
[1] 30.72665
```

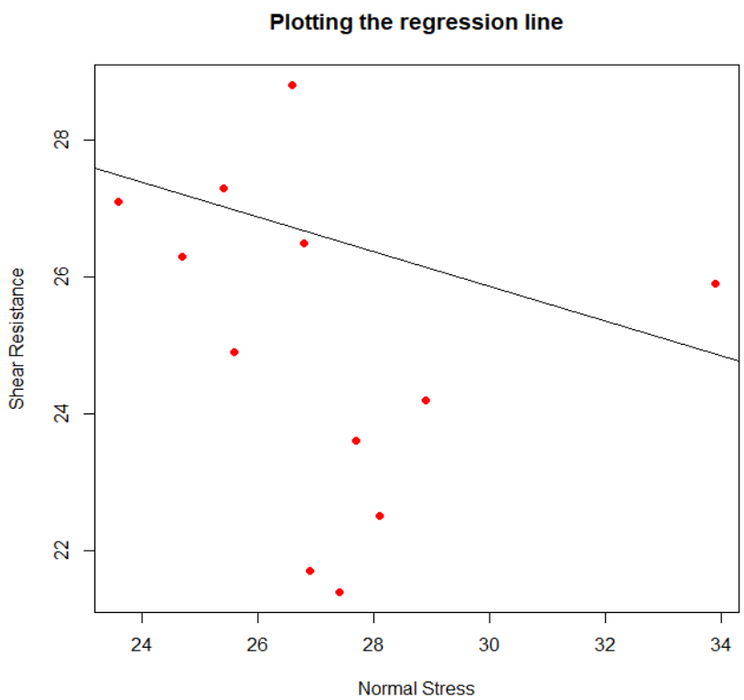
a) Estimate the shear resistance for a normal stress of 27.5.

```
> shear_stress = beta0 + beta1*27.5
> print(shear_stress)
[1] 24.9395
```


Submitted by Gudi Varaprasad – 19BCE7048

b) Plot the data; does it appear that a simple linear regression will be a suitable model?

```
> dataset=read.csv('dataset.csv',sep=',', header = TRUE)
>
> x = dataset$ï..NS
> y = dataset$SR
>
> windows()
> plot(x, y, col="red", main="Plotting the regression line", xlab="Normal Stress", ylab="Shear Resistance")
>
> y = beta0 + beta1*x
> par(new=TRUE)
> plot(x, y, col="black", type="l", pch=16, main="Plotting the regression line", xlab="Normal Stress", ylab="Shear Resistance")
```




```
>  # Conclusion :
>  # it appears that a simple linear regression will be a suitable model, this is a good model because line is passing near to the data.
```

Verification using **lm( )** inbuilt function :

```
> dataset=read.csv('dataset.csv',sep=',',header = TRUE)
> x = dataset$ï..NS
> y = dataset$SR
> relation = lm(y~x)  #to get the relation between normal stress and shear resistance i.e. slope, beta0 value
```

> print(relation)

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept) x

30.7266 -0.2104

> # print(summary(relation))

> a = data.frame(x=27.5) #giving the value at which we need to find the shear resistance

> result = predict(relation,a) #using predict to find the value

> print(result)

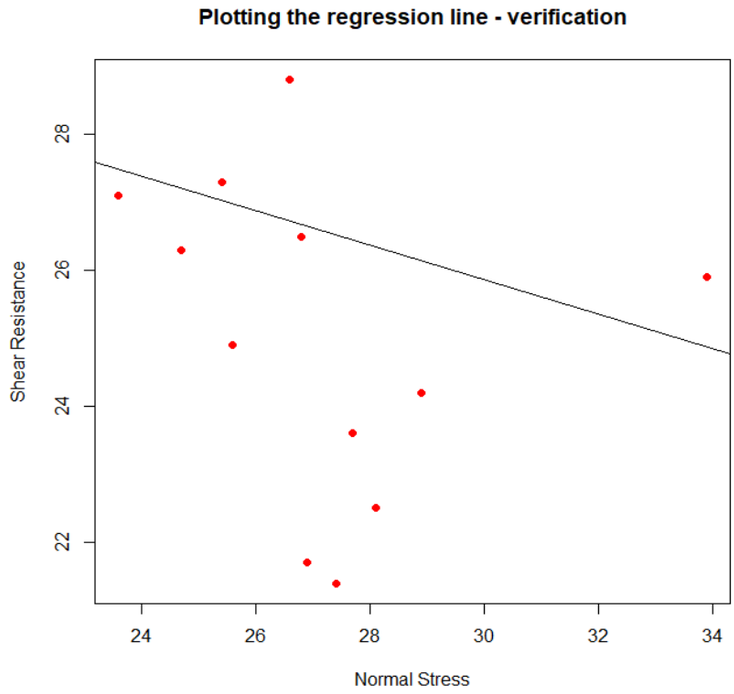
1

24.9395

> # b)

> windows()

> plot(x, y, col="red", main="Plotting the regression line - verification", abline(lm(x~y)), pch=16, xlab="Normal Stress", ylab="Shear Resistance")



> # Conclusion :

> # it appears that a simple linear regression will be a suitable model, this is a good model because line is passing near to the data.

Day 12 :

Date : 08-12-2020

Case Study: Working Through a Real-Life Problem

1. Introduction :

We look at a sample real-life problem and the R commands necessary to explore the problem. It is based on all of the commands discussed throughout this lab course.

2. Problem Statement :

In the following examples we will look at the carbon monoxide data which is one of the columns of this data set. First, we will transform the data so that it is close to being normally distributed. We will then find the confidence interval for the mean and then perform a significance test to evaluate whether or not the data is away from a fixed standard. Finally, we will find the power of the test to detect a fixed difference from that standard. We will assume that a confidence level of 95% is used throughout.

3. Analysis of the Data :

The (table.csv) .csv file used in this problem (carbon monoxide data) is uploaded → HERE ← you can download.

```
> engine <- read.csv(file="table.csv",sep=",",head=TRUE)
> names(engine)
[1] "en"  "hc"  "co"  "nox"
> summary(engine)
      en              hc              co             nox
 Min.   : 1.00   Min.   :0.3400   Min.   : 1.850   Min.   :0.490
 1st Qu.:12.75   1st Qu.:0.4375   1st Qu.: 4.388   1st Qu.:1.110
 Median :24.50   Median :0.5100   Median : 5.905   Median :1.315
 Mean   :24.00   Mean   :0.5502   Mean   : 7.879   Mean   :1.340
 3rd Qu.:35.25   3rd Qu.:0.6025   3rd Qu.:10.015   3rd Qu.:1.495
 Max.   :46.00   Max.   :1.1000   Max.   :23.530   Max.   :2.940
```

Analysis 1 : A boxplot is show in Figure 1. showing that the data appears to be skewed. This is further confirmed in the histogram which is shown in Figure 2. Finally, a normal qq plot is given in Figure 3. The data does not appear to be normal.

```
> qqnorm(engine$co,main="Carbon Monoxide")
> qqline(engine$co)
> windows()
> par(mfrow=c(1,3))
> boxplot(engine$co,main="Carbon Monoxide")
> hist(engine$co,main="Carbon Monoxide")
```

```
> qqnorm(engine$co,main="Carbon Monoxide")
> qqline(engine$co)
```

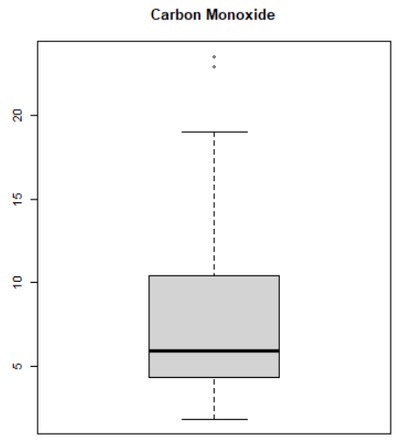


Fig 1 : Boxplot

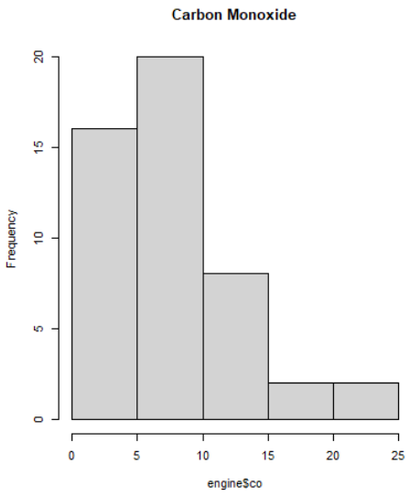


Figure 2 : Histogram

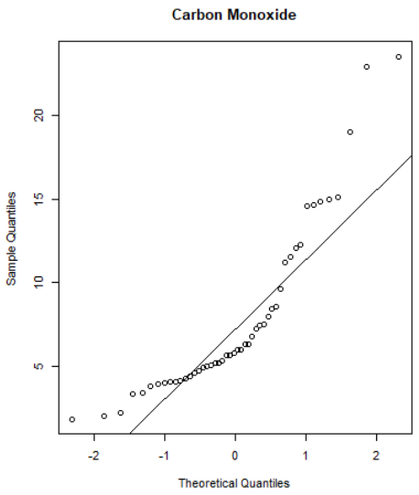


Figure 3 : Normal QQ Plot

Analysis 2 : We next see if the data can be transformed to something that is closer to being normally distributed. We examine the logarithm of the data. First, the boxplot of the log of the data appears to be more evenly distributed as shown in Figure 4. Also, the histogram appears to be centred and closer to normal in Figure 5. Finally, the normal qq plot is shown in in Figure 6. It shows that the data is more consistent with what we would expect from normal data.

```
> lengine <- log(engine$co)
> windows()
> par(mfrow=c(1,3))
> boxplot(lengine,main="Carbon Monoxide")
> hist(lengine,main="Carbon Monoxide")
> qqnorm(lengine,main="QQ Plot for the Log of the Carbon Monoxide")
> qqline(lengine)
```

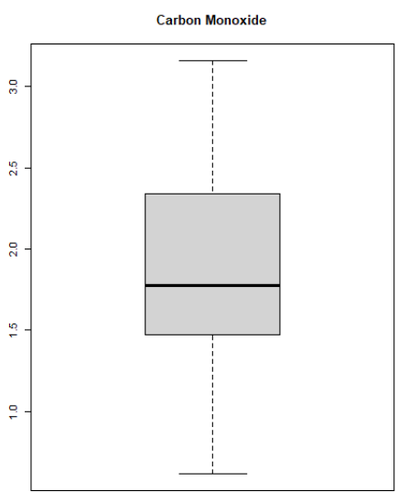


Fig 4 : Boxplot

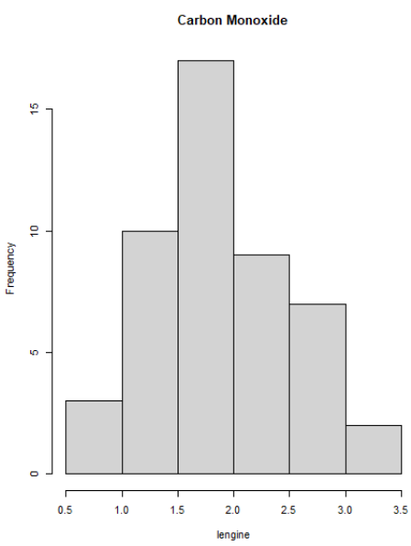


Figure 5 : Histogram

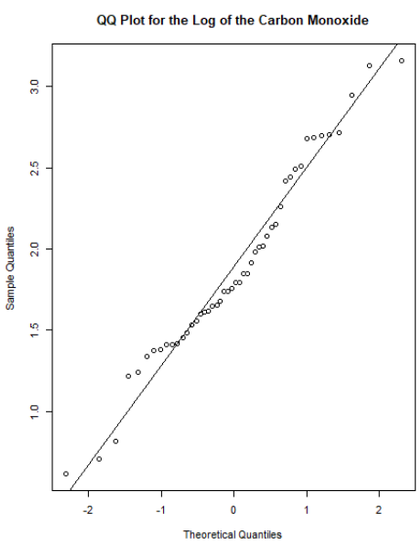


Figure 6 : Normal QQ Plot

3. The Confidence Interval :

Assumption : We now find the confidence interval for the carbon monoxide data. As stated above, we will work with the logarithm of the data because it appears to be closer to a normal distribution. This data is stored in the list called "lengine." Since we do not know the true standard deviation, we will use the sample standard deviation and will use a t-distribution.

We first find the sample mean, the sample standard deviation, and the number of observations:

```
> m = mean(lengine)
> print(m)
[1] 1.883678
> s = sd(lengine)
> print(s)
[1] 0.5983851
> n = length(lengine)
> print(n)
[1] 48
>
> se = s/sqrt(n) # standard error
> print(se)
[1] 0.08636945
>
```

Sub - Conclusion : Finally, the margin of error is found based on a 95% confidence level which can then be used to define the confidence interval:

```
> error = se*qt(0.975,df=n-1)
> print(error)
[1] 0.1737529
>
> left = m-error
> print(left)
[1] 1.709925
> right = m + error
> print(right)
[1] 2.057431
>
> exp(left)
[1] 5.528548
> exp(right)
[1] 7.82584
```

4. Test of Significance :

Assumption : We now perform a test of significance. Here we suppose that ideally the engines should have a mean level of 5.4 and do a two-sided hypothesis test. Here we assume that the true mean is labelled meanx and state the hypothesis test:

Consider Null Hypothesis, **H0 : meanx = 5.4**

and, Alternative Hypothesis, **H1 : meanx (not equal) != 5.4**

To perform the hypothesis test we first assume that the null hypothesis is true and find the confidence interval around the assumed mean. Fortunately, we can use the values from the previous step:

```
> lNull = log(5.4) - error
> print(lNull)
[1] 1.512646
> rNull = log(5.4) + error
> print(rNull)
[1] 1.860152
> print(m)
[1] 1.883678
>
```

Sub - Conclusion : The sample mean lies outside of the assumed confidence interval so we can reject the null hypothesis. There is a low probability that we would have obtained our sample mean if the true mean really were 5.4.

Another way to approach the problem would be to calculate the actual p-value for the sample mean that was found. Since the sample mean is greater than 5.4 it can be like this :

```
> 2*(1-pt((m-log(5.4))/se,df=n-1))
[1] 0.02692539
>
```

Since the p-value is 2.7% which is less than 5% we can reject the null hypothesis.
Note that there is yet another way to do this. The function t.test will do a lot of this work for us.

```
> t.test(lengine,mu = log(5.4),alternative = "two.sided")

        One Sample t-test

data:  lengine
t = 2.2841, df = 47, p-value = 0.02693
alternative hypothesis: true mean is not equal to 1.686399
95 percent confidence interval:
 1.709925 2.057431
sample estimates:
mean of x
 1.883678
```

>

> tLeft = (lNull-log(7))/(s/sqrt(n))

> tRight = (rNull-log(7))/(s/sqrt(n))

> p = pt(tRight,df=n-1) - pt(tLeft,df=n-1)

> print(p)

[1] 0.1629119

> print(1-p)

[1] 0.8370881

5. Conclusion of the Problem : So, the probability of making a type II error is approximately 16.3%, and the probability of detecting a difference if the level really is 7 is approximately 83.7%.

6. References :

i. More information and a more complete list of the options for this command can be found using the help command: **> help()**

ii. This problem comes from the 5th edition of *Moore and McCabe's " Introduction to the Practice of Statistics "* and can be found on *pp. 466-467*. The data consists of the emissions of three different pollutants from 46 different engines. A copy of the data is used here in this experiment. The problem examined here is different from that given in the book but is motivated by the discussion in the book.

iii. The (table.csv) .csv file used in this problem (carbon monoxide data) is uploaded → HERE ← you can download.

iv. R Documentation - Hypothesis Testing.

-------- End of Final Lab Report -------

Submitted on : 8th December 2020 (Tuesday)

Submitted by : Gudi Varaprasad

*Registration No. : **19BCE7048***

Submitted to : Prof. Tanuj Kumar