Week-1:

2+3

a=2

b=3

c=a+b

c

plot(a,b)

a<-3

b<-3

c<-a+b

c

2->a

3->b

a+b->c

c

for(i in 1:5)print(1:i)

1+1

1+2+3

3\*7\*2

4/2

4/3

4\*6+5

(4\*6)+5

4 \* (6+5)

x<-2

x

2->x

x

a<-b<-7

a

assign("i",7)

i

i

theVariable<- 17

theVariable

is.numeric(a)

x<-"Narasimham"

is.numeric(x)

class(x)

class(a)

a<-2L

is.integer(a)

is.numeric(a)

class(4L)

class(2.8)

4L\*2.8

class(4L\*2.8)

5L/2L

class(5L/2L)

date<-as.Date("2001-11-30")

date

class(date)

as.numeric(date)

date1<-as.POSIXct('2012-06-28 17:42')

date1

as.numeric(date1)

class(date1)

class(as.numeric(date))

TRUE\*5

FALSE\*4

K<-TRUE

class(K)

d<-"Narasimham"

nchar(d)

2<=3

"date"<="stats"

Week-2:

Task-2:Vectors Operations

x<-c(1,2,3,4,5,6,7,8,9,10)

x

x+2

x-4

x\*2

x/2

x^2

sqrt(x)

y<-10:1

y

x+y

x-y

x\*y

x/y

x^y

length(x)

length(y)

length(x+y)

x+c(2,4)

x+c(1,2,3,4)

x<=5

x>0

any(x<y)

all(x<y)

m<-c("kohli","Dhoni","Rohit","Rahul","Gill")

nchar(m)

length(m)

m[1]

m[c(2,4)]

c(four="a",two="b",three="c")

w<-1:3

names(w)<-c("a","b","c")

w

m2<-c(m,"kohli","siraj","duplessis","dineshkarthik","Dhoni","Rohit","Gill")

m2factor<-as.factor(m2)

m2factor

z<-1:10:11

z

Week-3:

Task – 01 :

Aim :Apply different operations on Vectors.

`Write R code to create a vector of a specified type and length. Create vector of numeric, complex, logical and character types of length 6.

Program :

int=c(1,5,3,7,8,9)

int

complex<-c(5i,-1+6i,2i,0.3i,4+6i,-7i+8)

complex

logical<-c(2>3,3==2,4<6,4!=4,3<=6,4>=2)

logical

c("w","r","t","y","u","i")->char

char

Output :

int=c(1,5,3,7,8,9)

> int

[1] 1 5 3 7 8 9

> complex<-c(5i,-1+6i,2i,0.3i,4+6i,-7i+8)

> complex

[1] 0+5.0i -1+6.0i 0+2.0i 0+0.3i 4+6.0i

[6] 8-7.0i

> logical<-c(2>3,3==2,4<6,4!=4,3<=6,4>=2)

> logical

[1] FALSE FALSE TRUE FALSE TRUE TRUE

> c("w","r","t","y","u","i")->char

> char

[1] "w" "r" "t" "y" "u" "i"

2. Write R code to add two vectors of integer’s type and length 3

Program :

int1<-c(2,7,9)

length(int1)

int2<-c(0.8,1,0.66)

length(int2)

int1+int2

Output :

> int1<-c(2,7,9)

> length(int1)

[1] 3

> int2<-c(0.8,1,0.66)

> length(int2)

[1] 3

> int1+int2

[1] 2.80 8.00 9.66

3. Write R code to append value to a given empty vector

Program :

empty<-c()

empty<-append(int,10)

empty

empty<-append(empty,456)

empty

Output :

> empty<-c()

> empty<-append(int,10)

> empty

[1] 1 5 3 7 8 9 10

> empty<-append(empty,456)

> empty

[1] 1 5 3 7 8 9 10 456

4. Write R code to multiply two vectors of integer’s type and length 3.

Program :

int1\*int2

Output :

> int1\*int2

[1] 1.60 7.00 5.94a

5. Write R code to divide two vectors of integer’s type and length 3.

Program :

int1/int2

Output :

> int1/int2

[1] 2.50000 7.00000 13.63636

6. Write R code to find Sum, Mean and Product of a Vector

Program :

sum(int1)

mean(int1)

prod(int1)

Output :

> sum(int1)

[1] 18

> mean(int1)

[1] 6

> prod(int1)

[1] 126

7. Write R code to find Sum, Mean and Product of a Vector, ignore element like NA or NaN.

Program :

int3<-c(1,3,2,NA,5,NaN,4,NULL)

sum(int3,na.rm=TRUE)

prod(int3,na.rm=TRUE)

mean(int3,na.rm=TRUE)

Output :

> int3<-c(1,3,2,NA,5,NaN,4,NULL)

> sum(int3,na.rm=TRUE)

[1] 15

> prod(int3,na.rm=TRUE)

[1] 120

> mean(int3,na.rm=TRUE)

[1] 3

8. Write R code to find the minimum and the maximum of a Vector.

Program :

min(int1)

max(int1)

Output :

> min(int1)

[1] 2

> max(int1)

[1] 9

9. Write R code to sort a Vector in ascending and descending order.

Program :

order(int1)

order(-int1)

Output :

> order(int1)

[1] 1 2 3

> order(-int1)

[1] 3 2 1

10. Write R code to test whether a given vector contains a specified element.

Program :

array<-c(3,6,3,6,1,6,3,7,7,8,5,8,45)

is.element(4,array)

Output :

> array<-c(3,6,3,6,1,6,3,7,7,8,5,8,45)

>is.element(4,array)

[1] FALSE

11.Write R code to count the specific value in a given vector..

Program :

numbers<-c(3,6,9,5,3,4,6,9,1,3,8)

sum(numbers==3)

Output :

> numbers<-c(3,6,9,5,3,4,6,9,1,3,8)

> sum(numbers==3)

[1] 3

12.Write R code to access the last value in a given vector.

Program :

numbers[11]

Output :

>numbers[11]

[1] 8

1.Write R code to find second highest value in a given vector.

Program :

l=length(array)

sort(array,partial=l-1)[l-1]

Output :

> l=length(array)

> sort(array,partial=l-1)[l-1]

[1] 8

2.Write R code to find nth highest value in a given vector.

Program :

array1 = c(10, 20, 30, 20, 20, 25, 9, 26)

print("Original Vectors:")

print(array1)

print("nth highest value in a given vector:")

print(" n = 3")

n = 3

print(sort(array1, TRUE)[n])

Output :

> array1 = c(10, 20, 30, 20, 20, 25, 9, 26)

>print("Original Vectors:")

[1] "Original Vectors:"

> print(array1)

[1] 10 20 30 20 20 25 9 26

>print("nth highest value in a given vector:")

[1] "nth highest value in a given vector:"

>print(" n = 3")

[1] " n = 3"

> n = 3

>print(sort(array1, TRUE)[n])

[1] 25

3.Write R code to find common elements from multiple vector.

Program :

comele=c(1,4,2,5,8,9)

comele1=c(1,4,6,9,3,5)

comele2=c(7,8,3,4,6,2)

comele4=intersect(intersect(comele,comele1),comele2)

comele4

Output :

>comele=c(1,4,2,5,8,9)

> comele1=c(1,4,6,9,3,5)

> comele2=c(7,8,3,4,6,2)

> comele4=intersect(intersect(comele,comele1),comele2)

> comele4

[1] 4

4.Write R code to convert given dataframe column(s) to a vector.

Program :

df1 = c(1,21,13,41,52)

df2 = c(67,72,85,90,10)

df3 = c(11,12,13,14,15)

df4 = c(16,17,18,19,20)

df<- data.frame(df1, df2, df3, df4)

df

Output :

> df1 = c(1,21,13,41,52)

> df2 = c(67,72,85,90,10)

> df3 = c(11,12,13,14,15)

> df4 = c(16,17,18,19,20)

>df<- data.frame(df1, df2, df3, df4)

>df

df1 df2 df3 df4

1 1 67 11 16

2 21 72 12 17

3 13 85 13 18

4 41 90 14 19

5 52 10 15 20

5.Write R code to extract every nth element of a given vector.

Program :

gv = 1:10

gv1 = gv[seq(1, length(gv), 5)]

gv1

Output :

>gv = 1:10

> gv1 = gv[seq(1, length(gv), 5)]

> gv1

[1] 1 6

6. Write R code to list the distinct values in a vector from a given vector.

Program :

repv<- c(10,20,20,20,30,31,32,54,54)

repv1 <- unique(repv)

repv1

Output :

>repv<- c(10,20,20,20,30,31,32,54,54)

> repv1 <- unique(repv)

> repv1

[1] 10 20 30 31 32 54

7. Write R code to find the elements of a given vector that are not in another given vector.

Program :

de = c(2,2,3,3,3,5)

de1 = c(5,5,6,6,7)

de2 = setdiff(de,de1)

de2

Output :

> de = c(2,2,3,3,3,5)

> de1 = c(5,5,6,6,7)

> de2 = setdiff(de,de1)

> de2

[1] 2 3

8. Write R code to reverse the order of given vector.

Program :

rev1 = c(1,2,3,4,5)

rev2 = rev(rev1)

rev2

Output :

> rev1 = c(1,2,3,4,5)

> rev2 = rev(rev1)

> rev2

[1] 5 4 3 2 1

9.Write R code to concatenate a vector.

Program :

con1 = c(1,2,3)

con2 = c(4,5,6)

con3 = c(con1,con2)

con3

Output :

> con1 = c(1,2,3)

> con2 = c(4,5,6)

> con3 = c(con1,con2)

> con3

[1] 1 2 3 4 5 6

10.Write R code to count number of values in a range in a given vector.

Program :

r1 = c(0,1,2,3,4,5,6,7,8,9,10)

r2 = sum(r1 > 2 & r1 < 9)

r2

Output :

> r1 = c(0,1,2,3,4,5,6,7,8,9,10)

> r2 = sum(r1 > 2 & r1 < 9)

> r2

[1] 6

11.Write R code to convert two columns of a data frame to a named

vector.

Program :

bikes=c('Honda','SUZUKI','TVS','RE')

models = c('Shine','Access','Star City +','Classic 350')

tab = data.frame(BIKE = bikes, MODEL = models)

tab

setNames(as.character(tab$BIKE), as.character(tab$MODEL))

Output :

> bikes=c('Honda','SUZUKI','TVS','RE')

> models = c('Shine','Access','Star City +','Classic 350')

> tab = data.frame(BIKE = bikes, MODEL = models)

> tab

BIKE MODEL

1 Honda Shine

2 SUZUKI Access

3 TVS Star City +

4 RE Classic 350

>setNames(as.character(tab$BIKE), as.character(tab$MODEL))

Shine Access Star City +

"Honda" "SUZUKI" "TVS"

Classic 350

"RE"

12.Write R code to create a vector and find the length and the dimension of the vector.

Program :

real = c(1,2,3,4,5,6)

length(real)

dim(real)

Output :

> real = c(1,2,3,4,5,6)

> length(real)

[1] 6

> dim(real)

NULL

13.Write R code to test whether the value of the element of a given vector greater than 10 or not. Return TRUE or FALSE.

Program :

whole = c(0,1,2,3,25,15,99,100)

whole > 10

Output :

> whole = c(0,1,2,3,25,15,99,100)

> whole > 10

[1] FALSE FALSEFALSEFALSE TRUETRUETRUETRUE

14.Write R code to add 3 to each element in a given vector. Print the original and new vector.

Program :

ori= c(1,2,3,4)

new = ori + 3

ori

new

Output :

>ori = c(1,2,3,4)

> new= ov + 3

>ori

[1] 1 2 3 4

> new

[1] 4 5 6 7

15.Write a R code to create a vector using: operator and seq() function.

Program :

vec =seq(from = 1, to = 30)

vec

Output :

>vec =seq(from = 1, to = 30)

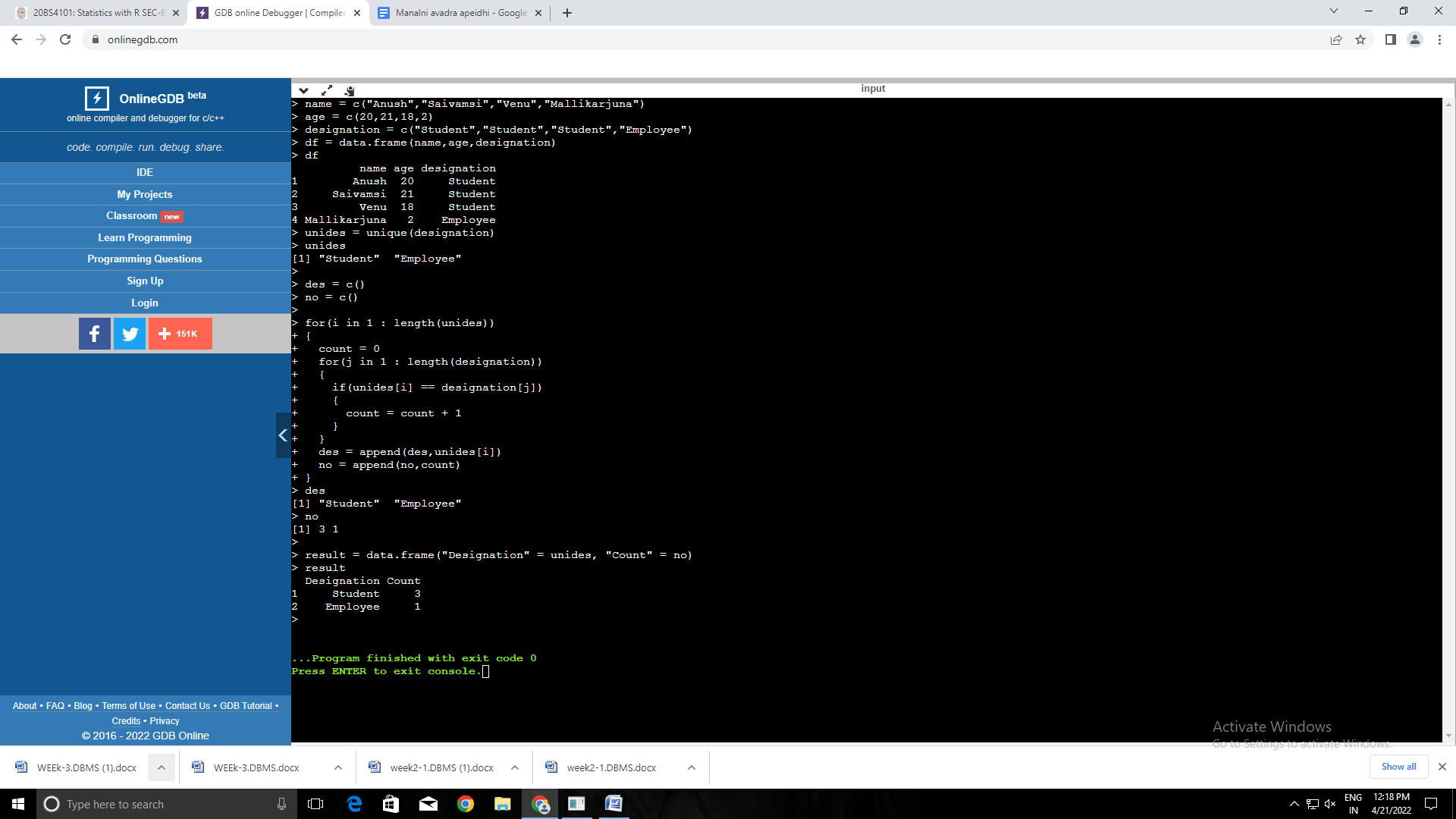
>vec

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

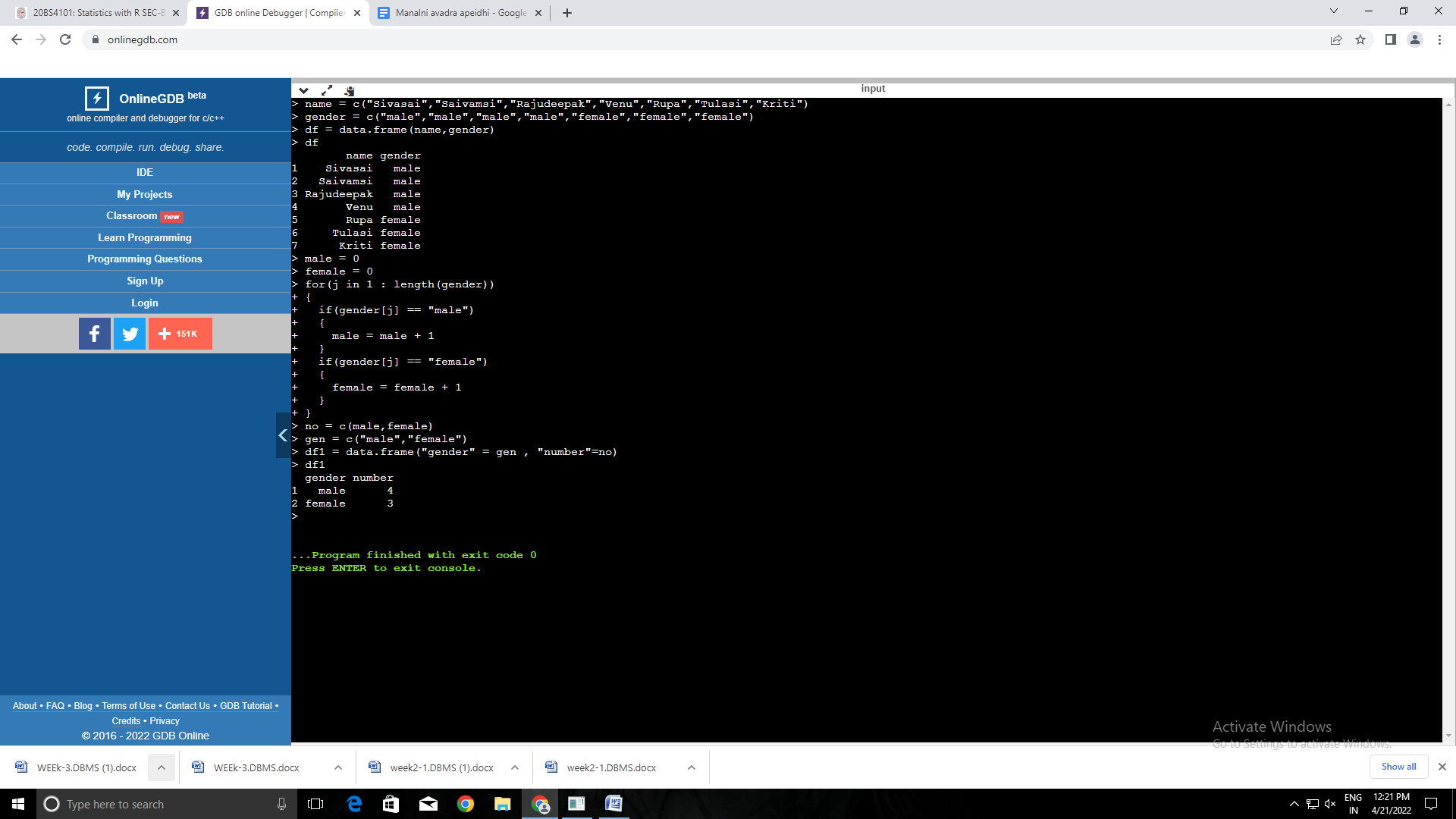
Week-4:

**DATAFRAMES**

**1.**

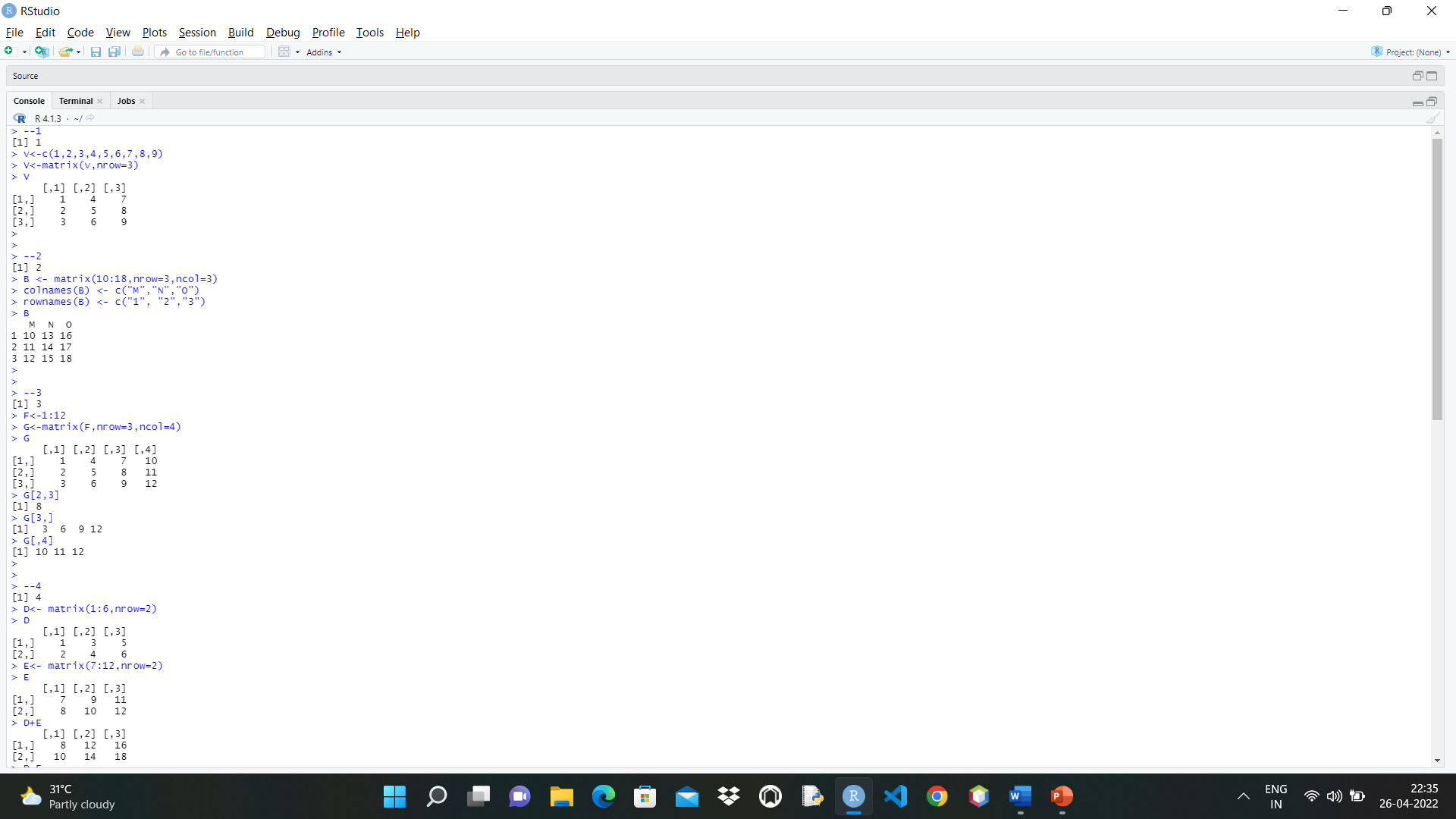


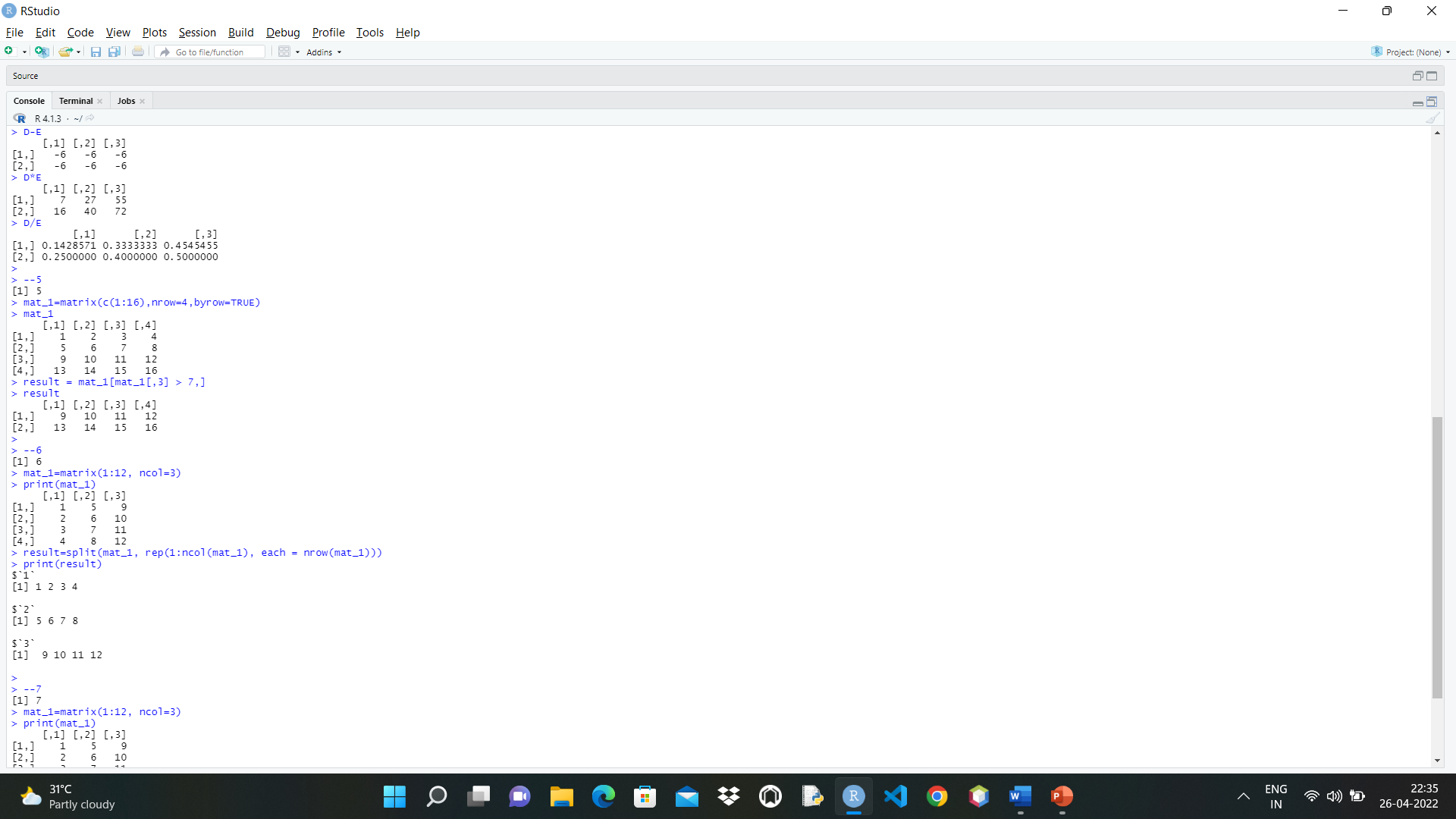
**2.**

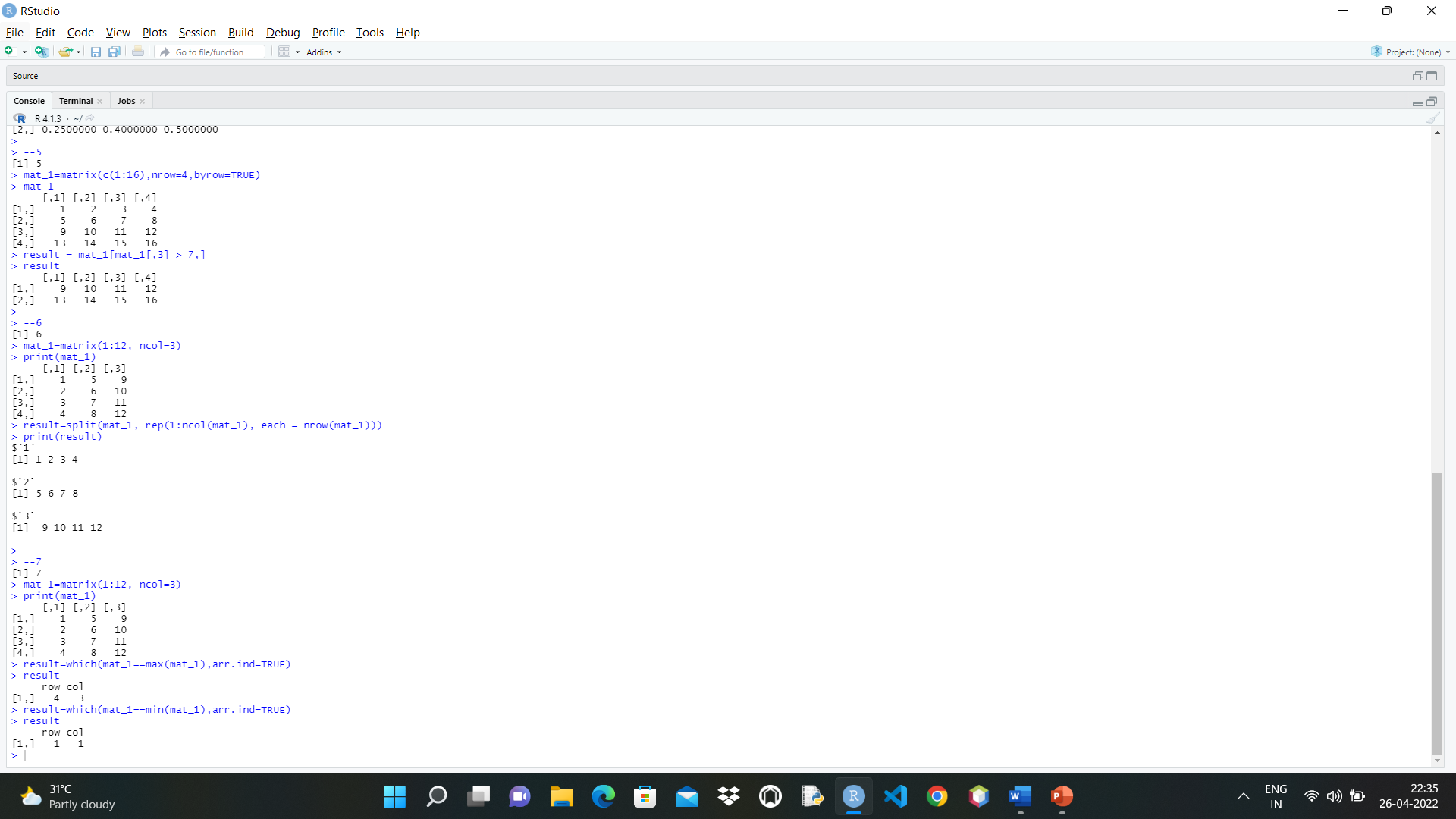


Week-5:

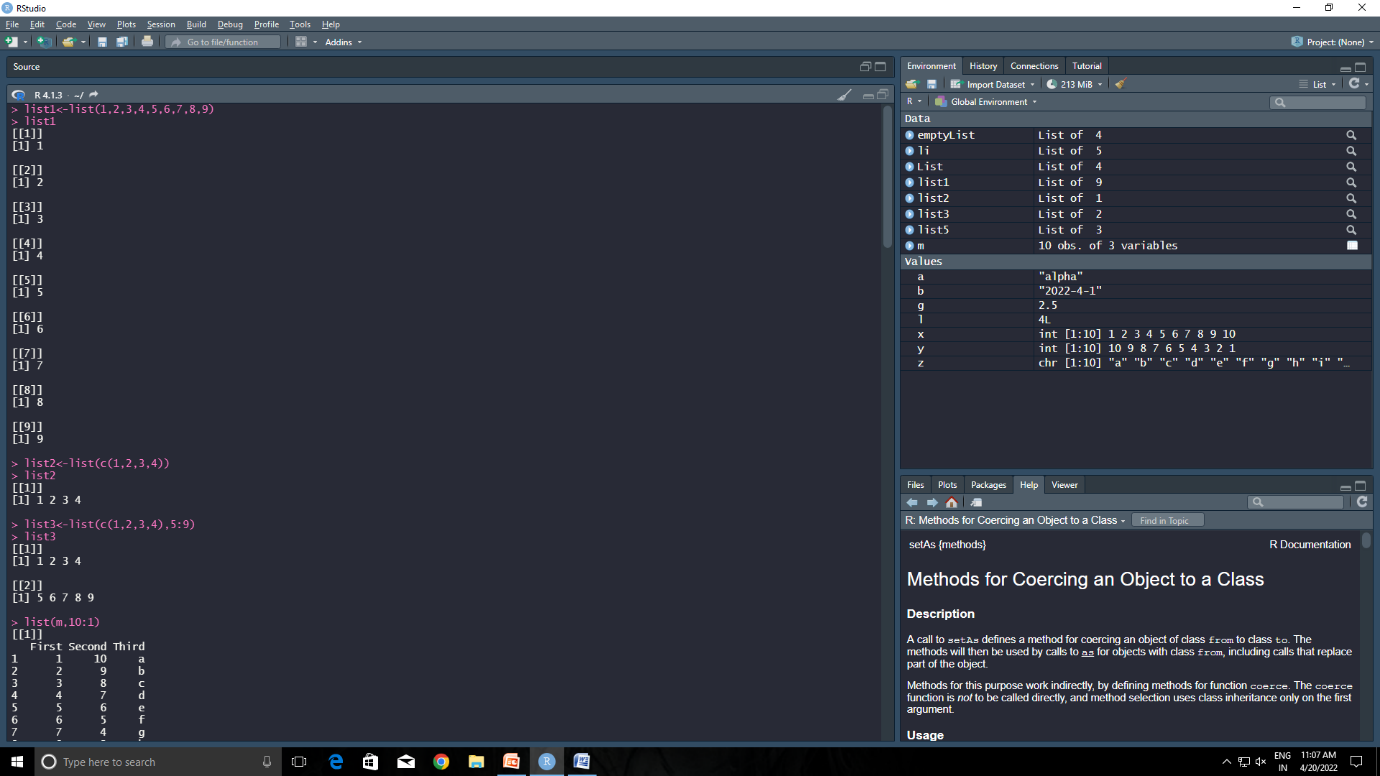
**Matrices**

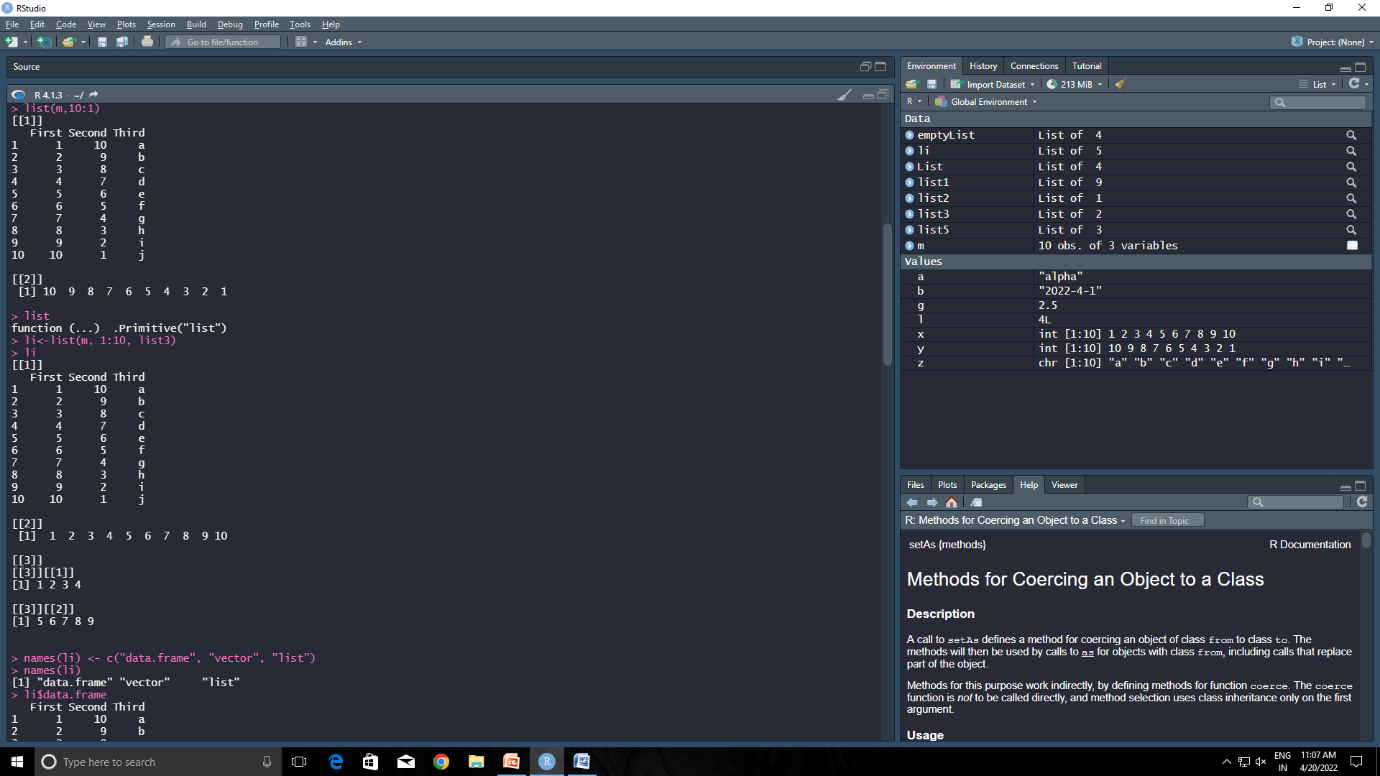


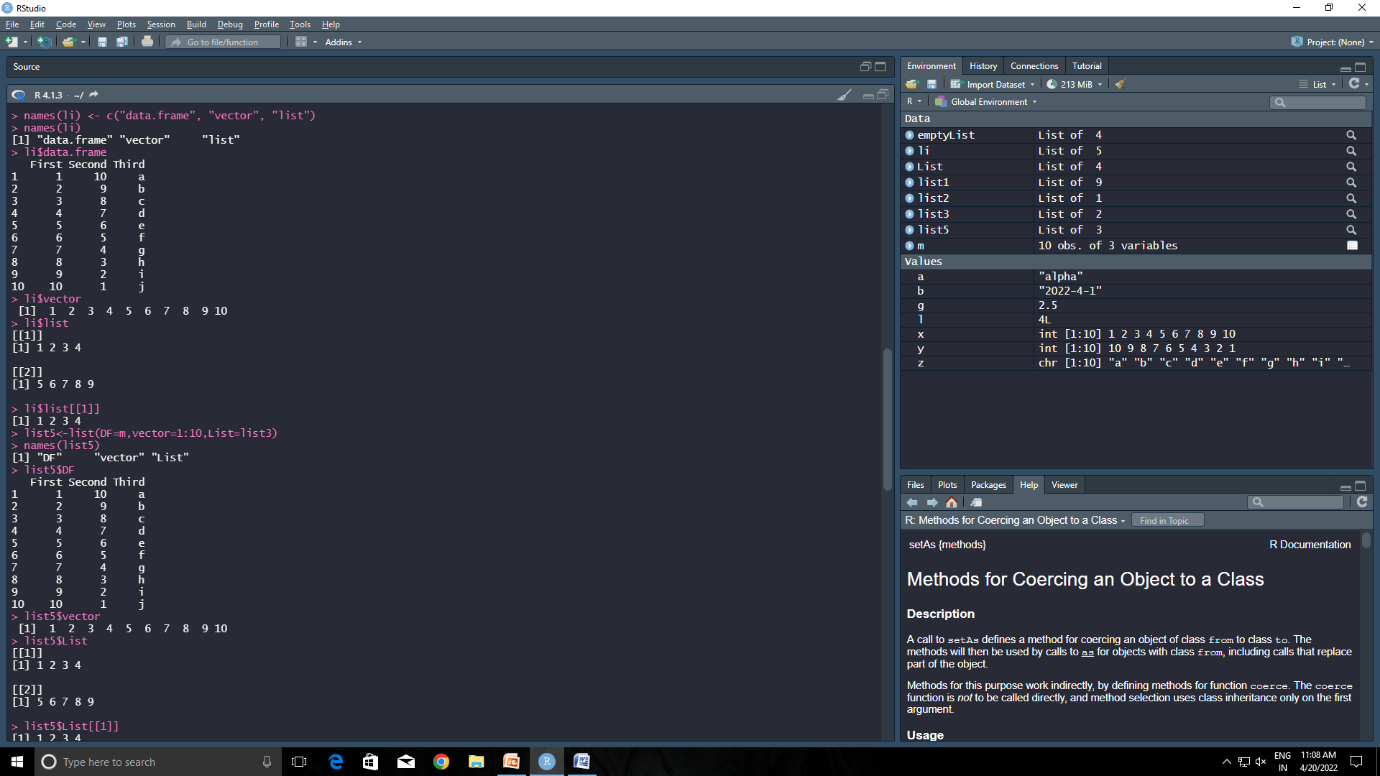


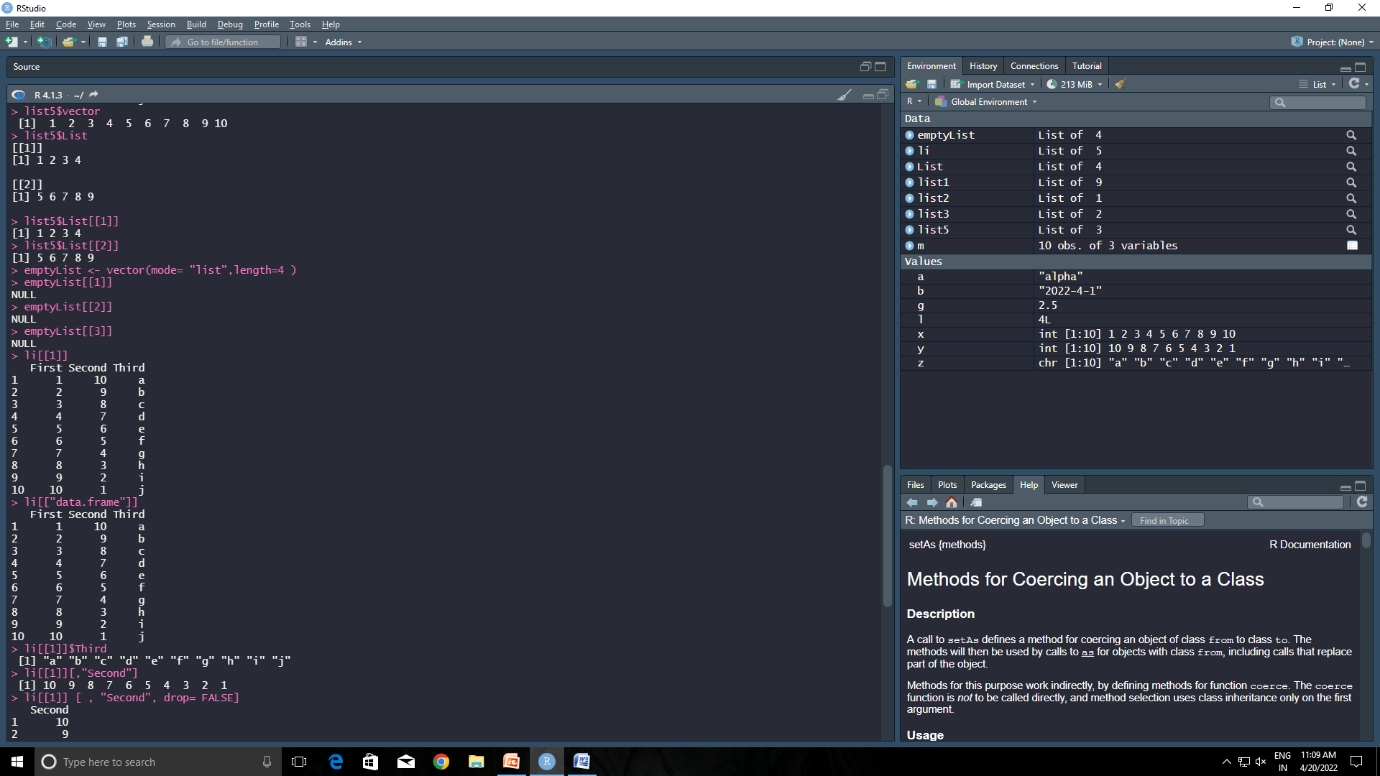


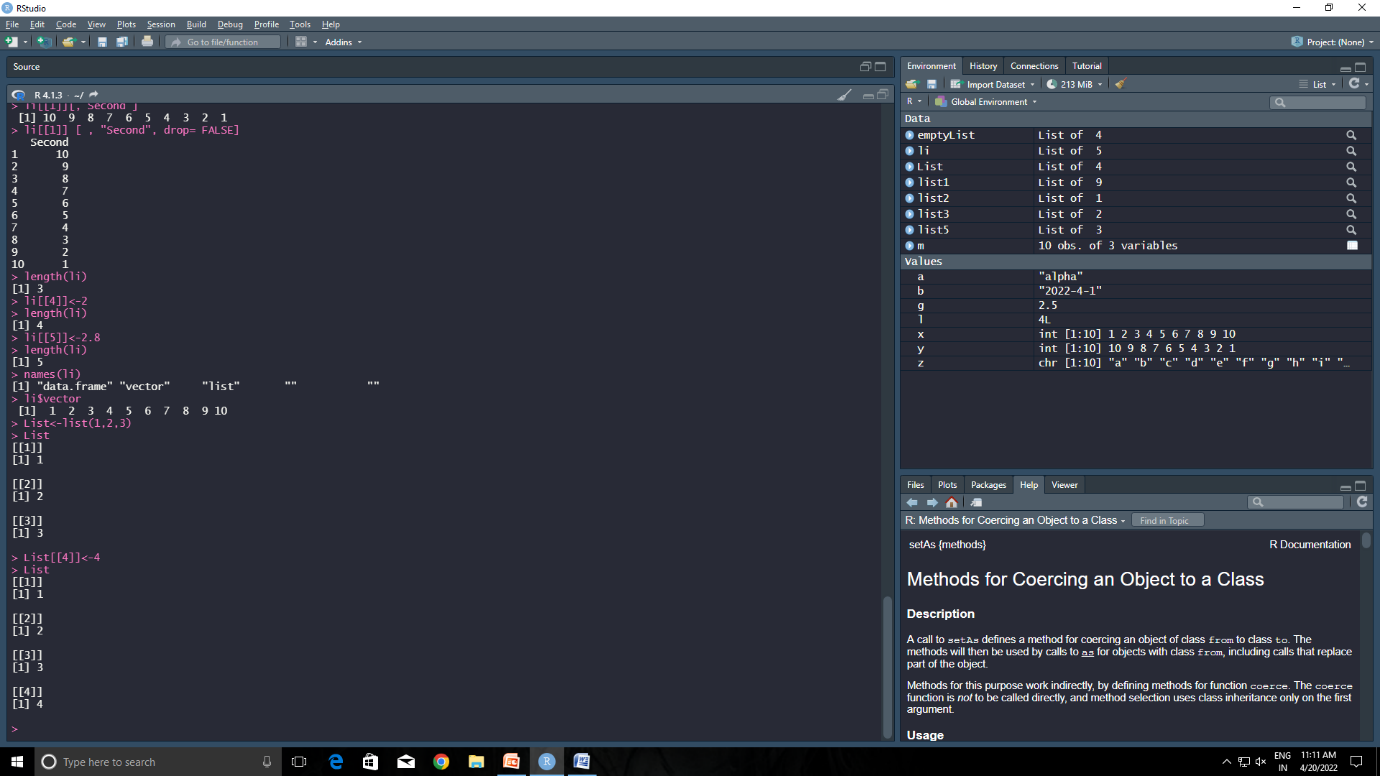
**Lists**











Week-6:

**Program:**

install.packages("RMySQL")

library(RMySQL)

mydb = dbConnect(MySQL(), user='root', password='', dbname='main', host='localhost')

mydb

dbListTables(mydb)

dbListFields(mydb, 'student\_info')

rs = dbSendQuery(mydb, "select \* from student\_info")

rs

data = fetch(rs, n=-1)

data

**output:**

>install.packages("RMySQL")

Error in install.packages : Updating loaded packages

>library(RMySQL)

>mydb = dbConnect(MySQL(), user='root', password='', dbname='main', host='localhost')

>mydb

<MySQLConnection:0,1>

>dbListTables(mydb)

[1] "student" "student\_info"

>dbListFields(mydb, 'student\_info')

[1] "name" "rollnumber"

>rs = dbSendQuery(mydb, "select \* from student\_info")

>rs

<MySQLResult:543833448,1,2>

> data = fetch(rs, n=-1)

> data

name rollnumber

1 Sai vamsi 208w1a1283

2 Siva sai208w1a1282

**CSV file:**

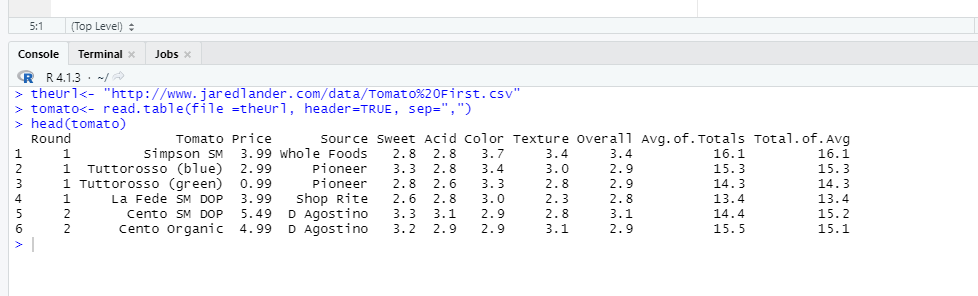
**Program:**

theUrl<- "http://www.jaredlander.com/data/Tomato%20First.csv"

tomato<- read.table(file =theUrl, header=TRUE, sep=",")

head(tomato)

**output:**

****

**Load .xl file into rstudio:**

**Program:**

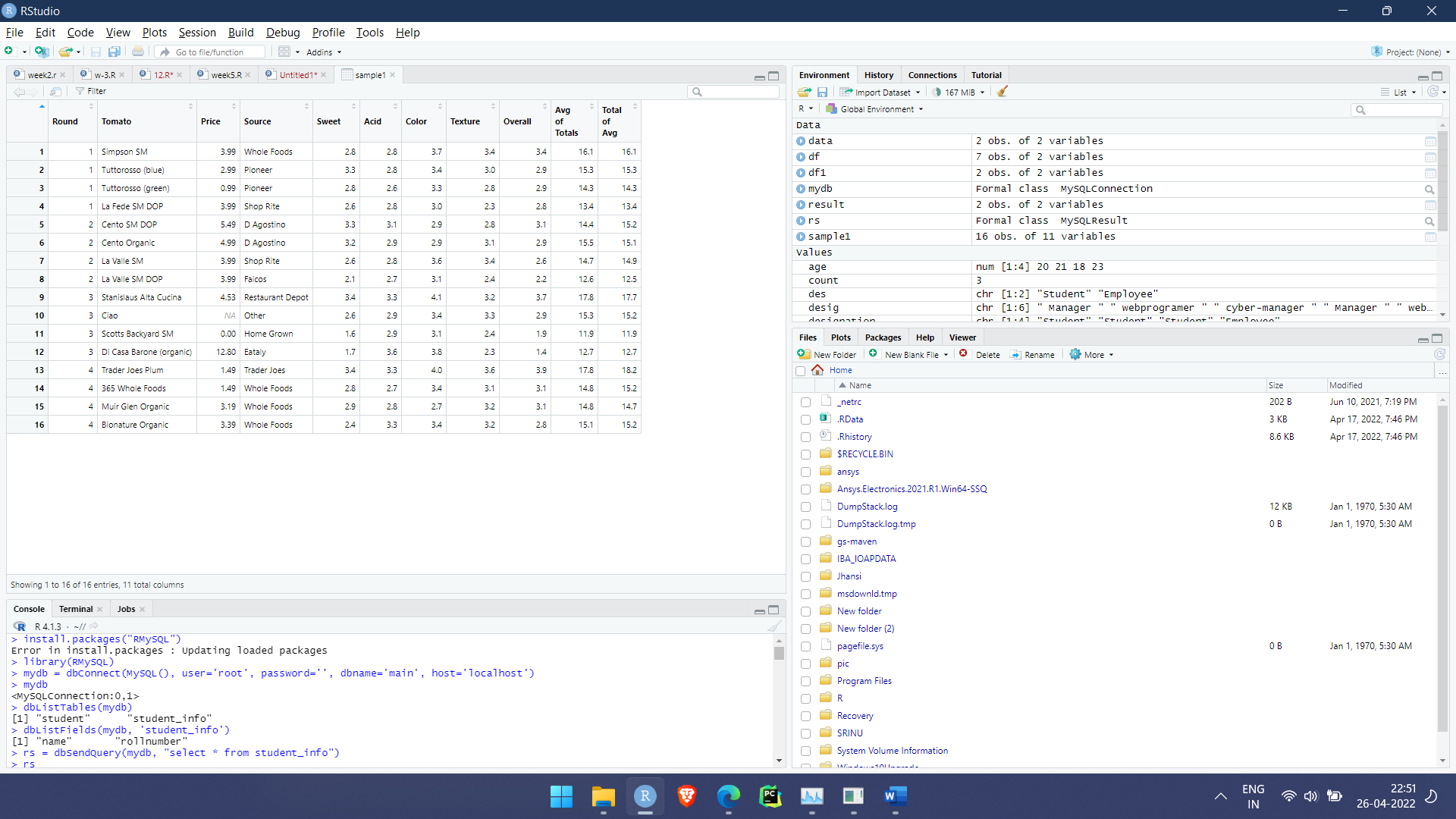
install.packages("readxl")

library(readxl)

sample1 <- read\_excel("C:/Users/hp/Desktop/mi/4th sem/statastics with r/Tomato First.xlsx")

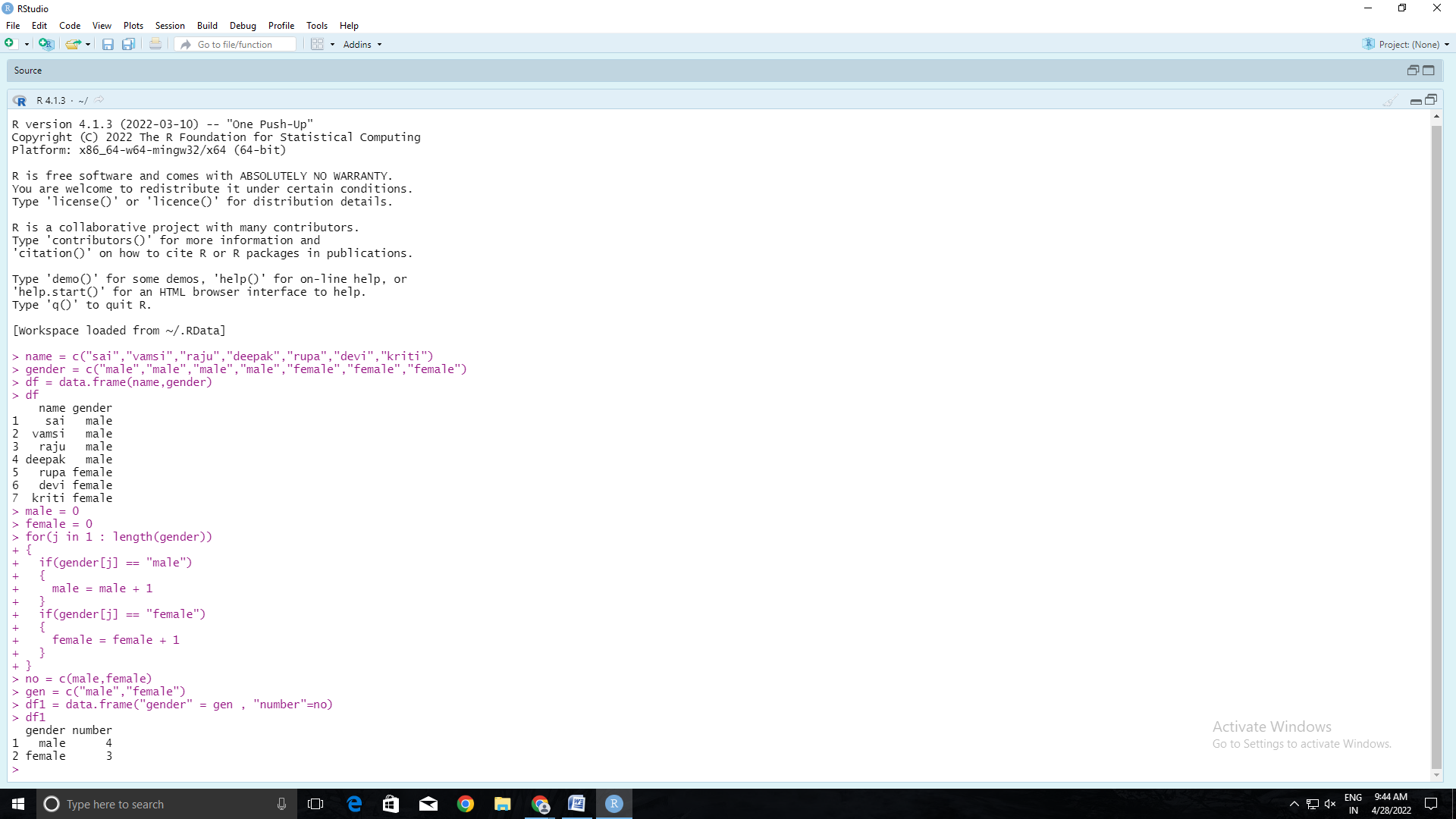
View(sample1)

**Output:**

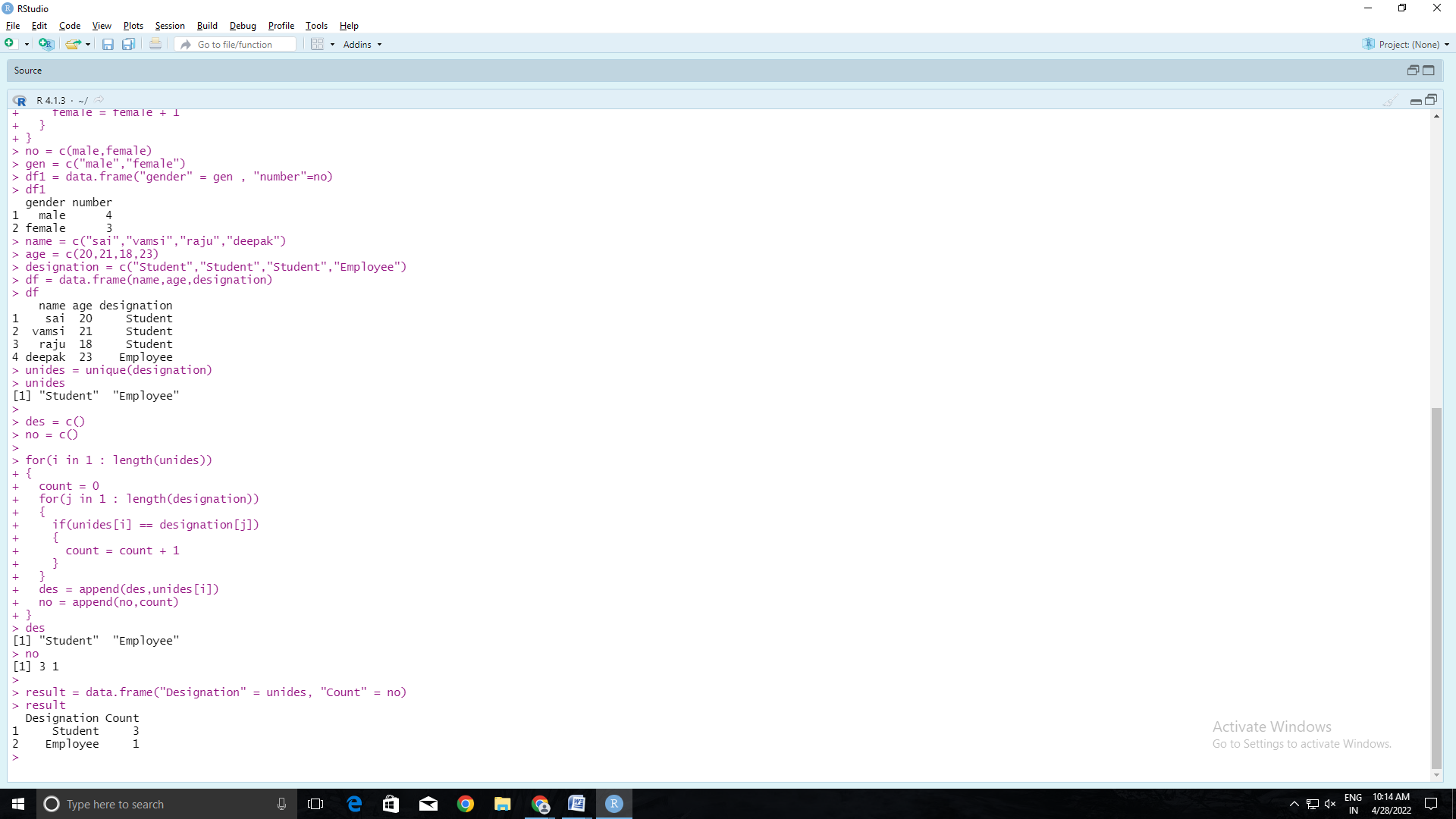
****

Assignment task:

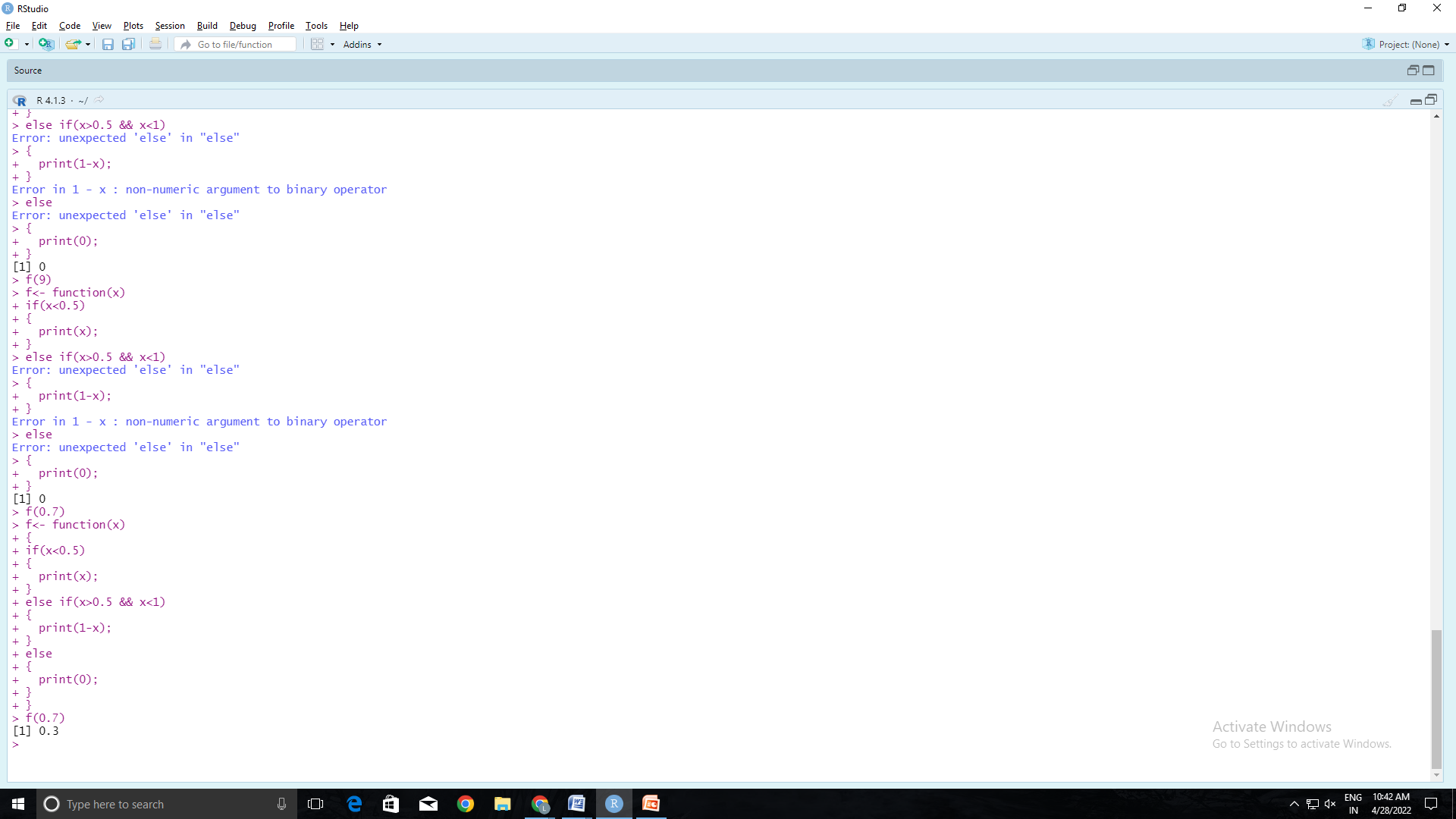
**3(b)**



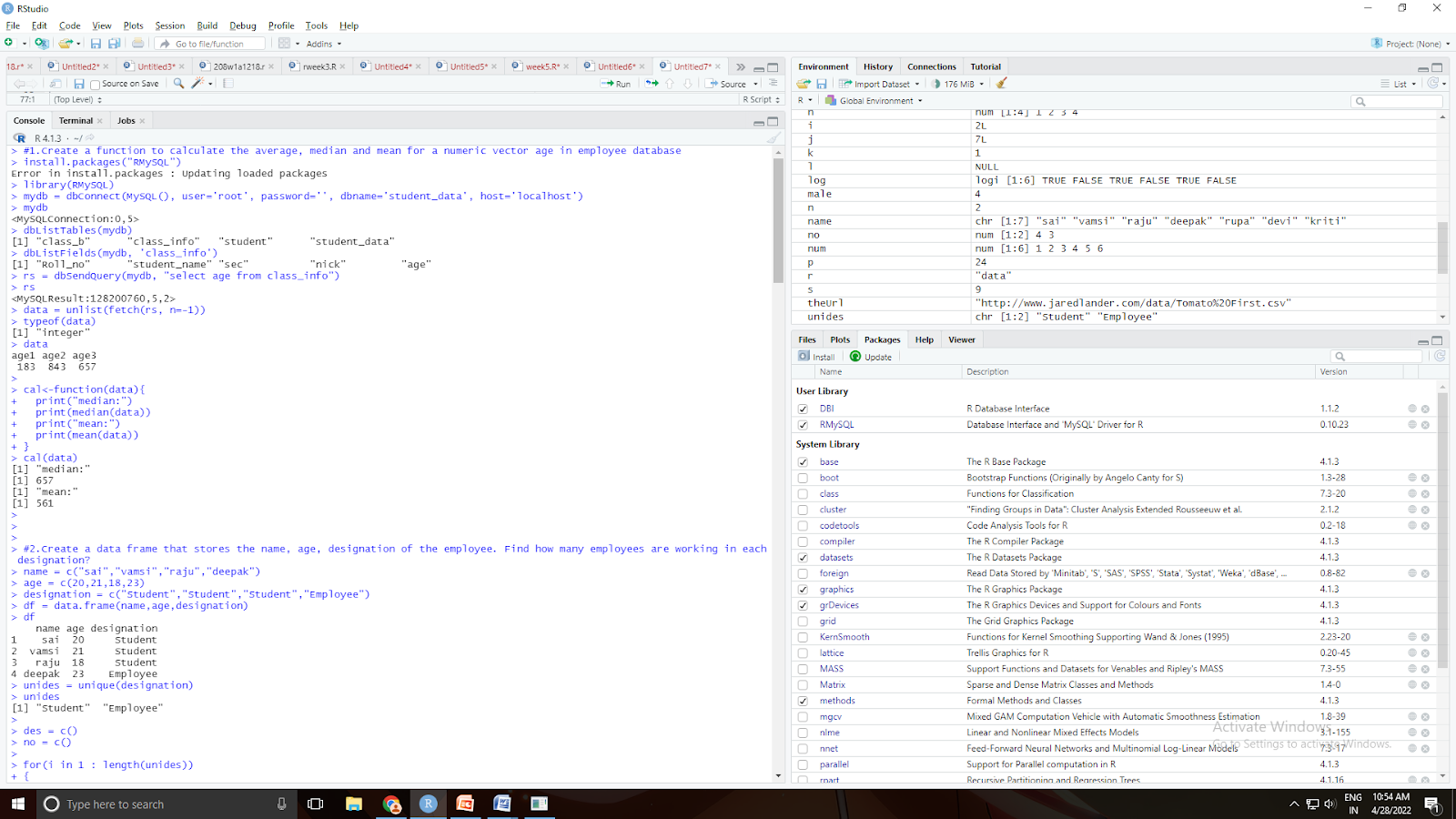
**2(c)**



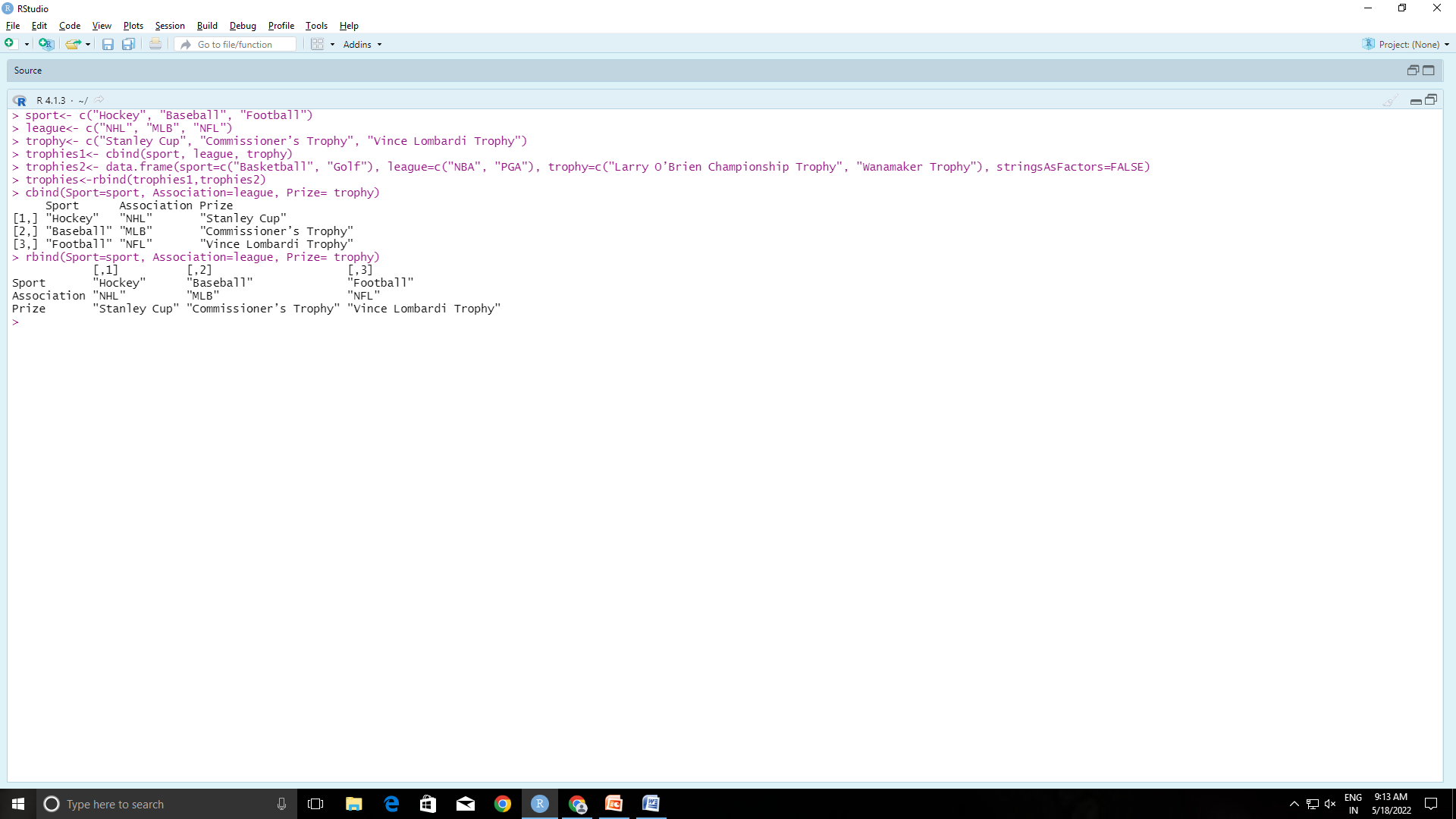
**4(c)**

****

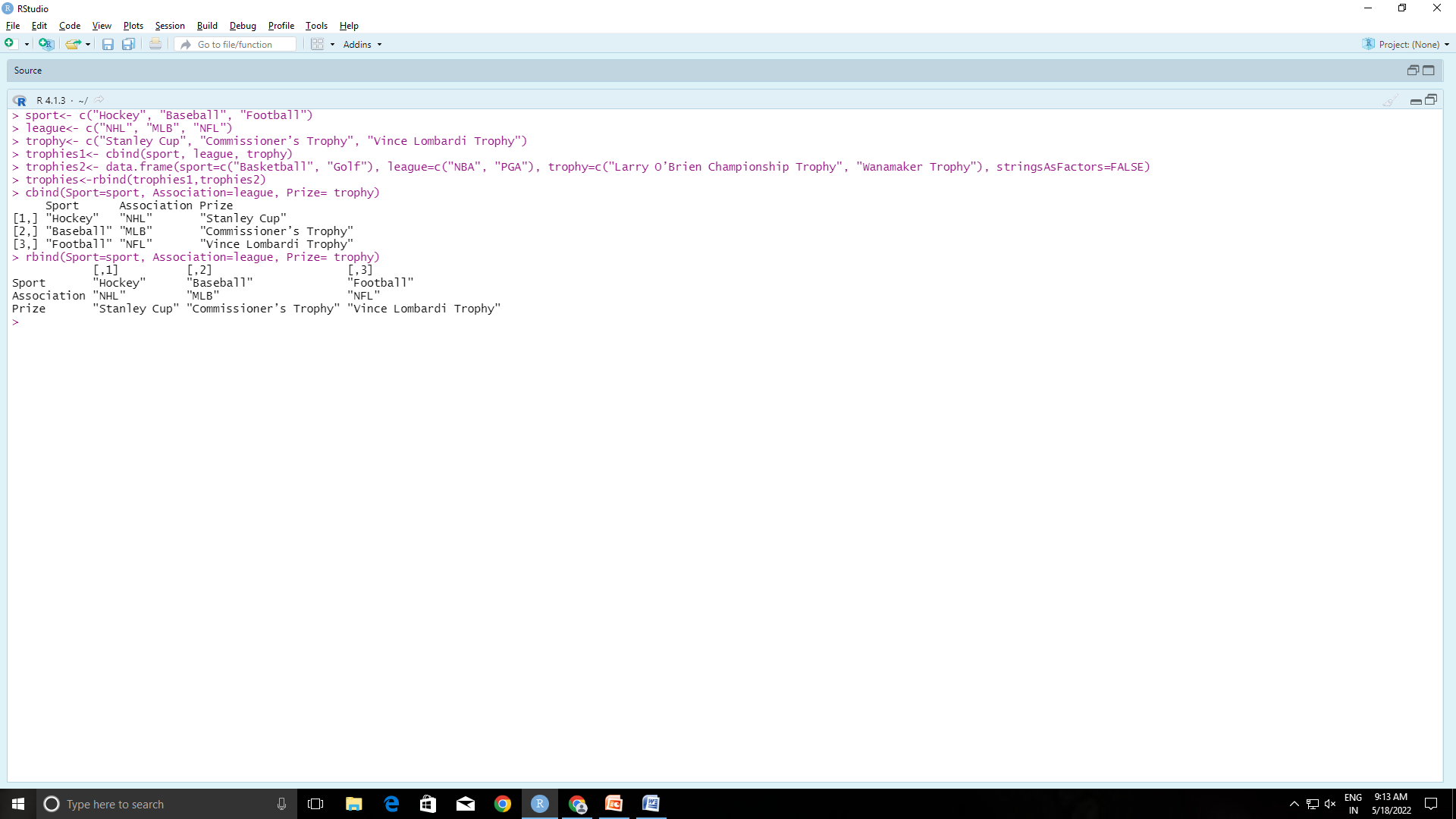
**1(b)**

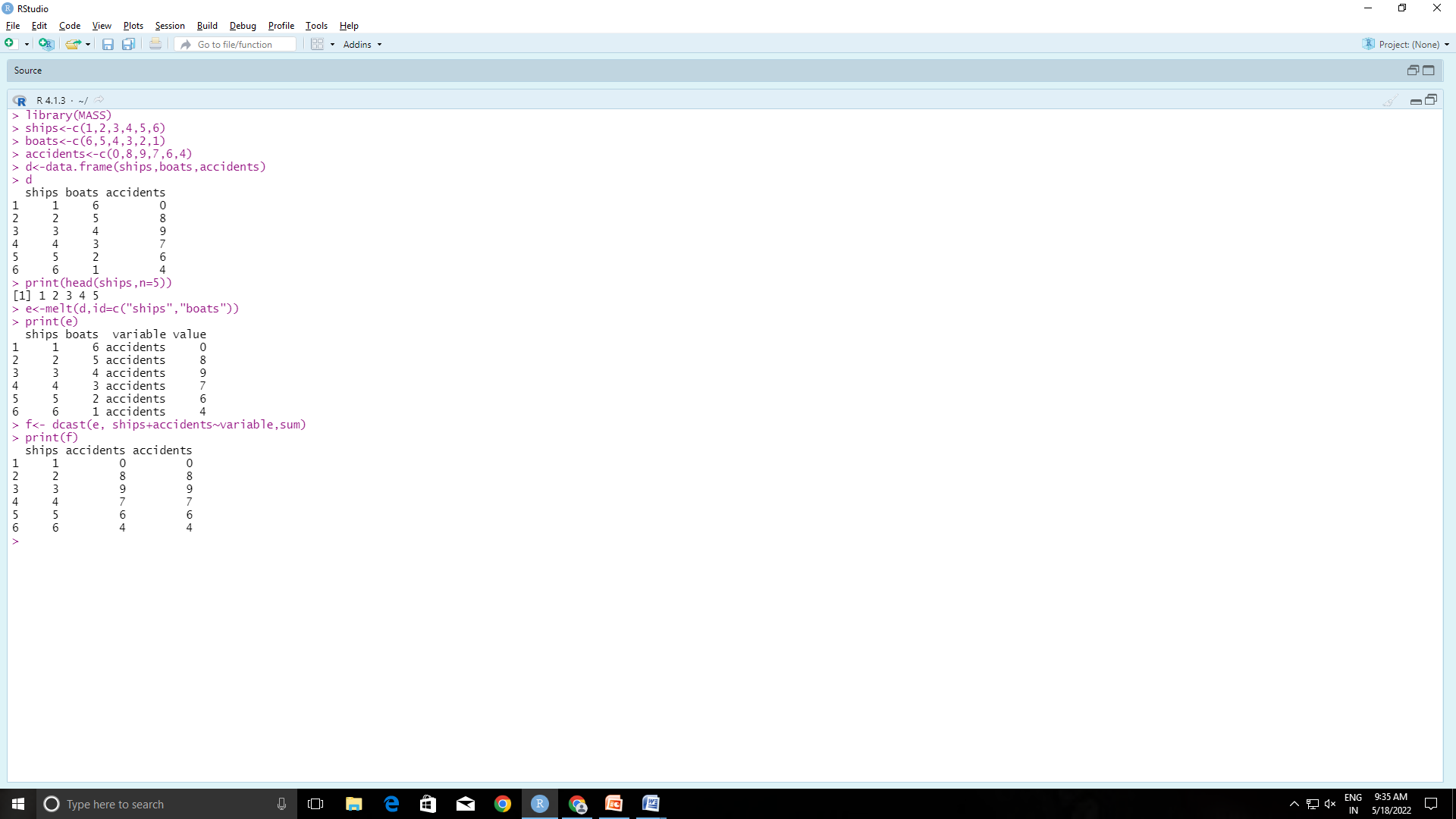


**cbind and rbind:**

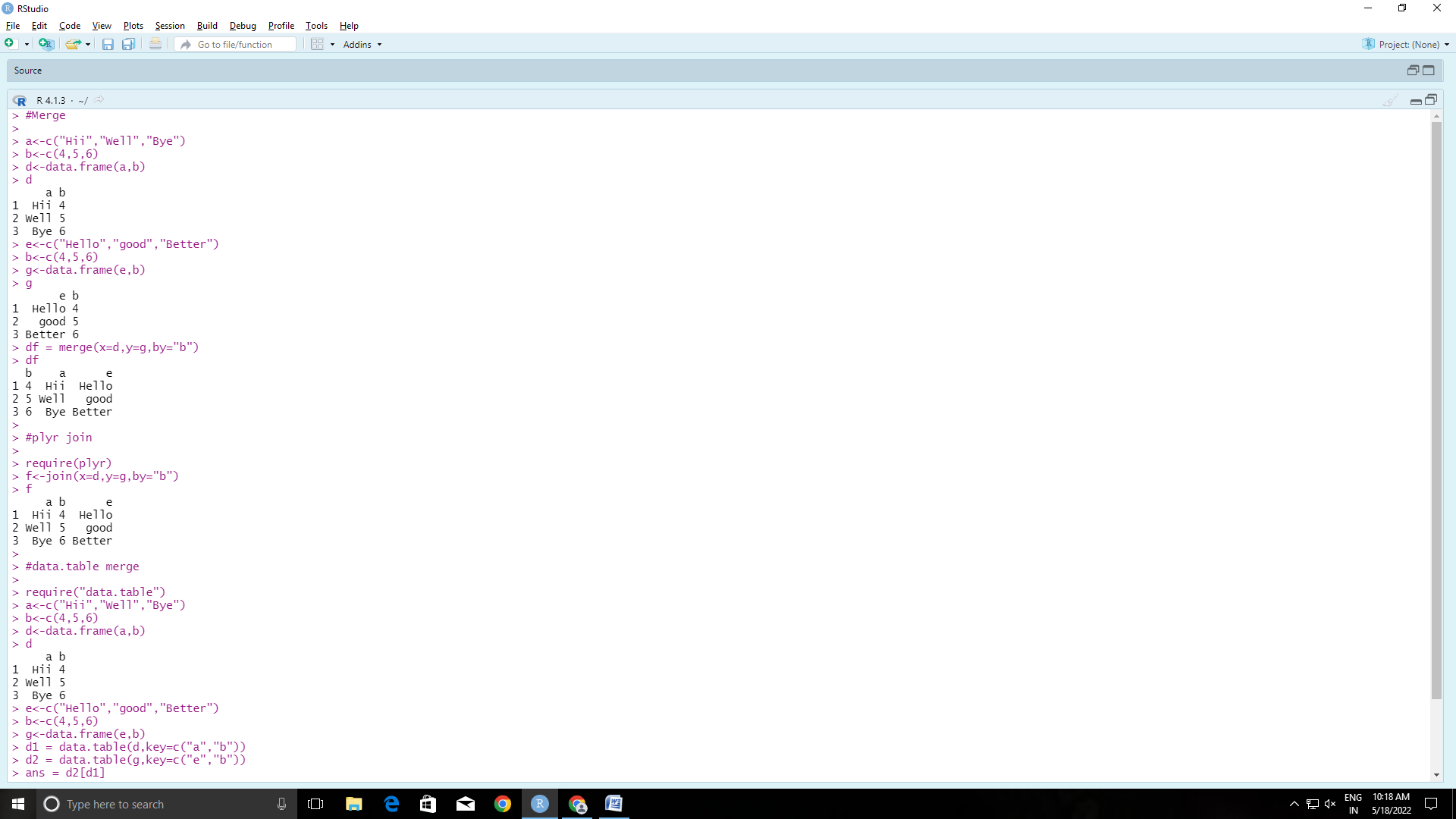
****

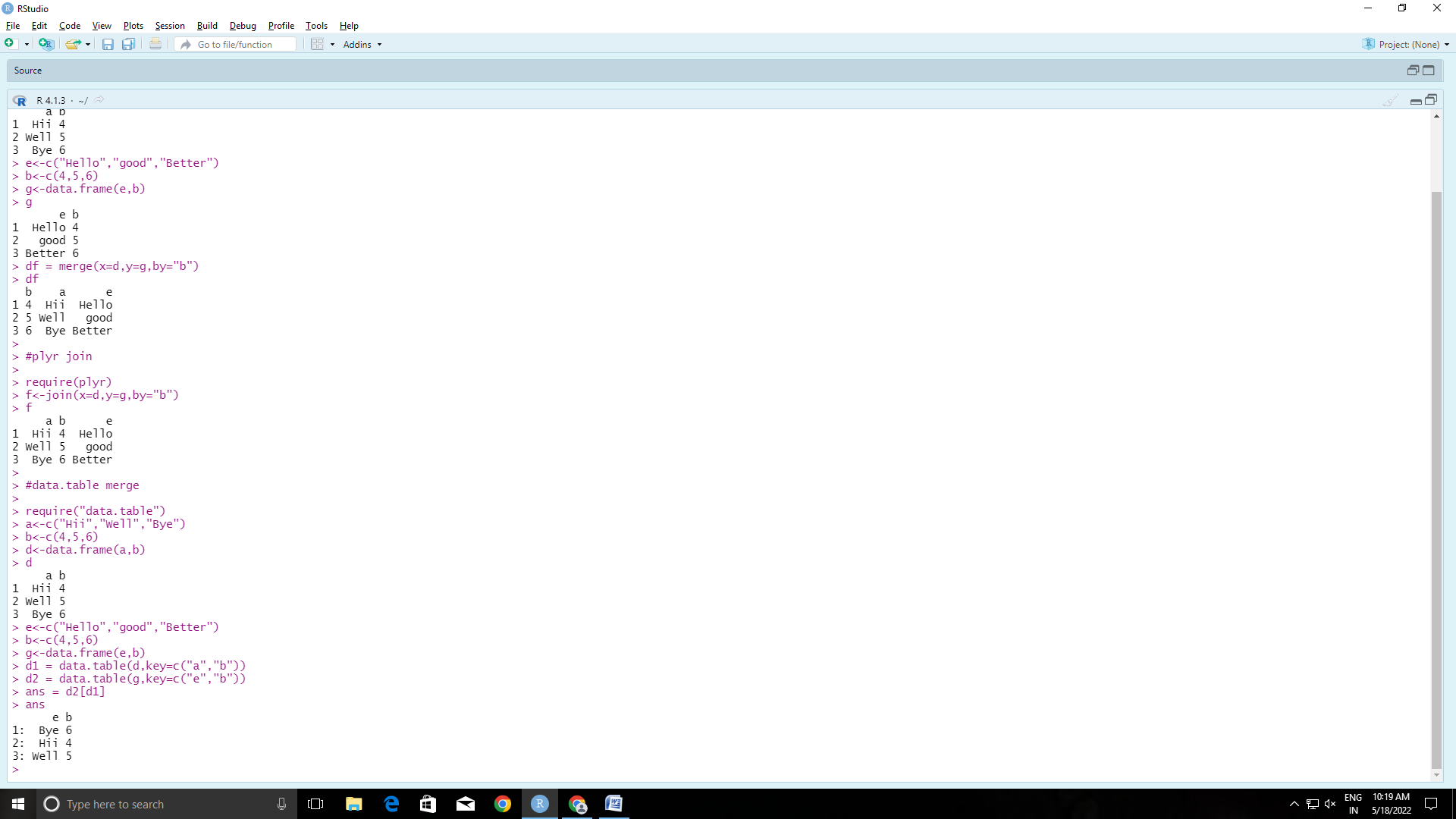
**Cast and melt:**

****



**Joins:**





#set operations

x<-c(1,2,3,4)

y<-c(4,5,6,7)

union(x,y)

intersect(x,y)

setdiff(x,y)

setequal(x,y)

setequal(x,c(1,2,3,4))

2 %in% x

2 %in% y

choose(5,2)

#Math functions

z<-c(8,9,7,6)

b<-c(10,2,3,4)

min(z)

max(z)

sqrt(16)

abs(-10.7)

ceiling(10.7)

floor(10.7)

exp(2)

log(10)

log10(10)

which.min(z)

which.max(z)

factorial(z)

a<-c(1.234,2.345,3.456)

round(a,2)

cumsum(z)

cumprod(z)

sum(z)

prod(z)

d<-c(1,7,3,9)

e<-c(5,6,8,8)

pmin(d,e)

pmax(d,e)

sin(90)

cos(0)

#Manipulating strings

paste("This","is","jerry")

paste("This","is","jerry",sep="/")

paste(c("This","is","tom"),c("This","is","jerry"))

paste("Hello",c("Prabhu","Sai","Anush"))

paste("Hello", c("Jared", "Bob", "David"), c("Goodbye", "Seeya"))

f<-c("Hii","Good","Morning",".")

paste(f, collapse=" ")

paste(f, collapse="\*")

#sprintf

person<- "Jared"

Size<-"eight"

WaitTime<- 25

sprintf("Hello %s,your party of %s will be seated in %s minutes.",person,Size,WaitTime)

#Extracting text

library("stringr")

hw <- "Hadley Wickham"

str\_sub(hw, 1, 6)

str\_sub(hw, end = 6)

str\_sub(hw, 8, 14)

str\_sub(hw, 8)

str\_sub(hw, c(1, 8), c(6, 14))

str\_sub(hw, -1)

str\_sub(hw, -7)

str\_sub(hw, end = -7)

pos <- str\_locate\_all(hw, "[aeio]")[[1]]

str\_sub(hw, pos)

str\_sub(hw, pos[, 1], pos[, 2])

str\_sub(hw, seq\_len(str\_length(hw)))

str\_sub(hw, end = seq\_len(str\_length(hw)))

x <- "BBCDEF"

str\_sub(x, 1, 1) <- "A"; x

str\_sub(x, -1, -1) <- "K"; x

str\_sub(x, -2, -2) <- "GHIJ"; x

str\_sub(x, 2, -2) <- ""; x

x1 <- x2 <- x3 <- x4 <- "AAA"

str\_sub(x1, 1, NA) <- "B"

str\_sub(x2, 1, 2) <- NA

str\_sub(x3, 1, NA, omit\_na = TRUE) <- "B"

str\_sub(x4, 1, 2, omit\_na = TRUE) <- NA

x1; x2; x3; x4

#Normal distribution

library(ggplot2)

a<-rnorm(100)

b<-dnorm(a)

d<-data.frame(x=a,y=b)

ggplot(d)+ aes(x=x,y=y) + geom\_point() + labs(x="Random Normal Variables", y= "Density")

ggplot(d)+ aes(x=x,y=y) + geom\_line() + labs(x="Random Normal Variables", y= "Density")

a

b

pnorm(a)

pnorm(c(1,2,3))

pnorm(-1)

pnorm(10)-pnorm(5)