

**CSE 240 Data Science with R**

**STUDENT WORK BOOK**

|  |  |  |
| --- | --- | --- |
| **Name** | **:** | M SIVANT |
| **Unique ID** | **:** | E00119004 |
| **Year** | **:** | II |
| **Quarter** | **:** | Q6 |
| **Department** | **:** | B.Tech CSE (CyS & IoT or AI &ML) |
| **Faculty Name** | **:** | Prof.B.Nirmala or Prof.N.Chiranjeevi |
| **Academic Year** | **:** | 2020-2021 |

**Date: 02-11-2020**

**Questions:**

1.Consider 2 vectors c(9,10,11,12) and c(13,14,15,16). Create a 4 by 2 matrix from these two vectors

2.Write an R program to take input from the user (user ID and Group/Branch) and display the values

3. Create a data frame Write a R program to create a data frame from four given vectors. a name b. Subject C. Score d. Rank

4.Write a R program to get the statistical summary and nature of the data of a given data frame. (use 3rd Question dataframe)

5. Write a R program to extract specific column from a data frame using column name

6. Write a R program to extract first two rows from a given data frame

**Program:**

# question 1

cat("Creation of Matrix:\n\n")

a <- c(9,10,11,12)

b <- c(13,14,15,16)

matrix\_4\_by\_2 <- matrix(data = c(a,b), nrow = 4, ncol =2)

print(matrix\_4\_by\_2)

# question 2

cat("\nGetting input from the user:\n\n")

id = readline("Enter userID : ")

batch = readline("Enter batch : ")

cat(id, batch,'\n')

# question 3

cat("\nCreating a dataframe:\n\n")

name = c("Siva","King","Star")

subject = c("AI & ML","Python","IOT")

score = c(19,20,18)

rank = c(2,1,3)

df = data.frame(name, subject, score, rank)

print(df)

# question 4

cat("\nDisplaying summary:\n\n")

print(summary(df))

# question 5

cat("\nDisplaying name column of the dataframe:\n\n")

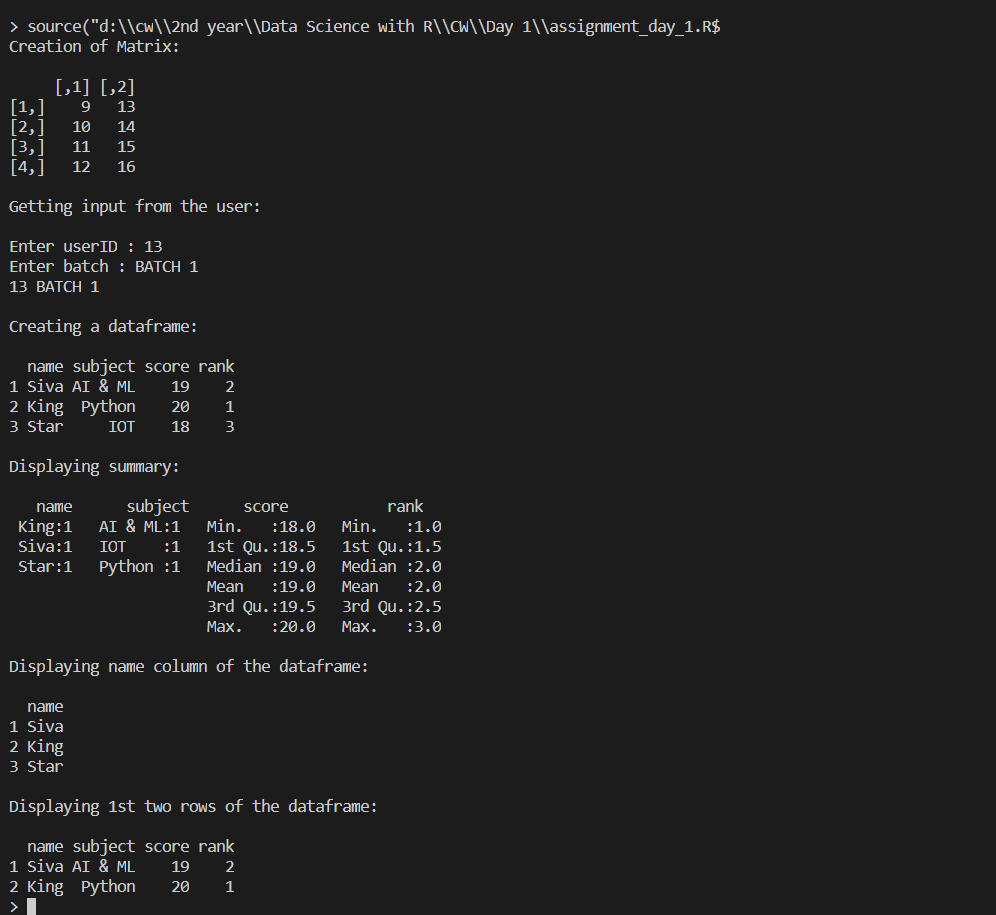
print(df['name'])

# question 6

cat("\nDisplaying 1st two rows of the dataframe:\n\n")

print(df[1:2,])

**Output:**



**Explanation: Concept or Program**

1. Vector:

Vector is a collection of elements

**Syntax** = c (val1, val2, val3, …, valn)

1. Matrix:

Matrices are the R objects in which the elements are arranged in a two-dimensional rectangular layout.

**Syntax** = matrix(data, nrow, ncol, byrow, dimnames)

1. DataFrame:

A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

**Syntax** = data.frame(column1, column2, …, columnn)

1. Summary:

The statistical summary and nature of the data can be obtained by applying summary() function.

**Syntax** = df.summary()

1. as.numeric(val):

as.numeric() is used to convert variable val to numberic type

1. print():

print() is used to display a message or value stored in a variable

1. cat():

cat () is used to concatenate 2 or more messages or values stored in a variable and finally display the values at the console

1. slicing:
   1. df['name']: display specific column in a data frame
   2. df[1:2,]: display first two columns in a data frame

**Date: 03-11-2020**

**Questions:**

1.Create an R script that calculates the square root of a given integer vector x of length one, if the value contained in x is negative it should return NA.

2. Demonstrate and examine the output of letter and LETTER

3.Create an R script that, given a numeric vector x with length 3, will print the elements by order from high to low.

4.Create an R script that returns the amount of values that are larger than the mean of a vector. You are allowed to use mean(). ( Use function)

5.Write a double for loop which prints 30 numbers (1:10, 2:11, 3:12). Those are three clusters of ten numbers each. The first loop determines the number of clusters (3) via its length; the second loop the numbers to be printed (1 to 10 at the beginning). Each cluster starts one number higher than the previous one

6 a. You have the data.frame ‘mydf’ with four columns like below

a = c(3,7,NA, 9)

b = c(2,NA,9,3)

f = c(5,2,5,6)

d = c(NA,3,4,NA)

You want to add another column ‘5’: the 5th column contains the value of col 2 if col 1 is NA; the 5th column contains the value of col 4 if col 2 is NA; the 5th column contains the value of col 3 in all other cases.

7.Write a while loop starting with x = 0. The loop prints all numbers up to 35 but it skips number 7. Condition: If x== 7 next

8. Examine the difference between typeof and class () method using R program

9. Create a function and demonstrate their features like required, keyword, default.

10. Create a dataframe and delete the row and column. (Use the own data values to create frame)

**Program:**

# question 1

cat("Square root of a number:\n\n")

x = as.integer(readline("Enter a number : "))

if(x>=0){

    print(paste("Square root = ",sqrt(x)))

}else {

   print(NA)

}

cat('\n')

# question 2

cat("\nletters vs LETTERS:\n\n")

print(letters) # lower case constant

print(LETTERS) # upper case constant

cat('\n')

# question 3

cat("\nDisplay values max to min:\n\n")

vector = c(13, 1, 6)

print(sort(vector,decreasing = TRUE))

cat('\n')

# question 4

cat("\nDisplay values greater than mean:\n\n")

values = c(1:10)

mean\_value = mean(values)

cat("MEAN : ",mean\_value,'\nVALUES GREATER THAN MEAN : ')

for(i in values)

    if(i>mean\_value)

        cat(i,' ')

cat('\n')

# question 5

cat("\nDisplay sequence of numbers:\n\n")

for(i in seq(1:3)){

    for(j in seq(i,i+9))

        cat(j,' ')

    cat('\n')

}

# question 6

cat("\nAdd a vector to dataframe:\n\n")

df = data.frame(

    a = c(3, 7, NA, 9),

    b = c(2, NA, 9, 3),

    f = c(5, 2, 5, 6),

    d = c(NA, 3, 4, NA)

)

e = c()

print(df)

cat('\n')

for(i in 1:nrow(df))

    if(is.na(df[i,1])){

        e = c(e,df[i,2])

    }else if (is.na(df[i,2])){

        e = c(e,df[i,4])

    }else{

        e = c(e,df[i,3])

    }

df = cbind(df,e)

print(df)

cat('\n')

# question 7

cat("\nDisplay numbers from 0 to 35 except 7:\n\n")

for(i in seq(0,35))

    if (i==7) {

       next

    }else {

        cat(i,' ')

    }

cat('\n')

# question 8

cat("\nTypeof vs class:\n\n")

a = 1L; b = 'a'; c = pi; d = c(1,2,3)

print(typeof(a))

print(typeof(b))

print(typeof(c))

print(typeof(d))

print(class(a))

print(class(b))

print(class(c))

print(class(d))

# question 9

cat("\nRequired, Keyword and Default:\n\n")

# required

square = function(num){

    print(seq(1,num)^2)

}

square(5)

cat('\n')

# keyword

even\_or\_odd = function(val){

    if(val%%2==0)

        return ('EVEN')

    else

        return ('ODD')

}

print(even\_or\_odd(val=13))

cat('\n')

# default

hello <- function(name = 'buddy',age=18){

    print(paste("Hello",name,",you are",age,"years old!"))

}

hello()

hello("siva",13)

cat('\n')

# question 10

cat("\nDelete row and column:\n\n")

print(df)

cat('\n')

# delete row

print(df[-c(1,4),])

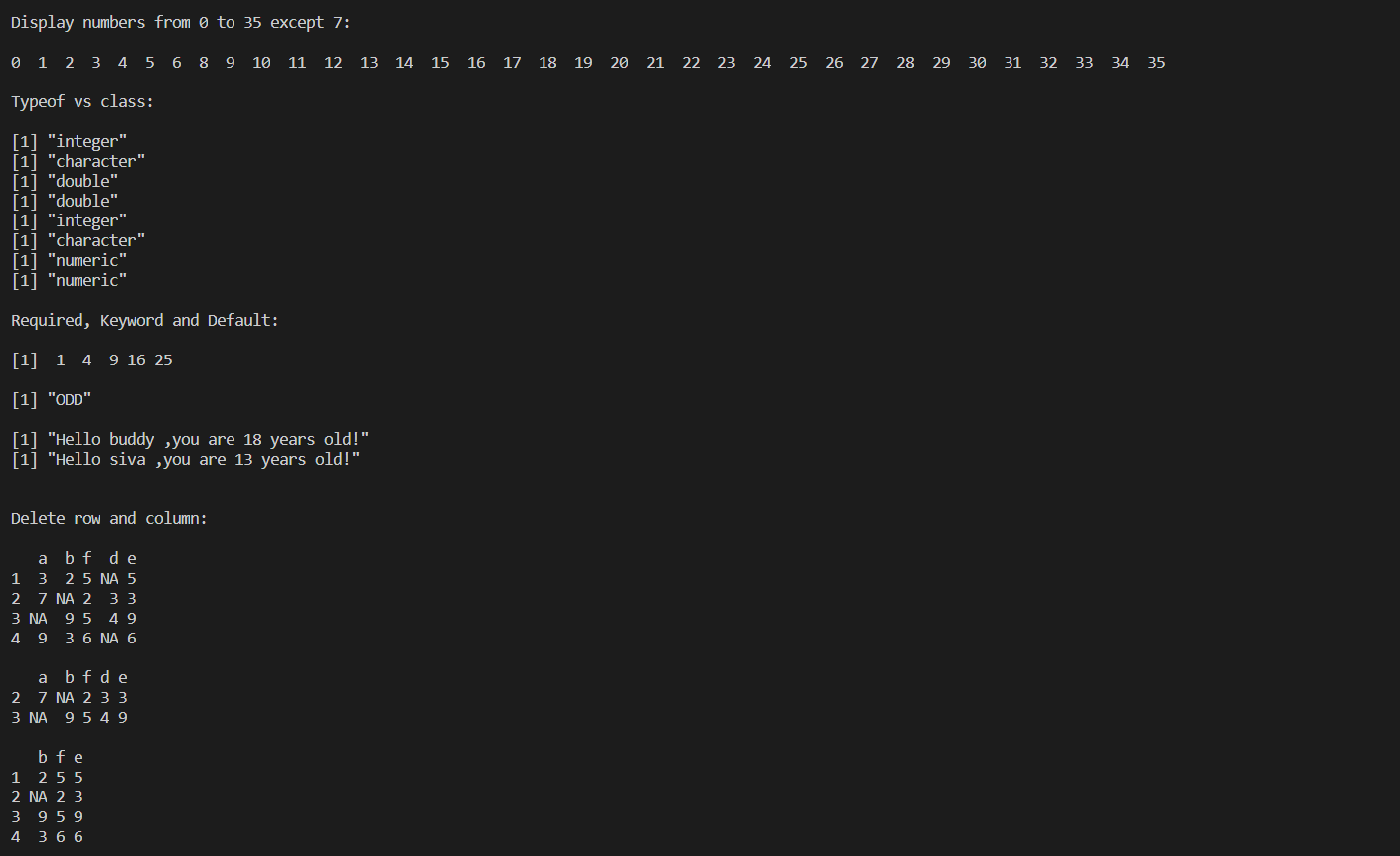
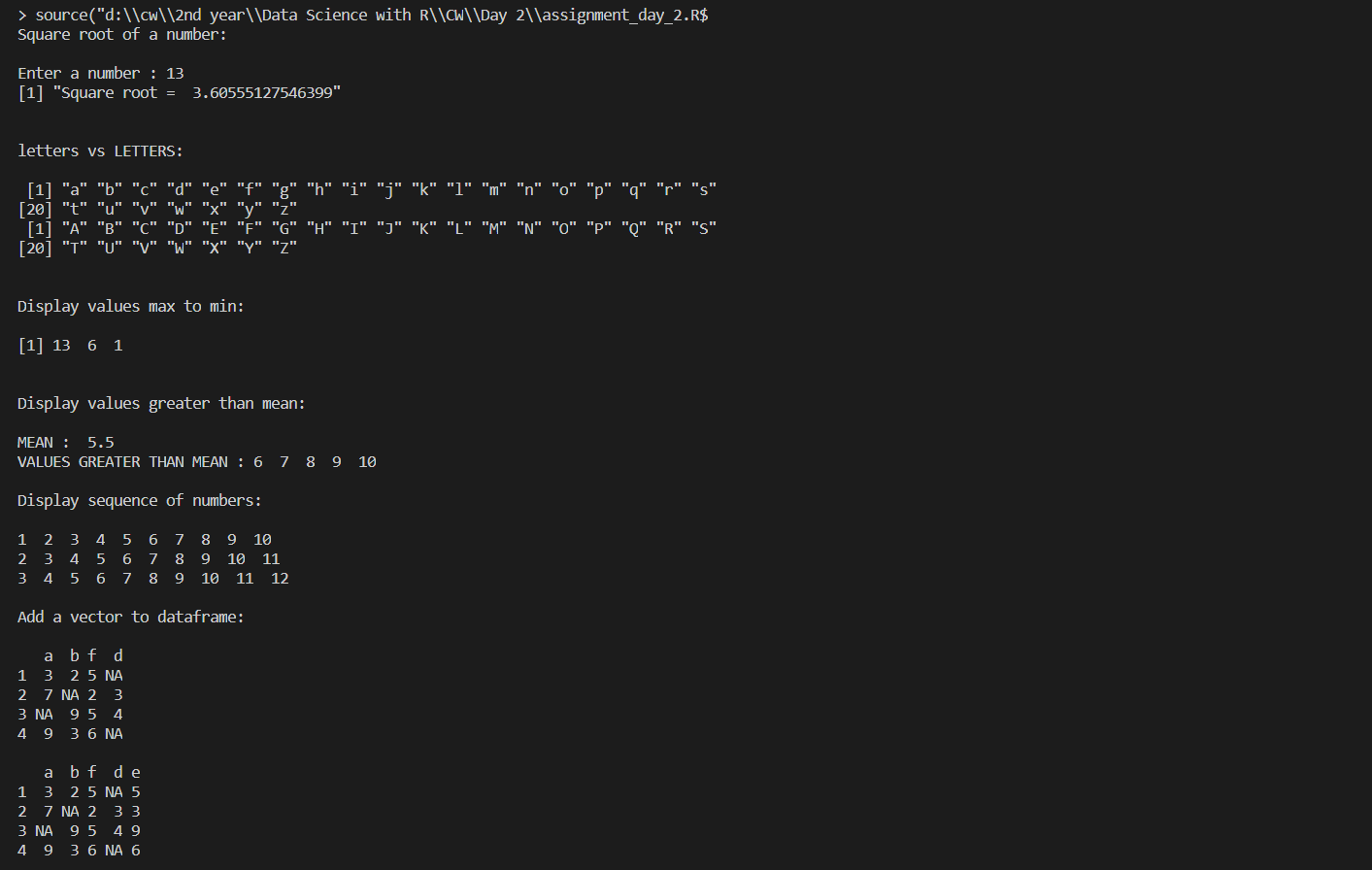
cat('\n')

# delete column

print(subset(df, select = -c(a,d)))

cat('\n')

**Output:**



**Explanation:**

1. sqrt():

sqrt() is an inbuilt function which returns square root of a number

1. letters and LETTERS:

letters and LETTERS are constants built in R

* 1. LETTERS: the 26 upper-case letters of the Roman alphabet
  2. letters: the 26 lower-case letters of the Roman alphabet

1. sort():

Sort (or order) a vector or factor (partially) into ascending or descending order.

1. mean():

Find mean for an array of elements

1. is.na():

returns true if the value of an element is NA

1. while():

It tests the condition before executing the loop body.

**Syntax:** while(condition) {

statements

}

1. for():

it tests the condition at the end of the loop body.

**Syntax:** for (value in vector) {

statements

}

1. if-else statements:

An if statement can be followed by an optional else statement which executes when the Boolean expression is false.

**Syntax:** if(Boolean expression) {

// statement(s) will execute if the Boolean expression is true.

} else {

// statement(s) will execute if the Boolean expression is false.

}

1. next:

next is used to continue the existing loop

1. class():

The function class prints the vector of names of classes an object inherits from.

1. typeof():

typeof() determines the (R internal) type or storage mode of any object.

1. function():

A function is a set of statements organized together to perform a specific task.

**Syntax:** function\_name <- function(arg\_1, arg\_2, ...) {

Function body

}

1. cbind():

cbind() function combines vector, matrix or data frame by columns. The row number of the two datasets must be equal. If two vectors do not have the same length, the elements of the short one will be repeated.

**Syntax:** cbind(x1, x2)

x1, x2 can be vector, matrix or data frame

1. subset():

Subset Function in R, returns subset of data frame, vectors or matrices which meet the specified conditions.

**Syntax:** subset(x, condition, select)

* x – can be a matrix ,data frame or vector
* condition- condition to be satisfied
* select – columns to be selected

1. paste():

Paste function in R is used to concatenate Vectors by converting them into character.

**Syntax:** paste (arg1, arg2, sep = “”, collapse = NULL)

1. seq():

seq() function in R generates a sequence of numbers

**Syntax:** seq(from, to, by, length.out)

* From beginning of the sequence
* To end of the sequence
* By increment by (default is 1)
* length.out length of the sequence

**Date: 04-11-2020**

**Questions:**

1.Write a function that turns (e.g.) a vector c("a", "b", "c") into the string "a, b, and c". Think carefully about what it should do if given a vector of length 0, 1, or 2.

2. Consider a data frame df:

Id=c(1:10)

Age=c(14,12,15,10,23,21,41,56,78,12)

Sex=c('F','M','M','F','M','F','M','M','F','M')

Code=letters[1:10]

df=data.frame(Id,Age,Sex,Code)

Create a function that, given a data frame and two indexes, exchanges two values ​​of the Code variable with each other.

For example, if the index is 1 and 3, you assign:

df[1,'Code']=df[3,'Code']

df[3,'Code']=df[1,'Code']

3.Create a function that given a numeric vector, sort this in ascending order and duplicate it by two.

4.Create a function that given a numeric vector X returns the digits 0 to 9 that are not in X. If X=0 2 4 8 the function return 1 3 5 6 7 9

5.Create a function that given one word, return the position of word’s letters on letters vector. For example, if the word is ‘abc’, the function will return 1 2 3.

6. Write a code to check the given string is anagram or not

**Program:**

# question 1

cat('\nVector to String\n\n')

to\_string = function(vector){

    if (length(vector)==0){

      return("NA")

    }

    else if(length(vector)<2){

      return(as.character(vector))

    }else{

        value = paste(vector[1:length(vector)-1],collapse=', ')

        return(paste(value,'and',vector[length(vector)]))

    }

}

print(to\_string(c()))

print(to\_string(c(1)))

print(to\_string(c(1,2)))

print(to\_string(c(1,2,3,4)))

cat('\n')

# question 2

cat('\nSwap Code\n\n')

swap\_code = function(df,a,b){

  if(a<=nrow(df) && b<=nrow(df)){

    temp = df[a,'Code'];

    df[a,'Code']=df[b,'Code']

    df[b,'Code']=temp

    print(df)

  }

}

Id=c(1:10)

Age=c(14,12,15,10,23,21,41,56,78,12)

Sex=c('F','M','M','F','M','F','M','M','F','M')

Code=letters[1:10]

df=data.frame(Id,Age,Sex,Code)

print(df)

cat('\n')

swap\_code(df,as.integer(readline("Enter index 1 : ")),as.integer(readline("Enter index 2 : ")))

cat('\n')

# question 3

cat('\nSort and Duplicate\n\n')

sort\_duplicate = function(vector) {

  val = sort(vector)

  return(c(val,val))

}

print(sort\_duplicate(c(1,3,6,4)))

cat('\n')

# question 4

cat('\nNot in 1 to 10\n\n')

not\_in\_1\_to\_10 = function(vector){

  test=0:9

  for(i in test) {

    val = as.integer(grep(i,vector))

    if(length(val)==0){

      print(i)

    }

  }

}

not\_in\_1\_to\_10(c(1,3,6,4))

cat('\n')

# question 5

cat('\nWord to Number\n\n')

word\_to\_number = function(string){

    string\_list = strsplit(string,"")

    num\_vectors = c()

    for(i in string\_list)

        num\_vectors = c(num\_vectors,match(i, letters[1:26]))

    return(num\_vectors)

}

string = "word"

print(word\_to\_number(string))

cat('\n')

# question 6

cat('\nAnagram\n\n')

anagram = function(str1,str2){

    str1\_number = sort(word\_to\_number(str1))

    str2\_number = sort(word\_to\_number(str2))

    print(str1\_number)

    print(str2\_number)

    if(all.equal(str1\_number,str2\_number)==TRUE)

        print("It is a anagram")

    else

        print("It is not a anagram")

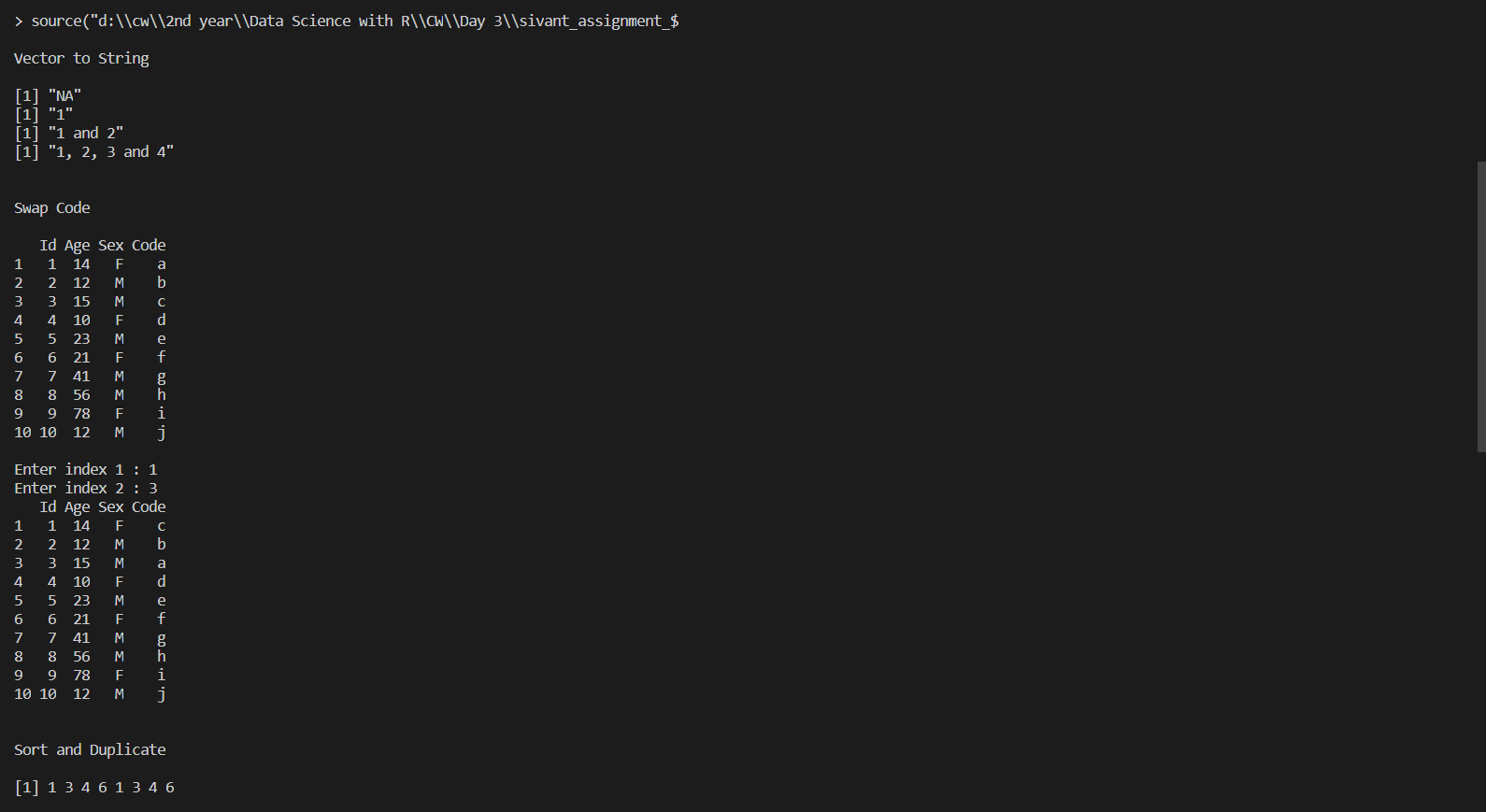
    cat('\n')

}

anagram('and','dan')

anagram('and','adan')

**Output:**



**Explanation:**

1. length():

length() is an inbuilt function which returns the length of vectors

1. grep():

grep() is an inbuilt function which returns the index of matched sequence

1. strsplit():

strsplit function in R is used to split the string into substrings with the specified delimiter.

**Syntax:** strsplit(string, delimiter)

1. match():

Match() Function in R , returns the position of match i.e. first occurrence of elements of Vector 1 in Vector 2. If an element of vector 1 doesn’t match any element of vector 2 then it returns “NA”.

**Syntax:** match(v1, v2)

1. all.equal():

Return true if all the values of vector1 (v1) and vector2 (v2) are equal

**Syntax:** all.equal(v1, v2)

**Date: 05-11-2020**

**Questions:**

1. List all example files available with the readr library.
2. Read the mtcars.csv file.
3. Read the first 10 lines from the mtcars.csv file.

**Program:**

# question 1

library(readr)

print(dir(system.file("extdata", package = "readr")))

cat('\n')

# question 2

df <- read\_csv(readr\_example('mtcars.csv'))

print(dim(df))

cat('\n')

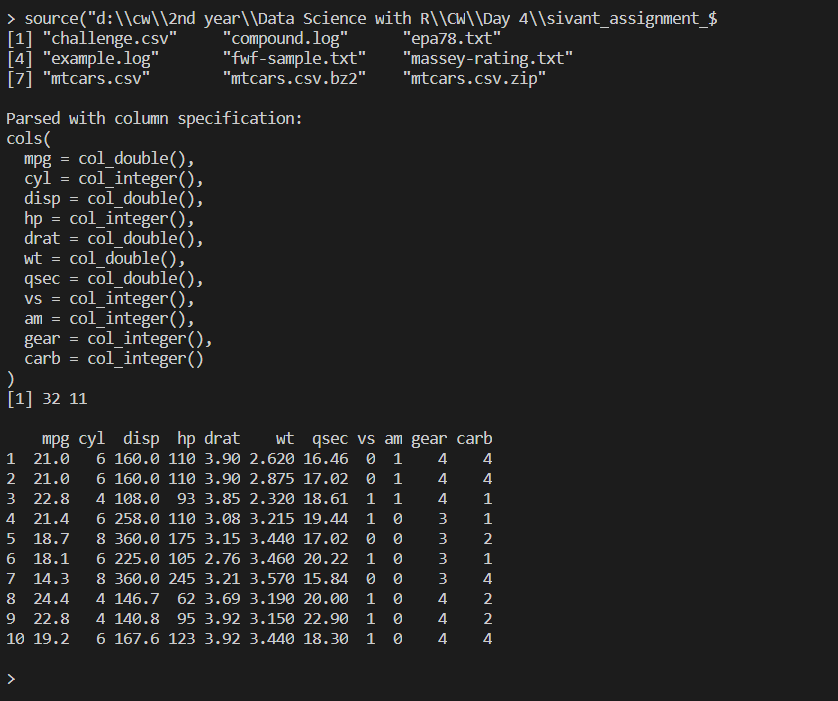
# question 3

df <- read.csv(readr\_example('mtcars.csv'),nrows=10)

print(df)

cat('\n')

**Output:**



**Explanation:**

1. read.csv():

read.csv() is an inbuilt r package use to read data from csv files

1. readr\_example():

readr\_example is use to load example files in readr package

**Date: 06-11-2020**

**Questions:**

Perform the following visualization on movies dataset

1. Bar plot

2. Grouped bar plot

3. Line chart

4. Scatter plot

5. Pie chart and 3D pie chart

6. Histogram

7. Compare and analysis the difference plot

8. Demonstrate the ggplot2 layer

9. Draw the histogram using ggplot2

10. Demonstrate wordcloud

**Program:**

library(ggplot2)

library(ggplot2movies)

library(wordcloud)

library(plotrix)

# bar plot

count=head(movies$rating,40)

count = table(count)

barplot(count,main="Bar Plot",xlab = "Rating",ylab= "Frequency",col=c("pink"))

# double bar plot

count = table(head(movies$r9,100),head(movies$r10,100))

barplot(count,main="Double Bar Plot",beside = TRUE,xlab = "Rating",ylab= "Frequency",col=c("pink","light yellow"))

# pie chart

slices = movies$length[130:133]

pct = round((slices/sum(slices))\*100)

labs = movies$title[130:133]

lbl = paste(labs,pct,"%",sep=" ")

pie(slices,lbl,main="Pie Chart",col = c("grey","pink","light blue","light yellow"))

# 3D Pie Chart

pie3D(slices,labels = paste(pct,'%'),main="Pie Chart 3D",col = c("grey","pink","light blue","light yellow"))

# bar plot

hist(movies$year,col = "pink", main="Movies Frequency", xlab="Year")

# density plot

data=density(movies$year)

plot(data,main="Movies Frequency", xlab="Year")

polygon(data,col='light yellow',border="grey")

# line plot

rating = head(movies$rating,13)

year = head(movies$year,13)

plot(y=rating,x=year,type="b",main="Line Graph")

# scatter plot

rating = head(movies$rating,30)

year = head(movies$year,30)

plot(y=rating,x=year,main="Scatter Plot", col = rep(1:2, each = 20), pch = 19)

legend("bottomright", legend = paste("Group", 1:2), col = 1:2, pch = 19, bty = "n")

# wordcloud

wordcloud(words = movies$title,freq = movies$length, min.freq = 2,max.words = 35,random.order = FALSE)

# scatter plot using ggplot

pl<-ggplot(data=head(movies,100),aes(x=rating,y=year))

pl<-pl+geom\_point()

print(pl)

# histogram

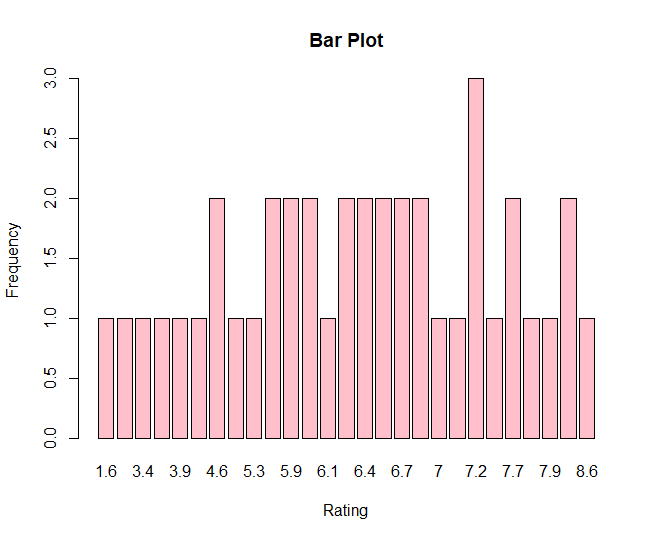
mov<-ggplot(data=movies,aes(x=rating))

mov<-mov+geom\_histogram(bins = 20,color = "black", fill="light yellow",alpha=1)+xlab("Movie Rating")+ylab("Movie Count")+ggtitle("Movies Plot")

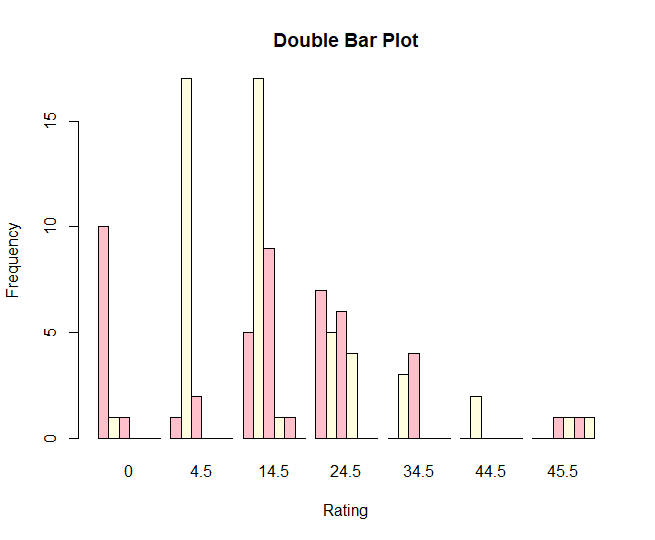
print(mov)

**Output:**

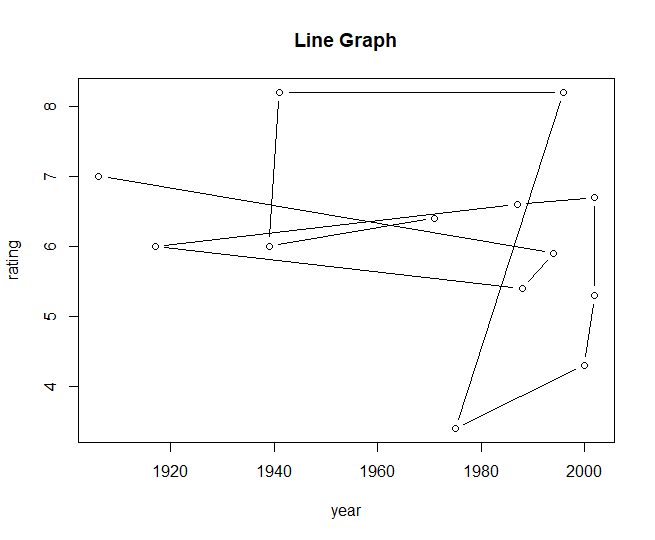
**Question 1:**



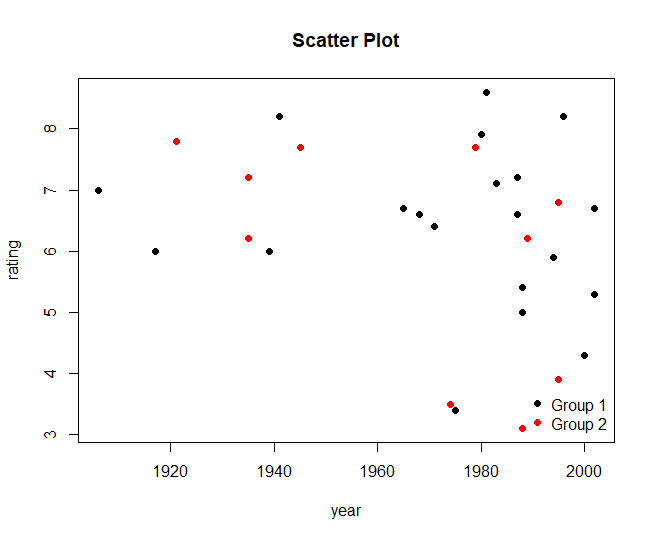
**Question 2:**



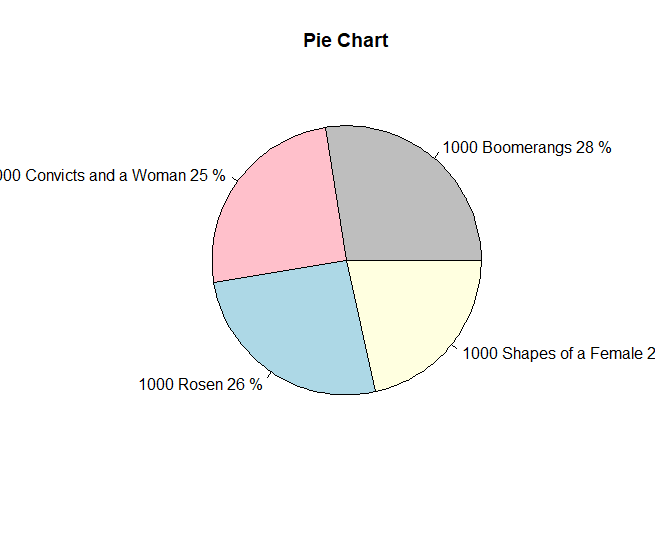
**Question 3:**

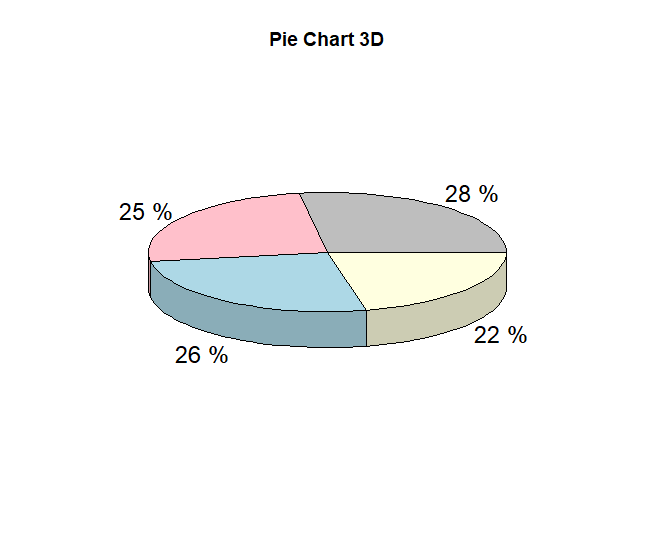


**Question 4:**

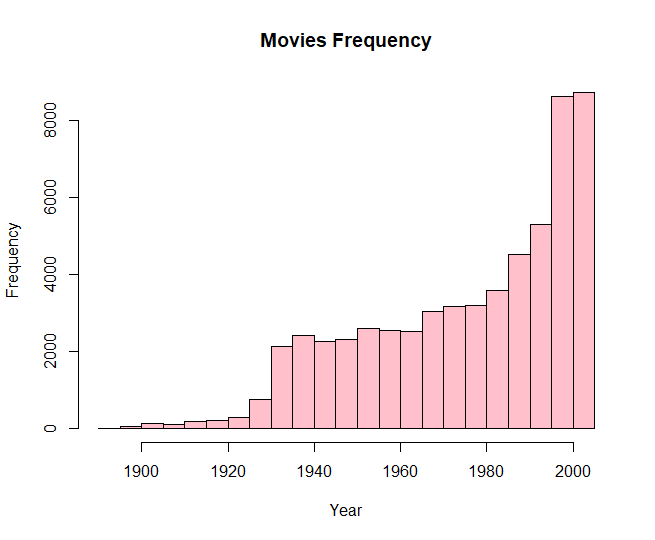


**Question 5:**

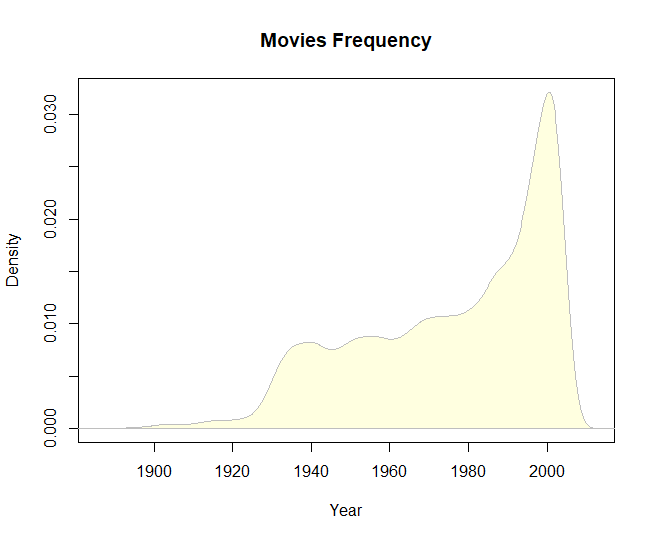




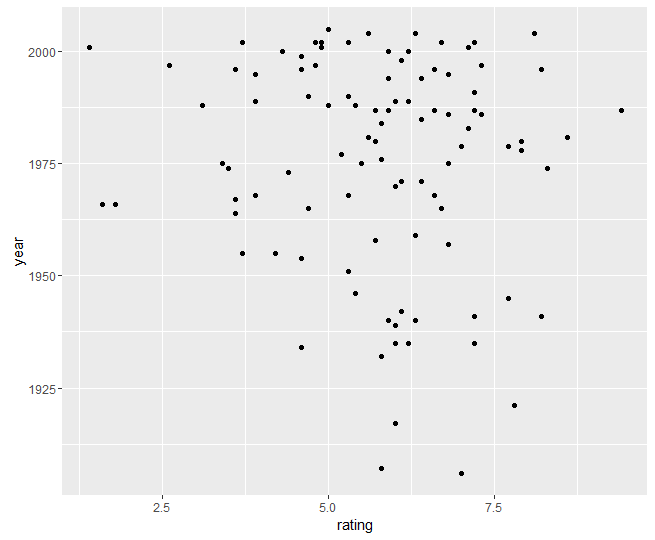
**Question 6:**



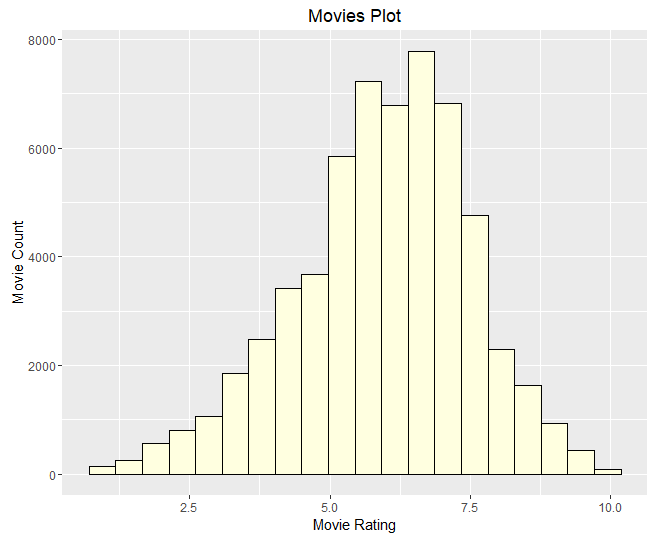
**Question 7:**



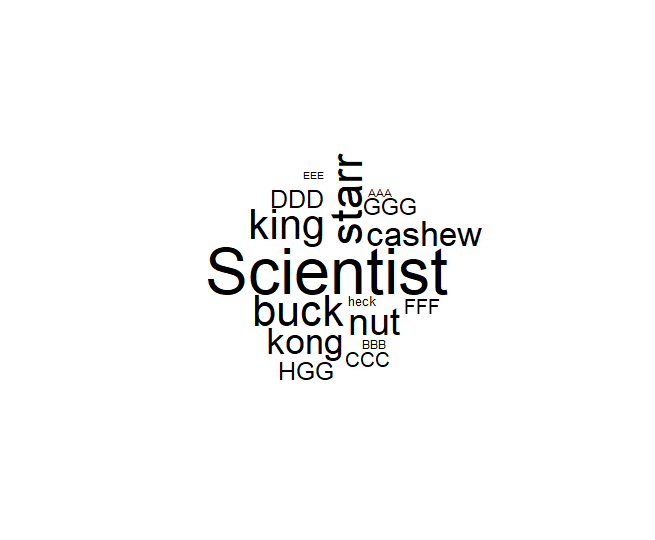
**Question 8:**



**Question 9:**



**Question 10:**



**Explanation:**

1. plot():

The plot function in R has a type argument that controls the type of plot that gets drawn.

The Help page for plot() has a list of all the different types that you can use with the type argument:

* “p”: Points
* “l”: Lines
* “b”: Both
* “c”: The lines part alone of “b”
* “o”: Both “overplotted”
* “h”: Histogram like (or high-density) vertical lines
* “n”: No plotting

1. Histogram:

A histogram plots the frequency of observations.

1. Box Plot:

A boxplot provides a graphical view of the median, quartiles, maximum, and minimum of a data set.

1. Pie Chart:

Pie charts are created with the function pie(x, labels=) where x is a non-negative numeric vector indicating the area of each slice and labels= notes a character vector of names for the slices.

1. Word cloud:

A word cloud (or tag cloud) is a text mining method to find the most frequently used words in a text

**Date: 07-11-2020**

**Questions:**

Perform the following operation:

1. T-Test ( score 1 and score 2) Manual calculation for the T-value Excel sheet R- Program

2. ANOVA Excel sheet R Program

3. Draw the Violin plot

4. Draw the Qplot for 5 Types

5. Demonstrate the skewness and kurtosis using built-in dataset or your own dataset to get the data summary.

**Program:**

# question 1

# T Test

a=c(3,3,3,12,15,16,17,19,23,24,32)

b=c(20,13,13,20,29,32,23,20,25,15,30)

print(t.test(a,b))

cat('\n')

# question 2

# Anova

d1 = c(0,2,3,5,8,10,12)

d2 = c(1,2,3,9,10,10,11)

d3 = c(1,4,5,5,8,9,10)

df = data.frame(d1,d2,d3)

stack\_group=stack(df)

a\_r = aov(values~ind,data=stack\_group)

print(a\_r)

cat('\n')

# question 3

library(ggplot2)

p <- ggplot(head(iris,130), aes(x=Petal.Length, y=Sepal.Width,fill=factor(Species))) +

  geom\_violin()

print(p)

# print(nrow(iris))

# question 4

# print(colnames(iris))

Species=factor(head(iris$Species,200))

Width=head(iris$Sepal.Width,200)

Length=head(iris$Sepal.Length,200)

# histogram

print(qplot(Width,geom="histogram", fill=Species))

# density

print(qplot(Width,geom="density", fill=Species))

# boxplot

print(qplot(Width,Length,geom="boxplot", fill=Species))

# dotplot

print(qplot(Width,Length,geom="dotplot", fill=Species))

# bar

print(qplot(Length,geom="bar", fill=Species))

# question 5

library(e1071)

library(ggplot2movies)

print(colnames(movies))

cat('\n')

# Right Skew

duration = head(movies$r1,20)

print(skewness(duration))

print(kurtosis(duration))

plot(density(duration),main="Right Skew")

cat('\n')

# Left Skew

duration = head(movies$r3,10)

print(skewness(duration))

print(kurtosis(duration))

plot(density(duration),main="Left Skew")

cat('\n')

# Normal Distribution

data=read.csv('D:\\cw\\2nd year\\Data Science with R\\CW\\Day 6\\income.csv')

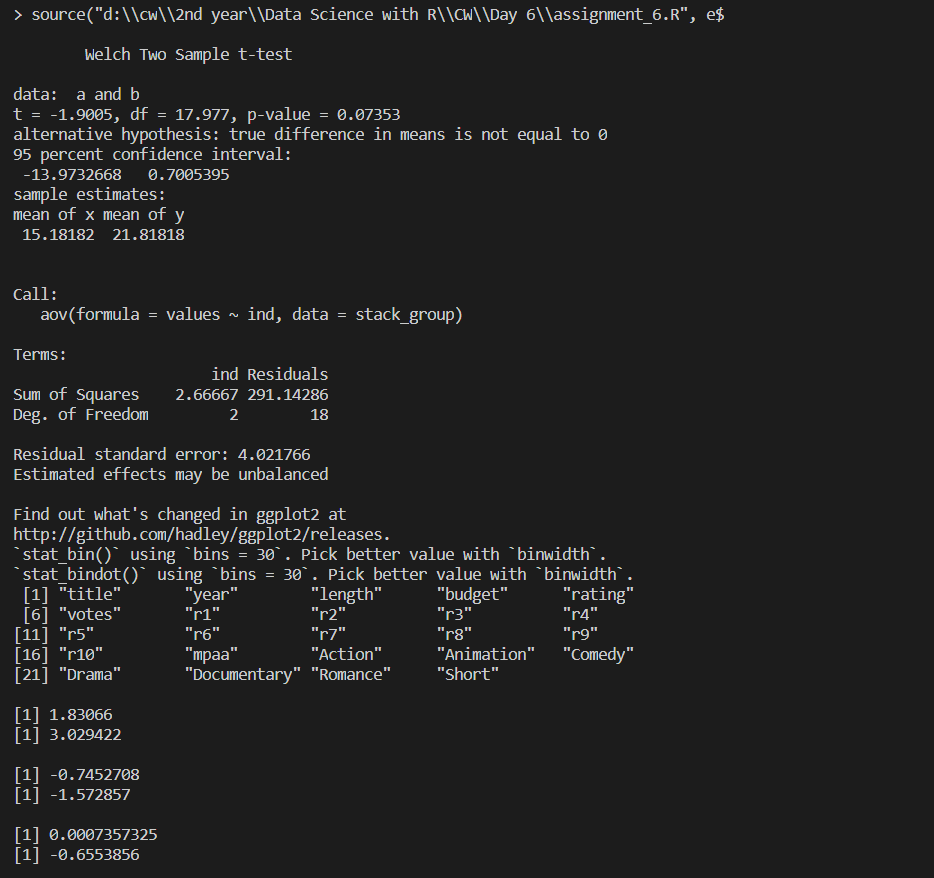
print(skewness(data$amount))

plot(density(data$amount), main="Normal Distribution")

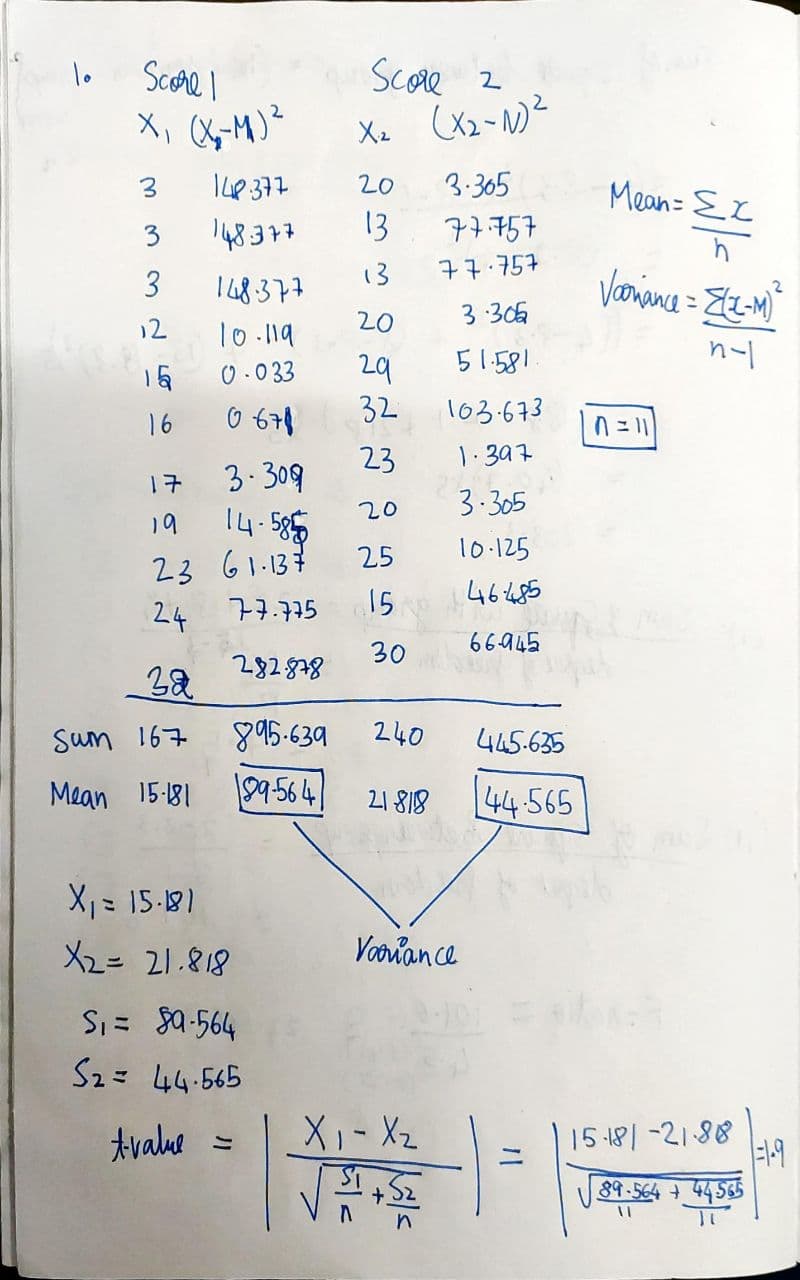
print(kurtosis(data$amount))

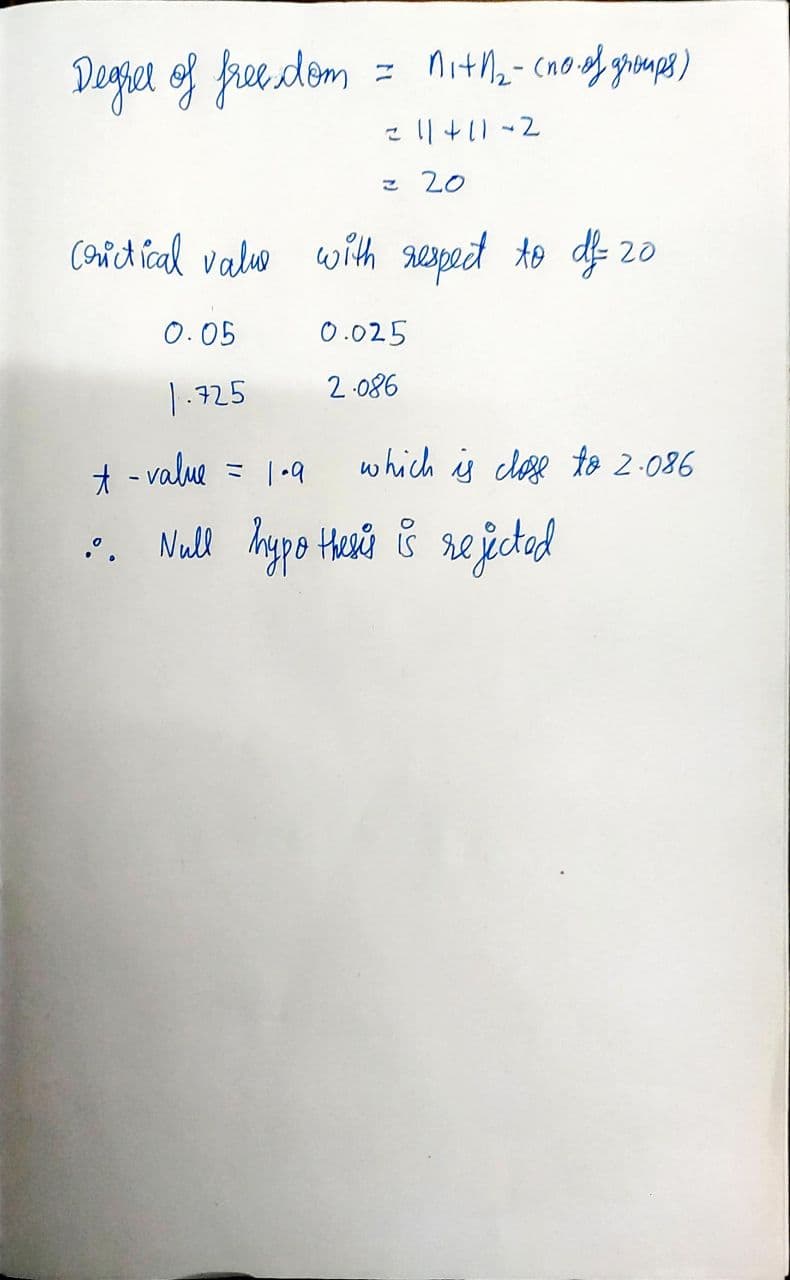
cat('\n')

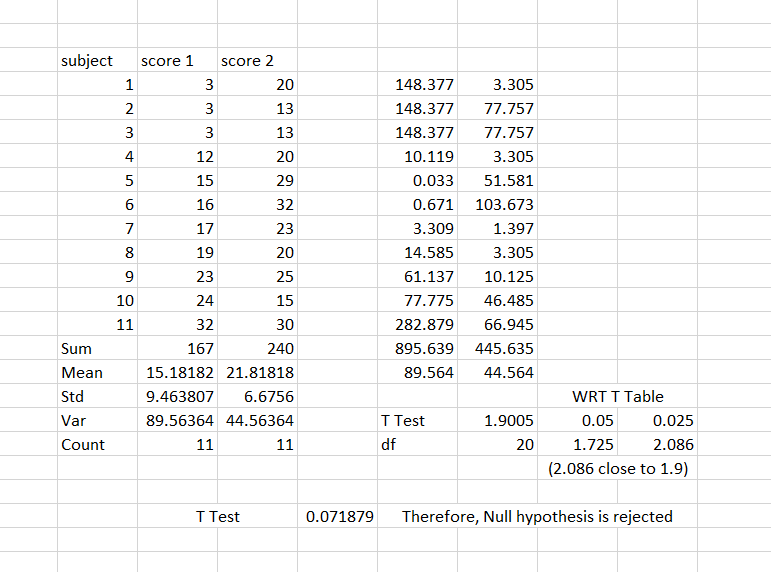
**Output:**



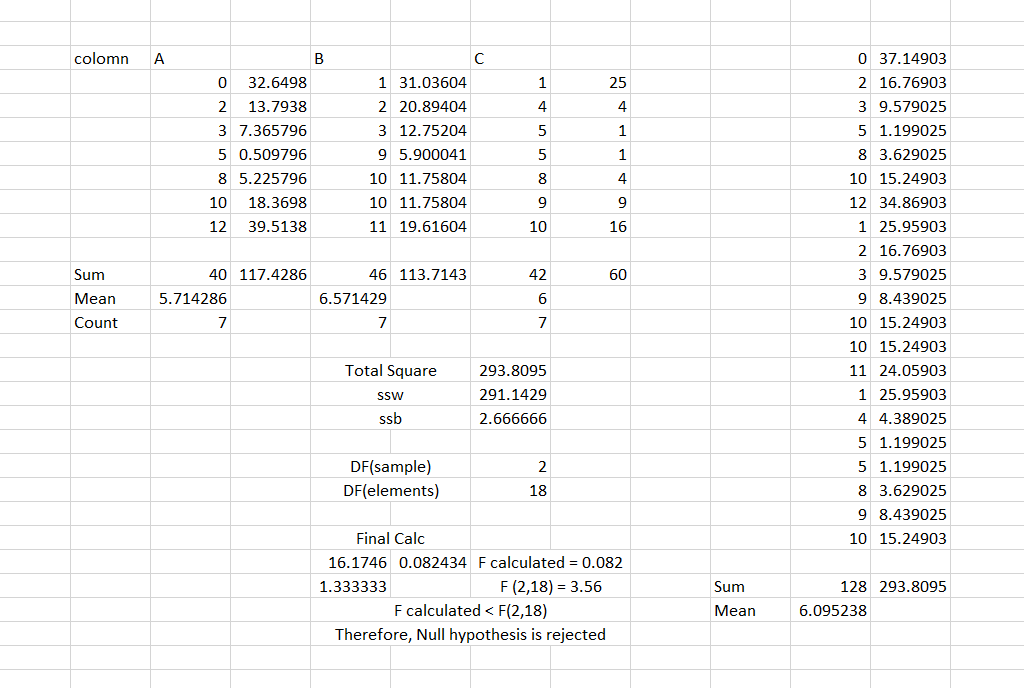
1. T Test:



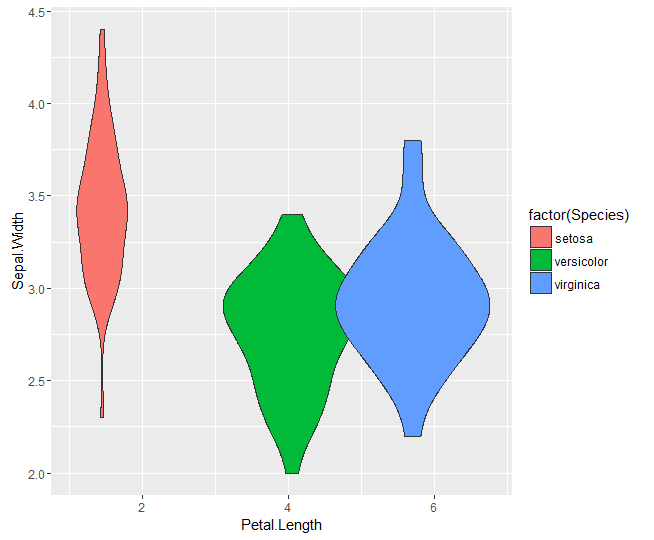




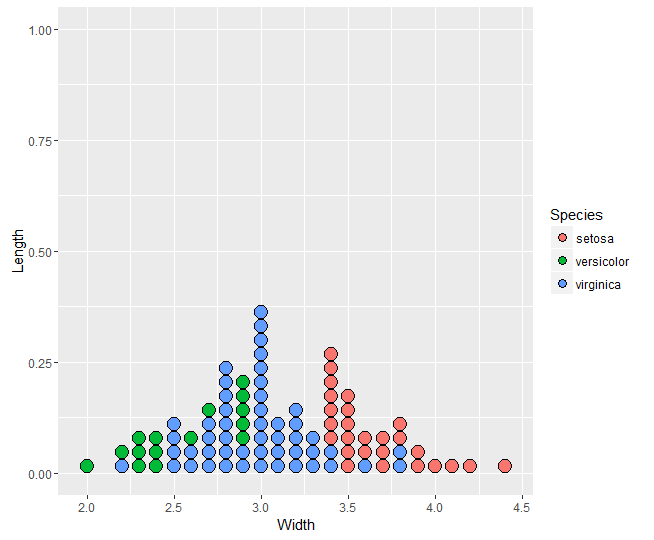
1. Anova



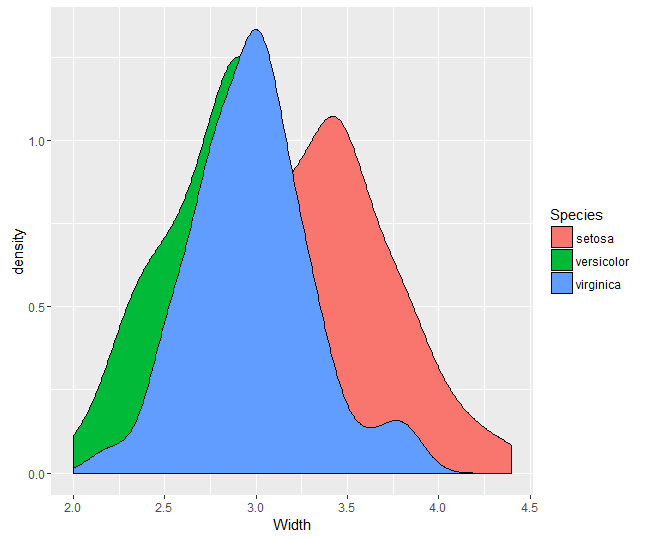
1. Violin Plot:



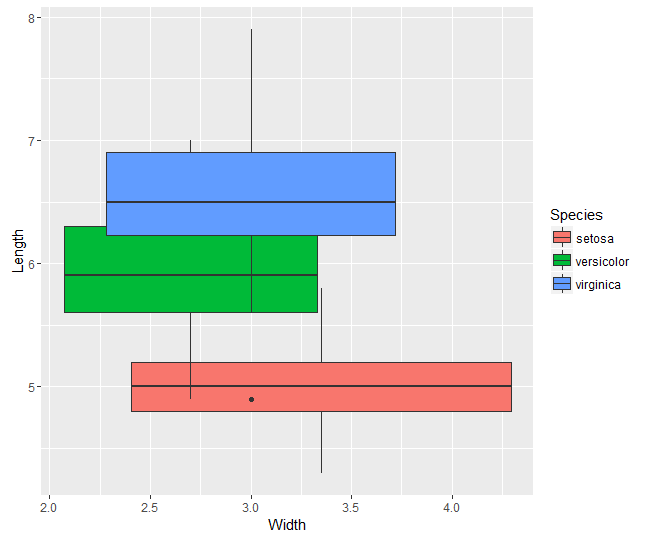
1. Q plot:
   1. Dot Plot



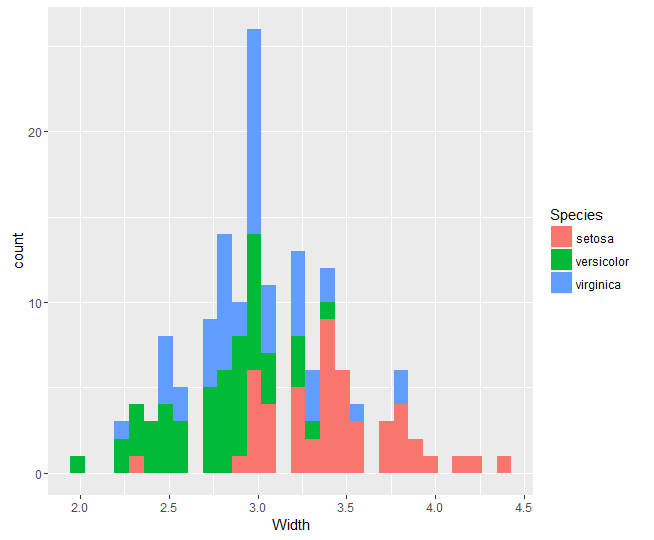
* 1. Density plot:



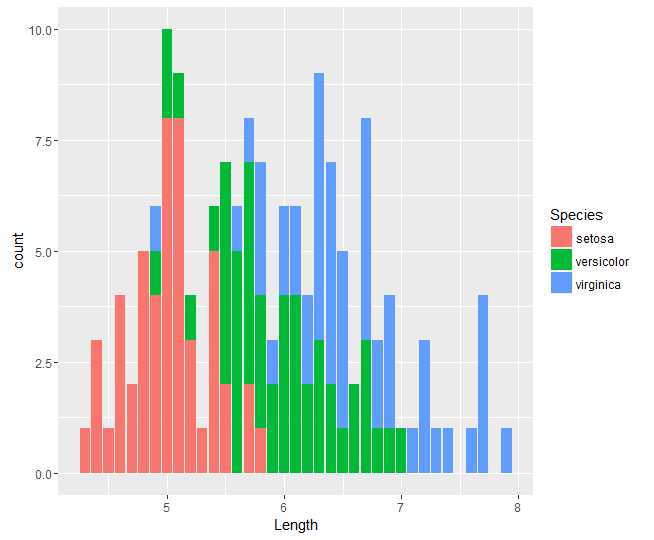
* 1. Bar plot:

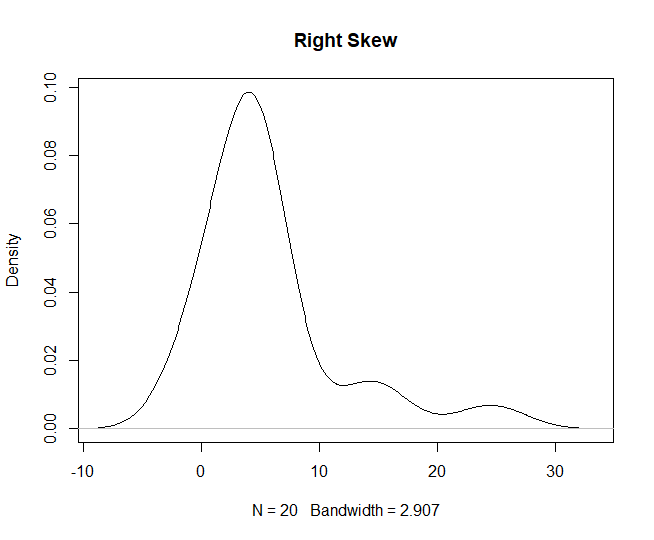
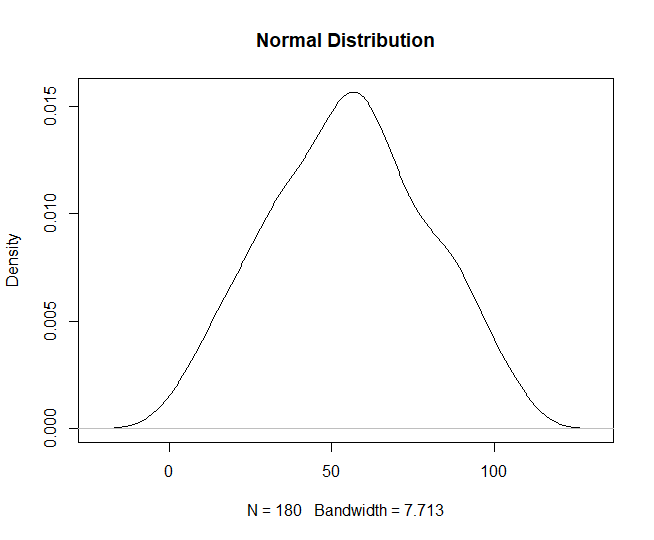


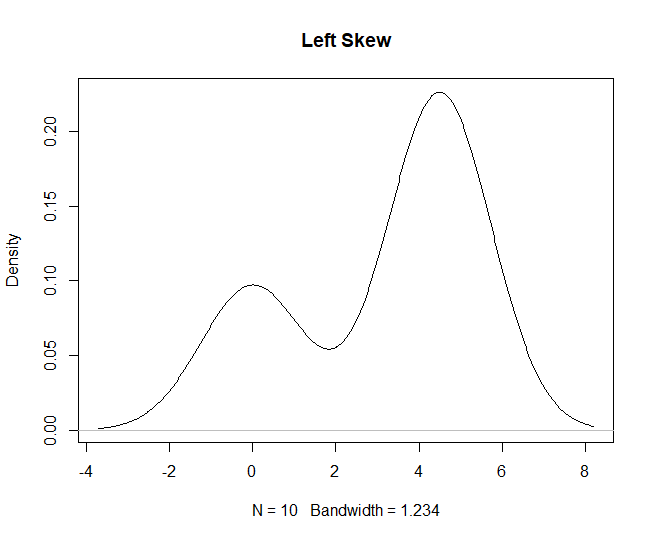
* 1. Density plot:



* 1. Histogram:



1. Skewness and Kurtosis:



**Explanation:**

1. T Test:

A t-test is commonly used to determine whether the mean of a population significantly differs from a specific value (called the hypothesized mean) or from the mean of another population.

1. P Value:

The p-value or probability value is the probability of obtaining test results at least as extreme as the results actually observed during the test, assuming that the null hypothesis is correct.

1. Critical Value:

The critical values of a statistical test are the boundaries of the acceptance region of the test.

The p-value is the variable that allows us to reject the null hypothesis (H₀: µ₁=µ₂) or, in other words, to establish that the two groups are different. However, since the p-value is just a value, we need to compare it with the critical value (⍺):

p\_value > ⍺ (Critical value): Fail to reject the null hypothesis of the statistical test.

p\_value ≤ ⍺ (Critical value): Reject the null hypothesis of the statistical test.

The critical value that most statisticians choose is ⍺ = 0.05. This 0.05 means that, if we run the experiment 100 times, 95% of the times we will be able to reject the null hypothesis and 5% we will not.

1. ANOVA:

An ANOVA test is a way to find out if survey or experiment results are significant. In other words, they help you to figure out if you need to reject the null hypothesis or accept the alternate hypothesis.

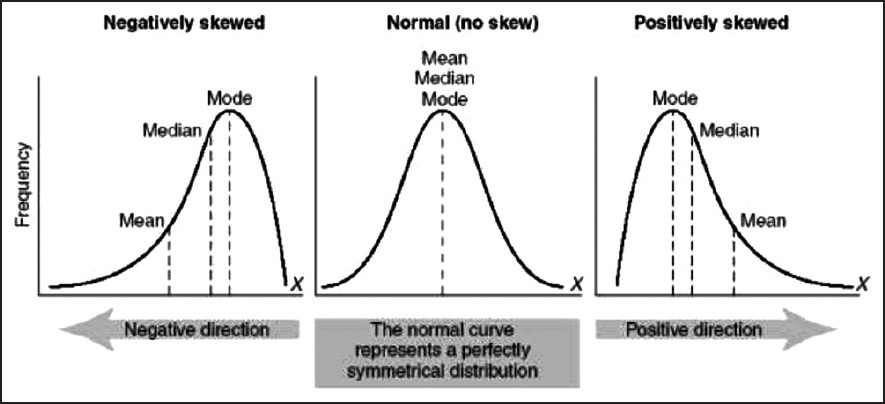
1. ANOVA vs T Test:

A Student’s t-test will tell you if there is a significant variation between groups. A t-test compares means, while the ANOVA compares variances between populations.

You could technically perform a series of t-tests on your data. However, as the groups grow in number, you may end up with a lot of pair comparisons that you need to run. ANOVA will give you a single number (the f-statistic) and one p-value to help you support or reject the null hypothesis.

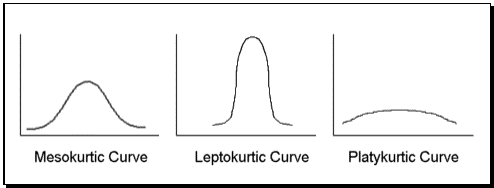
1. Skewness:

Skewness refers to distortion or asymmetry in a symmetrical bell curve, or normal distribution, in a set of data. If the curve is shifted to the left or to the right, it is said to be skewed. Negatively-skewed distributions are also known as left-skewed distributions.



1. Kurtosis:

Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. That is, data sets with high kurtosis tend to have heavy tails, or outliers. Data sets with low kurtosis tend to have light tails, or lack of outliers. A uniform distribution would be the extreme case.



**Date: 10-11-2020**

**Questions:**

1. The coin is flipped ten times. Find the probability of 7 heads occurring.
2. A card is selected three times (and replaced). Find the probability of 2 face cards occurring.
3. 3. A student decides to guess on a section of his ACT test. The section contains 50 multiple choice questions and each question has 5 possible answers.
   1. Find the expected number of correct responses.
4. A company ships 5000 cell phones. They are expected to last an average of 10,000 hours before needing repair; with a standard deviation of 500 hours. Assume the survival time of the phones are normally distributed. If a phone is randomly selected to be tracked for repairs find the expected number that needs repair,
   1. after 11,000 hours

**Program:**

# question 1

print(dbinom(7,size=10,prob=0.5)\*100)

# question 2

print(dbinom(2,size=3,prob=(3/13))\*100)

# val = 3\*((3/13)^2)\*((10/13)^1) # using formula

# question 3

n = 50  # (no of questions)

p = 0.2 # probability for answer is right

# number of correct responses

print(n\*p)

# question 4

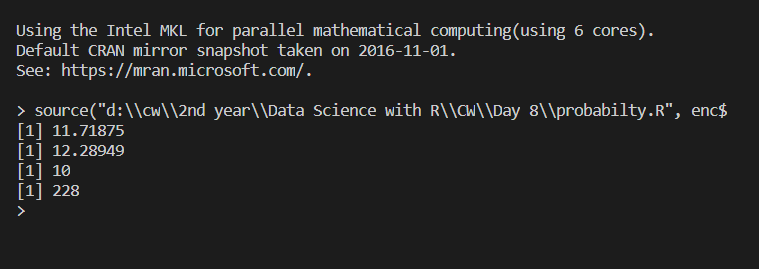
# number of phones that need repairs

p\_pval = pnorm(11000,mean=10000,sd=500,lower.tail=FALSE)

num\_phones = round(p\_pval \* 10000)

print(num\_phones)

**Output:**



**Explanation:**

1. Binomial distribution:

The binomial distribution is a discrete probability distribution.

      (    )
        n    x      (n− x)
f(x) =  x   p (1− p)      where x = 0,1,2,...,n


**Syntax:**

Discrete : dbinom(no of occurance event ,no. of trails, probility of the event)

Continuous : pbinom(no of occurance event ,no. of trails, probility of the event)

**Note:**

pbinorm gives summation of probability till n events where as dbinom gives the probability of nth event

1. Normal distribution:

The normal distribution is defined by the following probability density function, where μ is the population mean and σ2 is the variance.

**Syntax:** pnorm(value, mean, sd, lower.tail=FALSE)

**Date: 13-11-2020**

**Day - 7**

**Questions:**

1. Create a variable called my\_pattern and implement the required pattern for finding one digit and one uppercase alphanumeric character, in variable text1. This time, combine predefined classes in the regex pattern. Use function grepl to verify if the searched pattern exists on the string.

2.Using the sub function, replace the pattern found on the previous exercice by the string ” is not ” Place the resulting string in text2 variable.

3.Find in text2 the following pattern: Four digits starting at the end of the string. Use a function that returns the starting point of the found string and place its result in string\_pos4.

4.Probability that a normal random variable with mean 22 and variance 25

(i) lies between 16.2 and 27.5

(ii) is greater than 29

(iii) is less than 17

(iv) is less than 15 or greater than 25

5.Probability that in 60 tosses of a fair coin the head comes up

(i) 20,25 or 30 times (ii) less than 20 times (iii) between 20 and 30 times

**Code:**

# question 1

text1 = "Hello 123"

my\_pattern = '[[:upper:][:digit:]]'

print(grepl(my\_pattern,text1))

# question 2

text2 = "Hello 123"

print(sub(my\_pattern,'is not',text1))

# question 3

my\_pattern <- "\\d{4}$"

string\_pos4 <- gregexpr(my\_pattern,text2)

print(string\_pos4)

print(string\_pos4[[1]][1])

# question 4

# sd=5 since var =25

# lies between 16.2 and 27.5

res = pnorm(27.5,mean=22,sd=5)-pnorm(16.2,mean=22,sd=5)

print(res)

# greater than 29

res = pnorm(29,mean=22,sd=5)

print(res)

# less than 17

res = pnorm(17,mean=22,sd=5,lower.tail=FALSE)

print(res)

# less than 15 and greater than 25

res = pnorm(15,mean=22,sd=5)+pnorm(25,mean=22,sd=5,lower.tail=FALSE)

print(res)

# question 5

# (i)20,25 or 30 times

res = dbinom(20,60,prob=0.5) + dbinom(25,60,prob=0.5) +dbinom(30,60,prob=0.5)

print(res)

# (ii)less than 20 times

res = pbinom(19,60,prob=0.5)

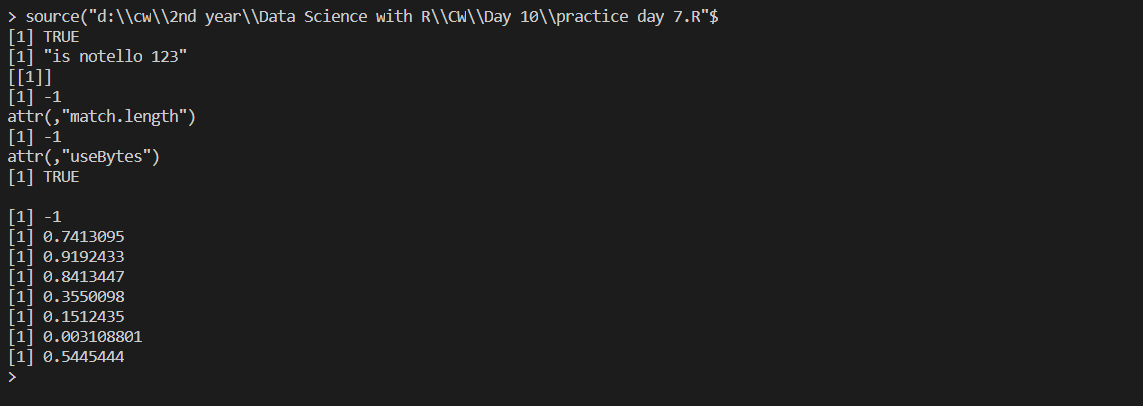
print(res)

# (iii)between 20 and 30 times

res = pbinom(30,60,prob=0.5) - pbinom(20,60,prob=0.5)

print(res)

**Output:**



**Day - 9**

**Questions:**

Exercise 1 Load the Boston Housing dataset from the mlbench library and inspect the different types of variables present.

Exercise 2 Explore and visualize the distribution of our target variable.

Exercise 3 Explore and visualize any potential correlations between medv and the variables crim, rm, age, rad, tax and lstat.

Exercise 4 Set a seed of 123 and split your data into a train and test set using a 75/25 split. You may find the caret library helpful here.

Exercise 5 We have seen that crim, rm, tax, and lstat could be good predictors of medv. To get the ball rolling, let us fit a linear model for these terms.

Exercise 6 Obtain an r-squared value for your model and examine the diagnostic plots found by plotting your linear model.

Exercise 7 Create a data frame of your predicted values and the original values.

Exercise 8 Plot this to visualize the performance of your model.

**Code:**

# install.packages('mlbench')

# install.packages("corrplot")

library(mlbench)

library(corrplot)

library(ggplot2)

library(caTools)

# question 1

data('BostonHousing')

data = BostonHousing

print(str(data))

# question 2

plot(density(data$medv),type='h',col='pink',main="Boston Housing")

# question 3

mydata <- data[data$medv, c("crim","rm","age","rad","tax","lstat")]

head(mydata)

cor<-round(cor(mydata),2)

head(cor)

print(corrplot(cor))

# question 4

s = sample.split(data,0.75)

train = data[s,]

test = data[s==FALSE,]

print(nrow(train))

print(nrow(test))

# question 5

linearModel = lm(medv~crim+rm+tax+lstat ,data=data)

pred = predict(linearModel,test)

print(head(pred))

# question 6

sum = summary(linearModel)

print(sum$r.squared)

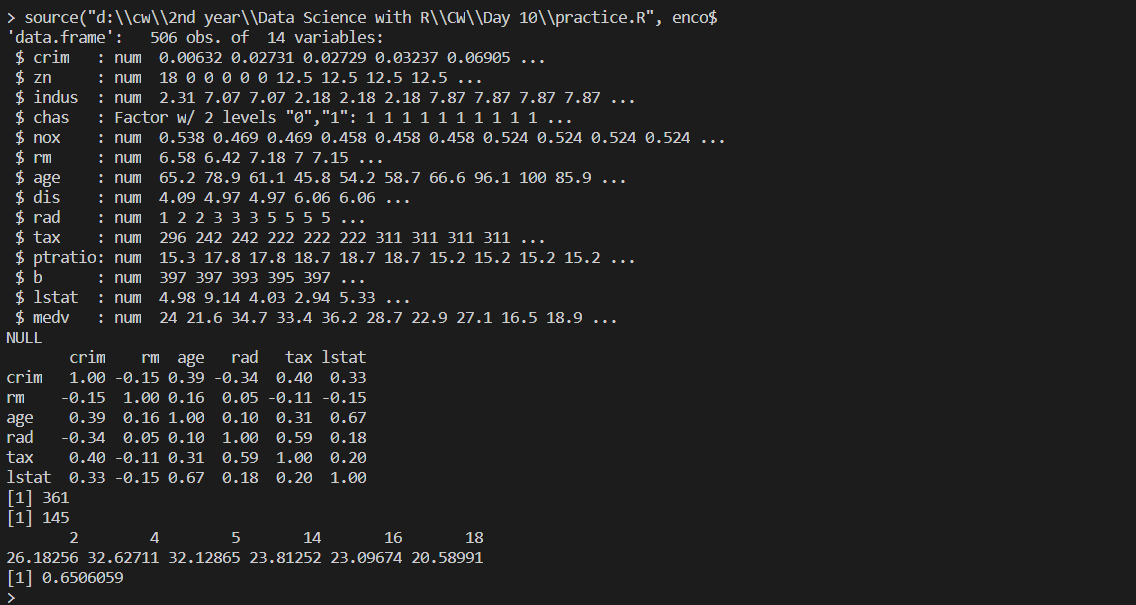
# question 7

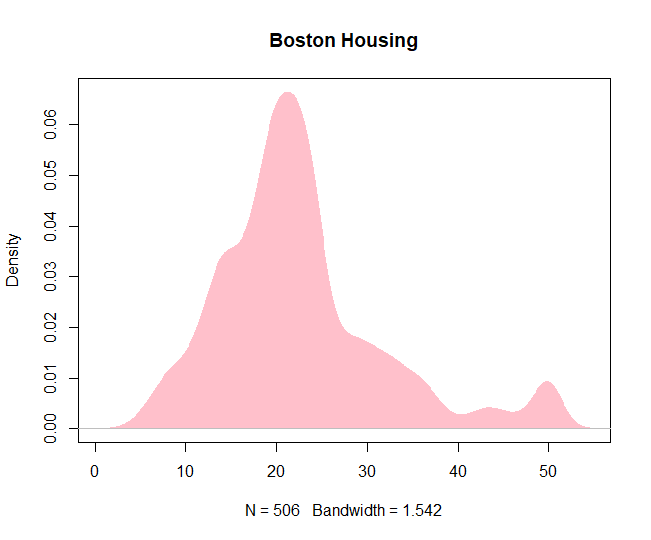
act\_vs\_pred = data.frame(actual=test$medv,predicted=pred)

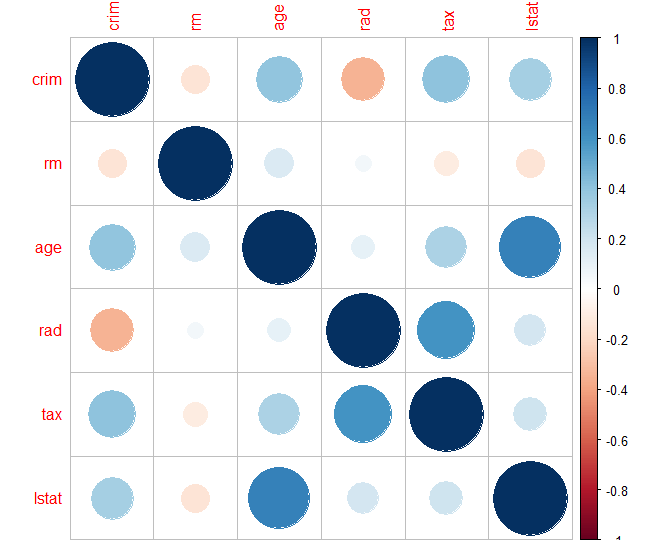
# question 8

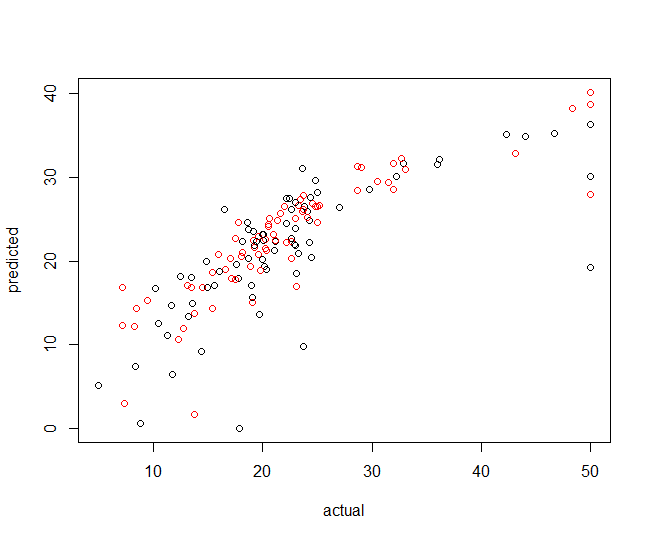
plot(act\_vs\_pred,type='p',col=c('black','red'))

**Output:**









**Day - 10**

**Questions:**

Exercise 1

Install and load the package googleVis.

Create a data frame

First of all let’s create an experimental data.frame to use for all our plots. This is an example:

dfr=data.frame(name=c("GRE", "ARG", "BRA"),

val1=c(20,32,19),

val2=c(25,52,12))

Exercise 2

Create a data frame named “df”. Give as variables the “Pts” (Points) and “Rbs” (Rebounds) of three NBA players. Names and values are up to you.Note:

Note:

Line Chart

To produce a Line Chart you can use:

LineC <- gvisLineChart(df)

plot(LineC)

Exercise 3

Create a list named “LineC” and pass to it the “df” data frame you just created as a line chart. HINT: Use gvisLineChart().

Exercise 4

Create a single axis Line chart that displays only the “Pts” of the “df” data frame.

Exercise 5

create a two axis line chart that displays both “Pts” and “Rbs” of the “df” data frame. HINT: Use list().

Note:

BarC <- gvisBarChart(df)

plot(BarC)

Exercise 6

Create a list named “BarC” and pass to it the “df” data frame you just created as a bar chart. HINT: Use gvisBarChart().

**Code:**

# install.packages('googleVis')

library(googleVis)

# question 1

dfr=data.frame(name=c("GRE", "ARG", "BRA"),val1=c(20,32,19),val2=c(25,52,12))

# question 2

df = data.frame(name=c('Thompson','Harden','James'),pts=dfr$val1,rbs=dfr$val2)

# question 3

LineC <- gvisLineChart(df)

# question 4

l=gvisLineChart(df,"name",c("pts"))

plot(l)

# question 5

l2=gvisLineChart(df,"name",c("pts","rbs"),

                 options=list(

                  series="[{targetAxisIndex: 0},

                         {targetAxisIndex:1}]",

                   vAxes="[{title:'Pts'}, {title:'Rbs'}]"))

plot(l2)

# question 6

BarC=gvisBarChart(df)

plot(BarC)

**Output:**

