SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0443 - STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEM

DAY 2 – LAB EXERCISES

Reg No:

Name:

IMPLEMENTATION OF VECTOR RECYCLING, APPLY FAMILY & RECURSION

1. Demonstrate Vector Recycling in R.

PROGRAM:

> (a<-1:10)

[1] 1 2 3 4 5 6 7 8 9 10

> (b<-1:5)

[1] 1 2 3 4 5

> (a+b)

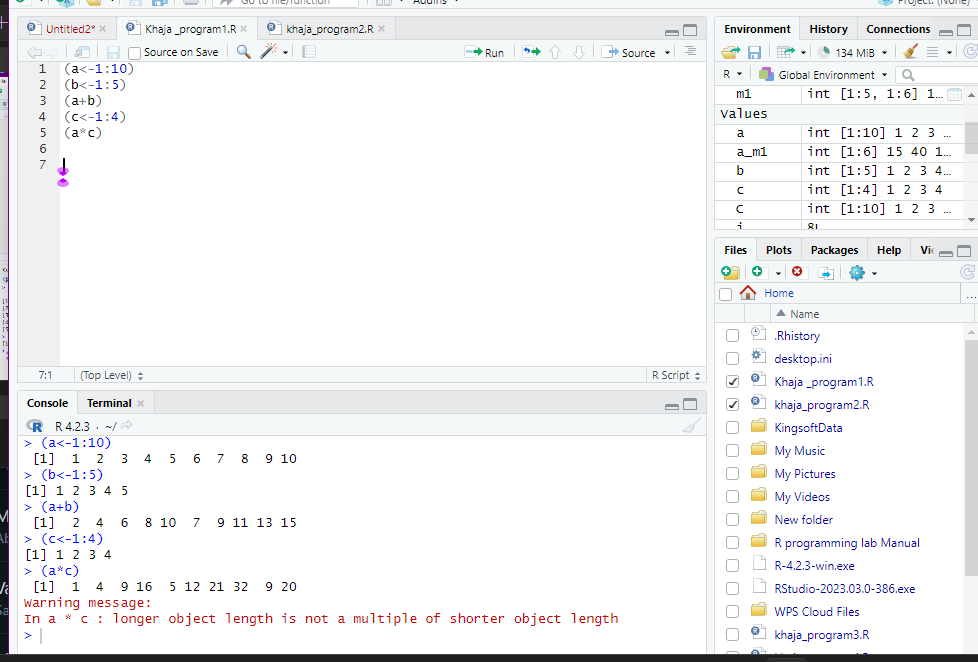
[1] 2 4 6 8 10 7 9 11 13 15

> (c<-1:4)

[1] 1 2 3 4

> (a\*c)

[1] 1 4 9 16 5 12 21 32 9 20



2. Demonstrate the usage of apply function in R

SOURCE CODE:

> m1 <- matrix(C<-(1:10),nrow=5, ncol=6)

> m1

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 1 6 1 6 1 6

[2,] 2 7 2 7 2 7

[3,] 3 8 3 8 3 8

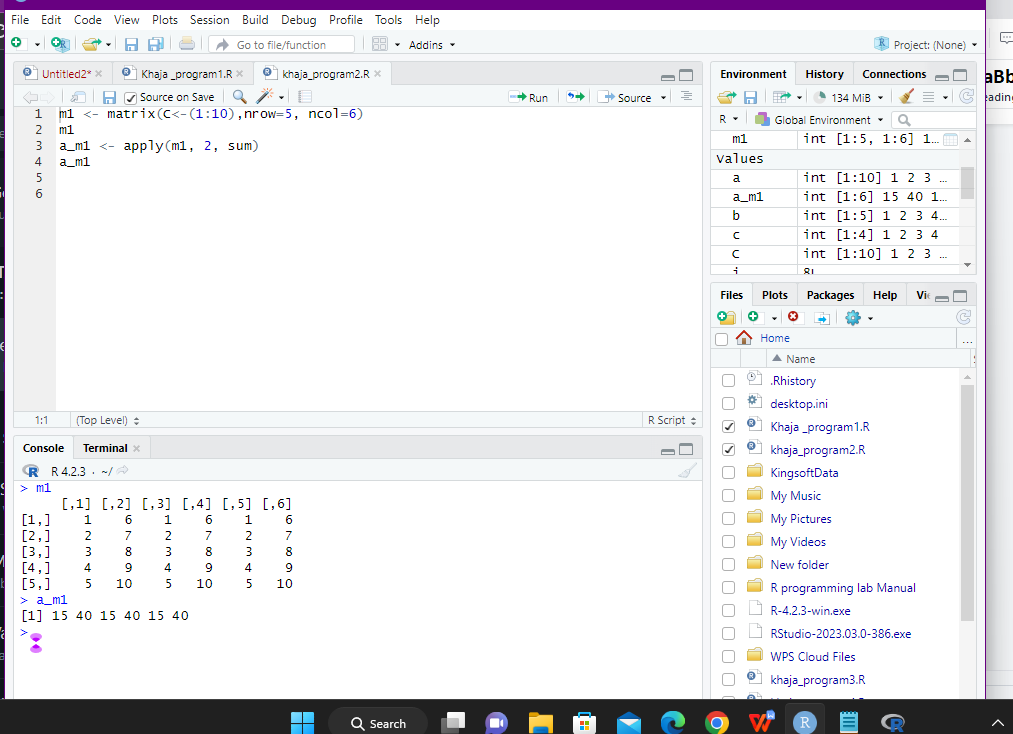
[4,] 4 9 4 9 4 9

[5,] 5 10 5 10 5 10

> a\_m1 <- apply(m1, 2, sum)

> a\_m1

1. 15 40 15 40 15 40



3. Demonstrate the usage of lapply function in R

SOURCE CODE:

> movies <- c("KATHTHI","MASTER","VIKRAM","INDIAN")

> movies\_lower <-lapply(movies, tolower)

> str(movies\_lower)

List of 4

$ : chr "kaththi"

$ : chr "master"

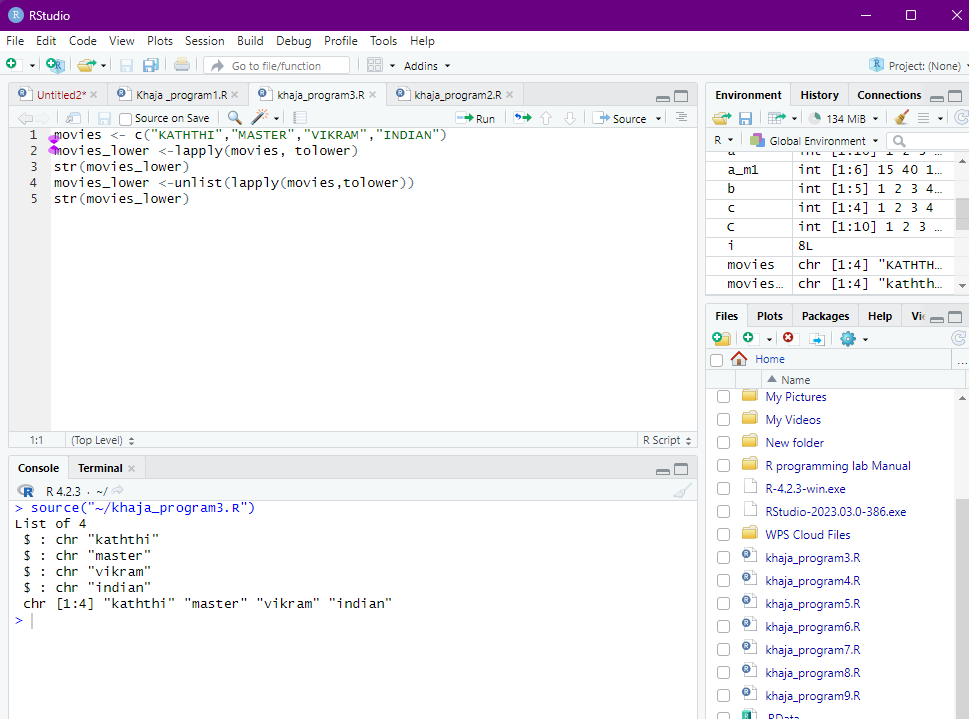
$ : chr "vikram"

$ : chr "indian"

> movies\_lower <-unlist(lapply(movies,tolower))

> str(movies\_lower)

chr [1:4] "kaththi" "master" …



4. Demonstrate the usage of sapply function in R

SOURCE CODE:

> dt <- cars

> lmn\_cars <- lapply(dt, min)

> smn\_cars <- sapply(dt, min)

> lmn\_cars

$speed

[1] 4

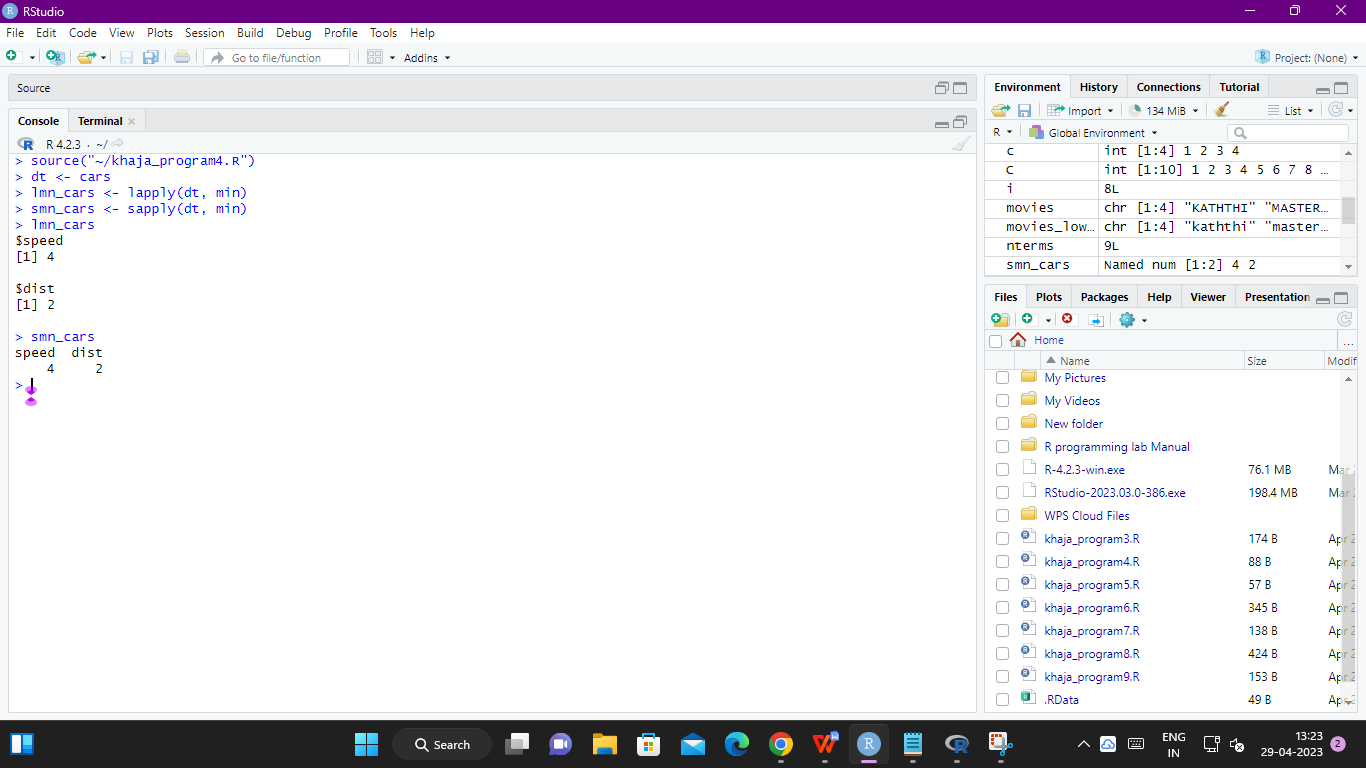
$dist

[1] 2

> smn\_cars

speed dist

4 2



5. Demonstrate the usage of tapply function in R

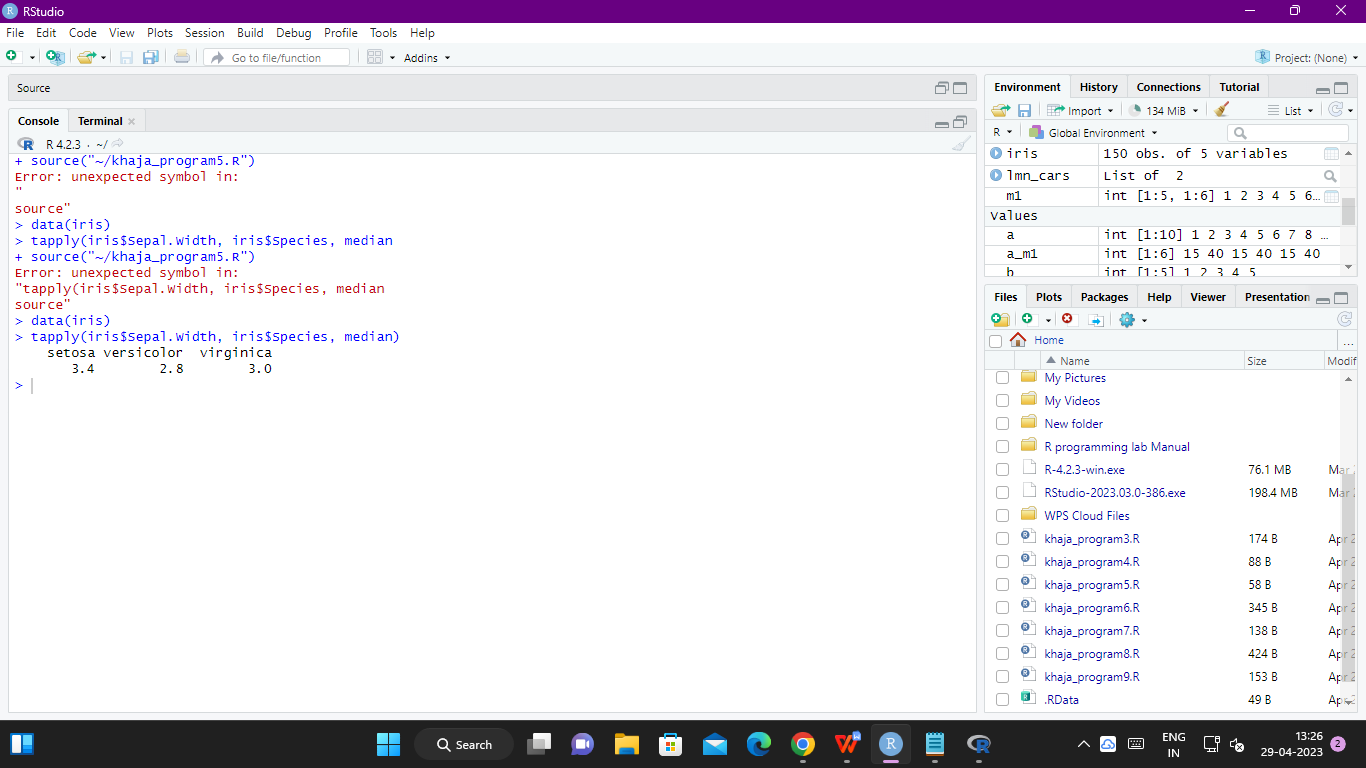
SOURCE CODE:

> data(iris)

> tapply(iris$Sepal.Width, iris$Species, median)

setosa versicolor virginica

3.4 2.8 3.0



6. Demonstrate the usage of mapply function in R

SOURCE CODE:

> vector1 <- c(1, 2, 3, 4, 5)

> vector2 <- c(2, 4, 1, 2, 10)

>

> #find max value of each corresponding elements in vectors

> mapply(max, vector1, vector2)

[1] 2 4 3 4 10

>

>

> vec1 <- c(1, 2, 3, 4)

> vec2 <- c(2, 4, 6, 8)

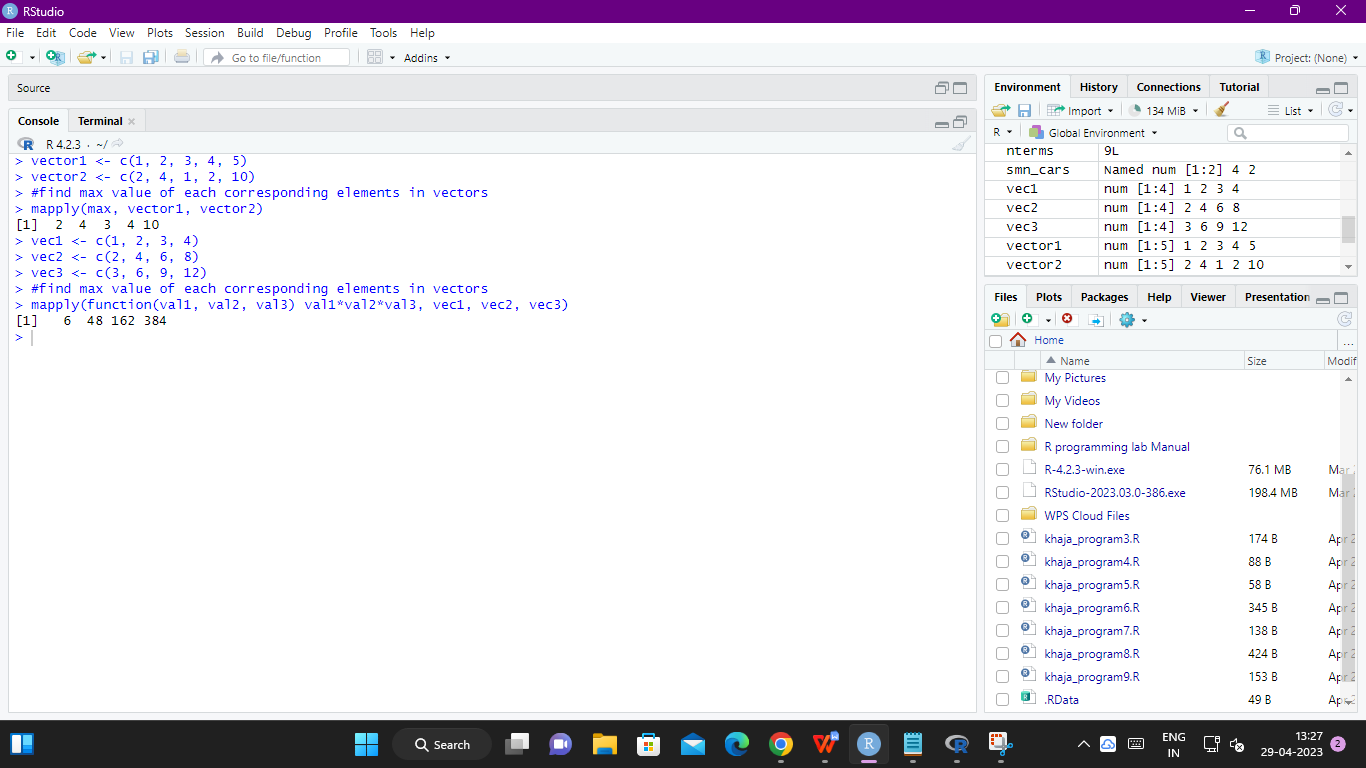
> vec3 <- c(3, 6, 9, 12)

>

> #find max value of each corresponding elements in vectors

> mapply(function(val1, val2, val3) val1\*val2\*val3, vec1, vec2, vec3)

[1] 6 48 162 384



7. Sum of Natural Numbers using Recursion

SOURCE CODE:

> recur\_sum <- function(n) {

+ if(n <= 1) {

+ return(n)

+ } else {

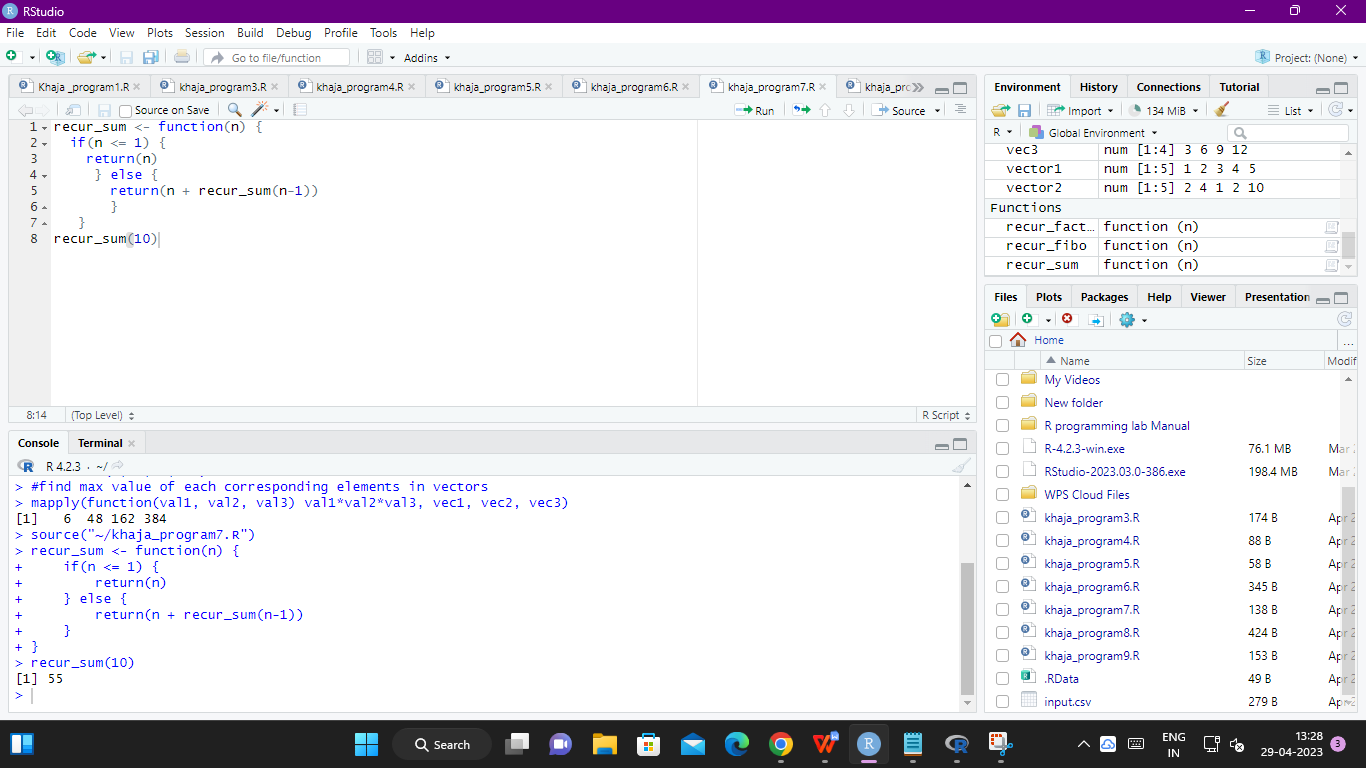
+ return(n + recur\_sum(n-1))

+ }

+ }

> recur\_sum(10)

[1] 55



8. Write a program to generate Fibonacci sequence using Recursion in R

SOURCE CODE:

recur\_fibo <- function(n) {

if(n <= 1) {

return(n)

} else {

return(recur\_fibo(n-1) + recur\_fibo(n-2))

}

}

# take input from the user

nterms = as.integer(readline(prompt="How many terms? "))

# check if the number of terms is valid

if(nterms <= 0) {

print("Plese enter a positive integer")

} else {

print("Fibonacci sequence:")

for(i in 0:(nterms-1)) {

print(recur\_fibo(i))

}

}

How many terms? 9

[1] "Fibonacci sequence:"

[1] 0

[1] 1

[1] 1

[1] 2

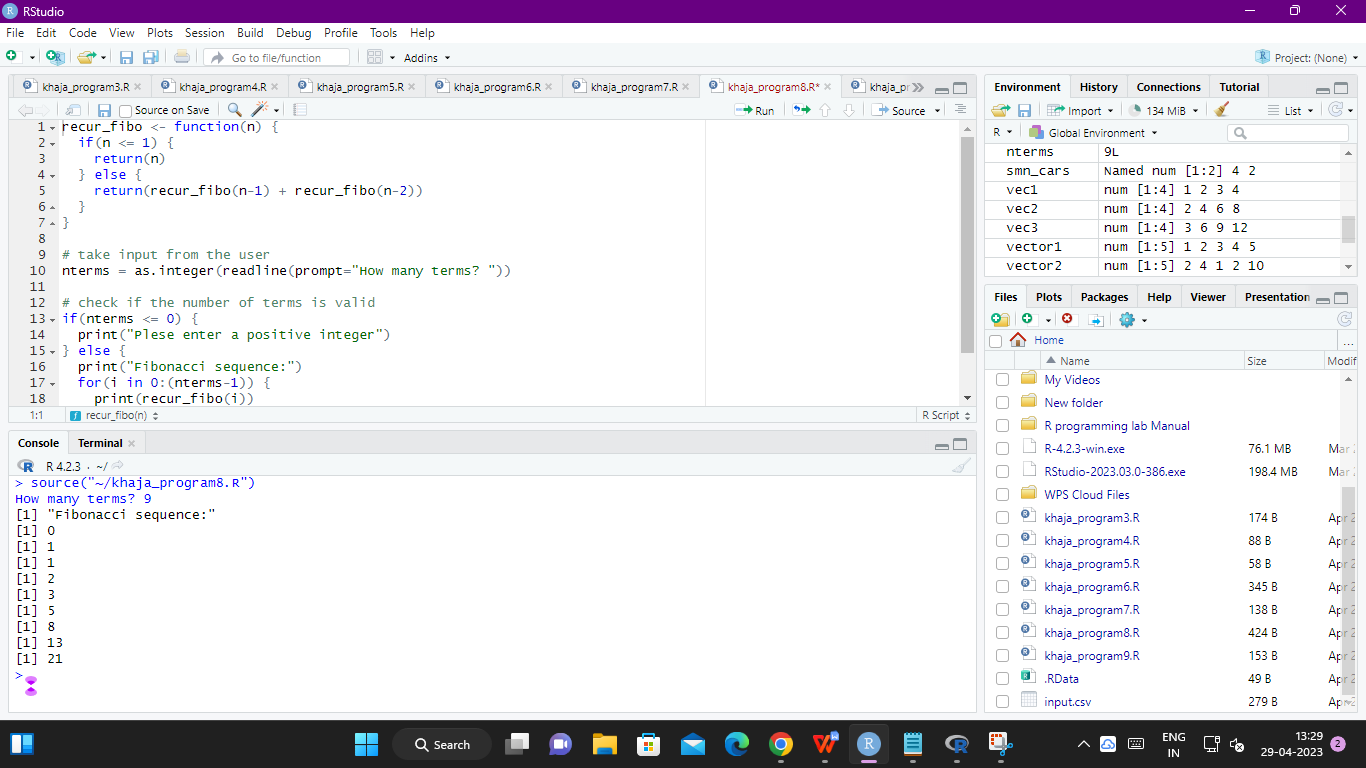
[1] 3

[1] 5

[1] 8

[1] 13

[1] 21



9. Write a program to find factorial of a number in R using recursion

SOURCE CODE:

> recur\_factorial <- function(n) {

+ if(n <= 1) {

+ return(1)

+ } else {

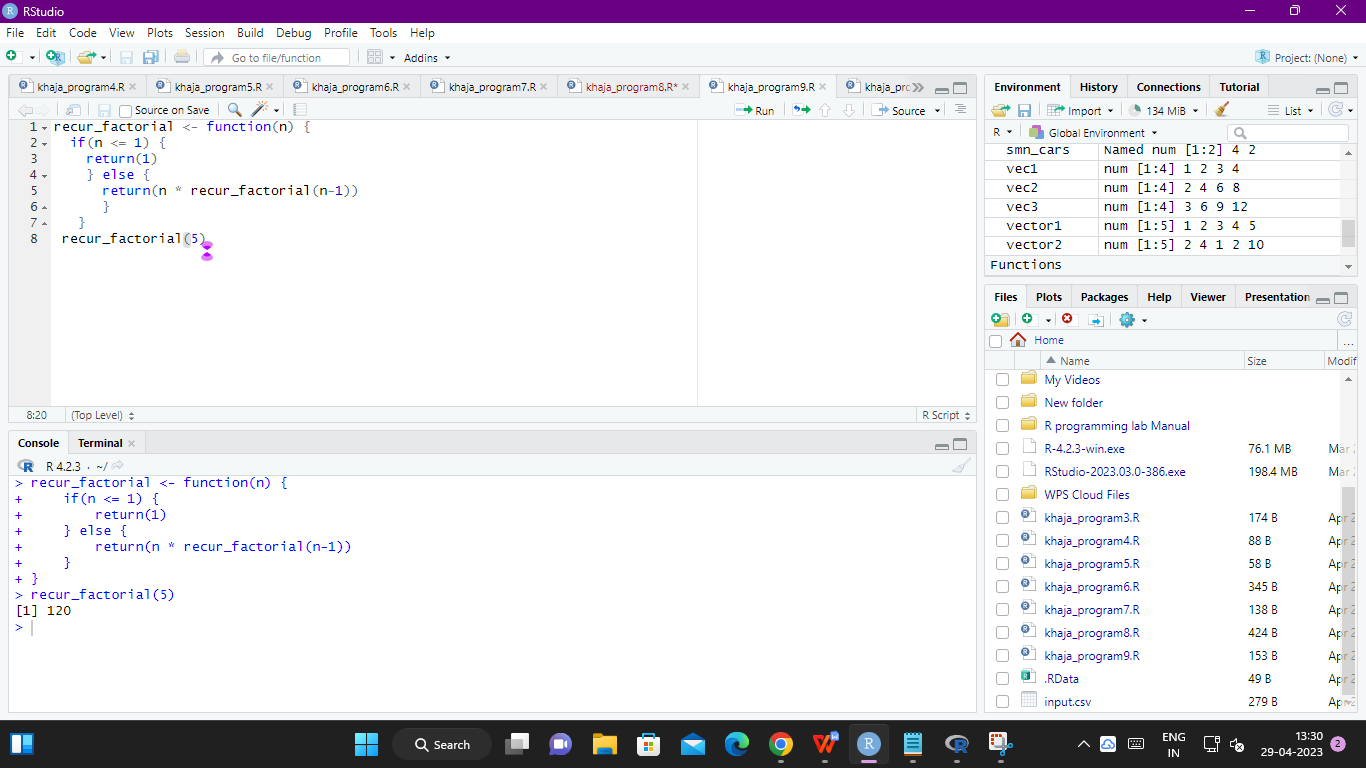
+ return(n \* recur\_factorial(n-1))

+ }

+ }

> recur\_factorial(5)

1. 120



**MELTING AND CASTING IN R**

1. Melt airquality data set and display as a long – format data ?

SOURCE CODE:

ibrary(reshape2)

# Melt the airquality data set

airquality\_long <- melt(airquality)

# Display the first few rows of the long-format data

head(airquality\_long)

1. Melt airquality data and specify month and day to be “ID variables” ?

Source code:

# Melt the airquality data set with Month and Day as ID variables

airquality\_long <- melt(airquality, id.vars = c("Month", "Day"))

# Display the first few rows of the long-format data

head(airquality\_long)

1. Cast the molten airquality data set .

SOURCE CODE:

# Cast the molten airquality data set and compute the average of Ozone, Solar.R, Wind, and Temperature per month

airquality\_wide <- dcast(airquality\_long, Month ~ variable, fun.aggregate = mean, value.var = c("Ozone", "Solar.R", "Wind", "Temp"))

# Display the wide-format data set

airquality\_wide

4. Use cast function appropriately and compute the average of Ozone, Solar.R , Wind

and temperature per month ?

SOURCE CODE:

airquality\_monthly <- cast(airquality, Month ~ ., mean, value=c("Ozone", "Solar.R", "Wind", "Temp"))

**FILE MANIPILATION IN R**

**1. Consider the following data present. Create this file using windows notepad . Save the**

**file as input.csv using the save As All files(\*.\*) option in notepad.**

**> data <- read.csv("input.csv")**

**> print(data)**

**id name salary start\_date dept**

**1 1 Rick 623.30 2012-01-01 IT**

**2 2 Dan 515.20 2013-09-23 Operations**

**3 3 Michelle 611.00 2014-11-15 IT**

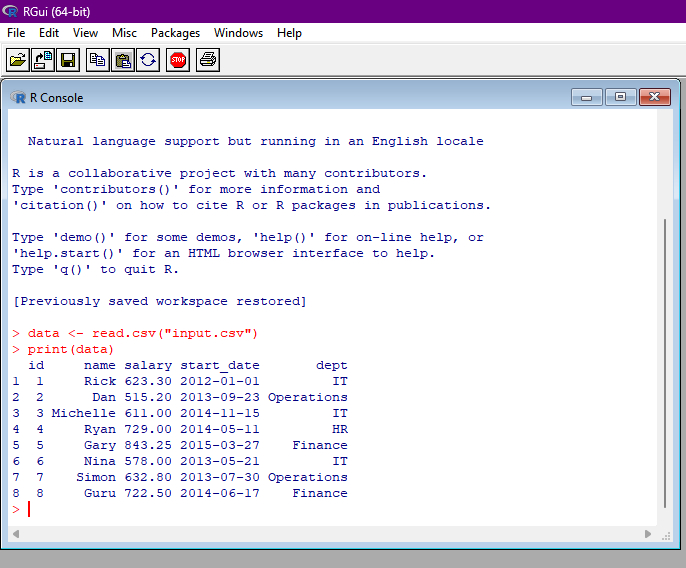
**4 4 Ryan 729.00 2014-05-11 HR**

**5 5 Gary 843.25 2015-03-27 Finance**

**6 6 Nina 578.00 2013-05-21 IT**

**7 7 Simon 632.80 2013-07-30 Operations**

**8 8 Guru 722.50 2014-06-17 Finance**



1. **Use appropriate R commands to read input.csv file.**

**> data <- read.csv("input.csv")**

**>**

**> print(is.data.frame(data))**

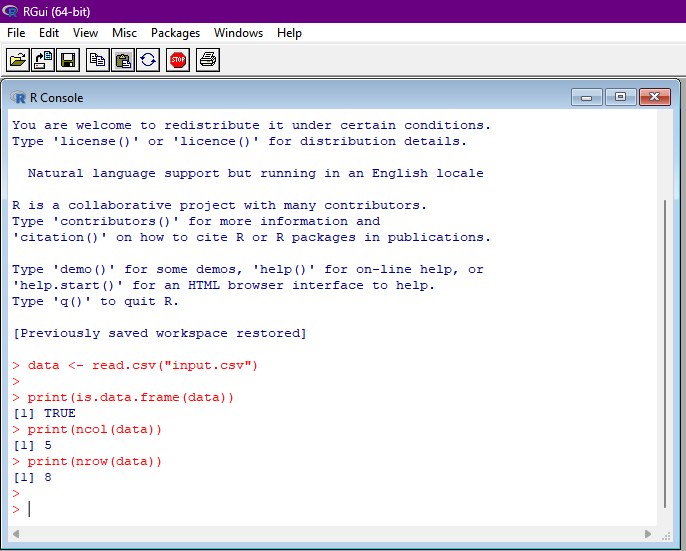
**[1] TRUE**

**> print(ncol(data))**

**[1] 5**

**> print(nrow(data))**

**[1] 8**



**3. Analyze the CSV File and compute the following.**

**a. Get the maximum salary**

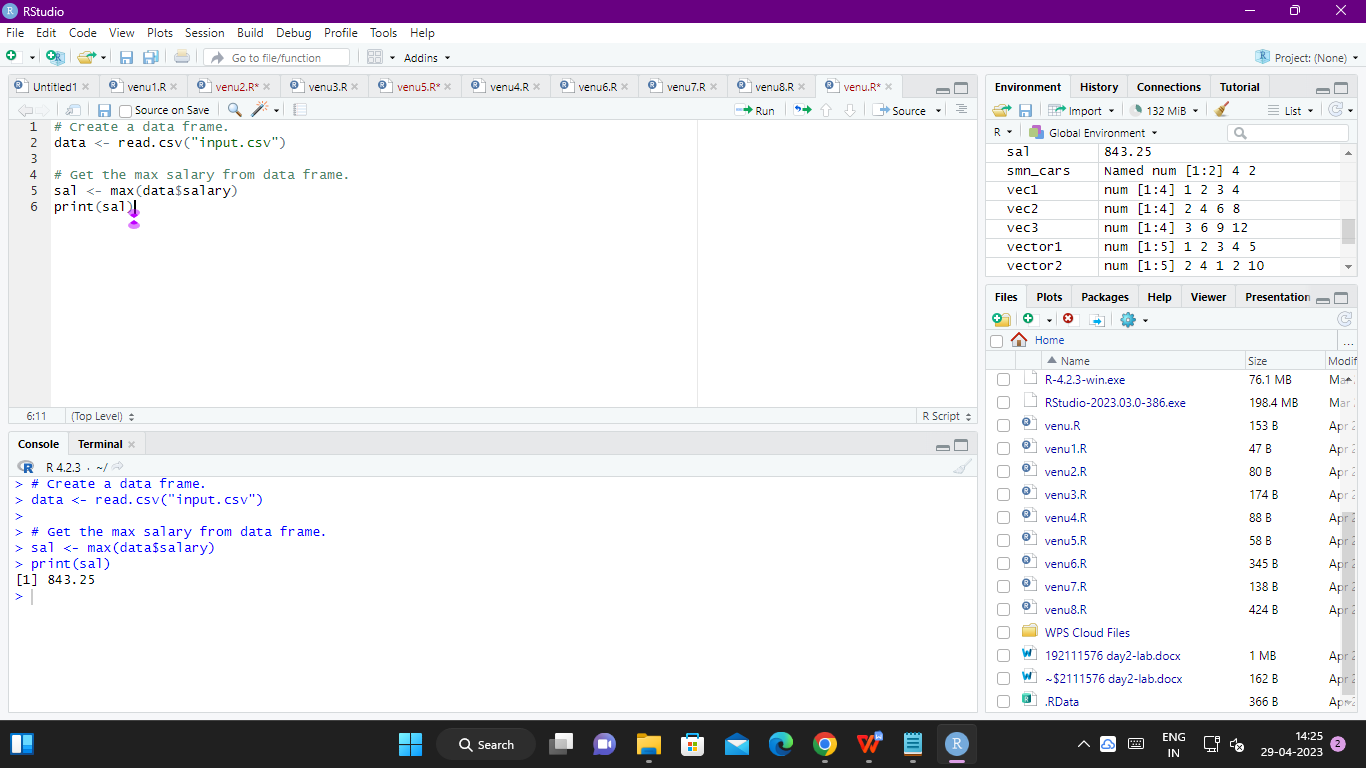
**#Create a data frame.**

**data <- read.csv("input.csv")**

**# Get the max salary from data frame.**

**sal <- max(data$salary)**

**print(sal)**



**b. Get the details of the person with max salary**

**data <- read.csv("input.csv")**

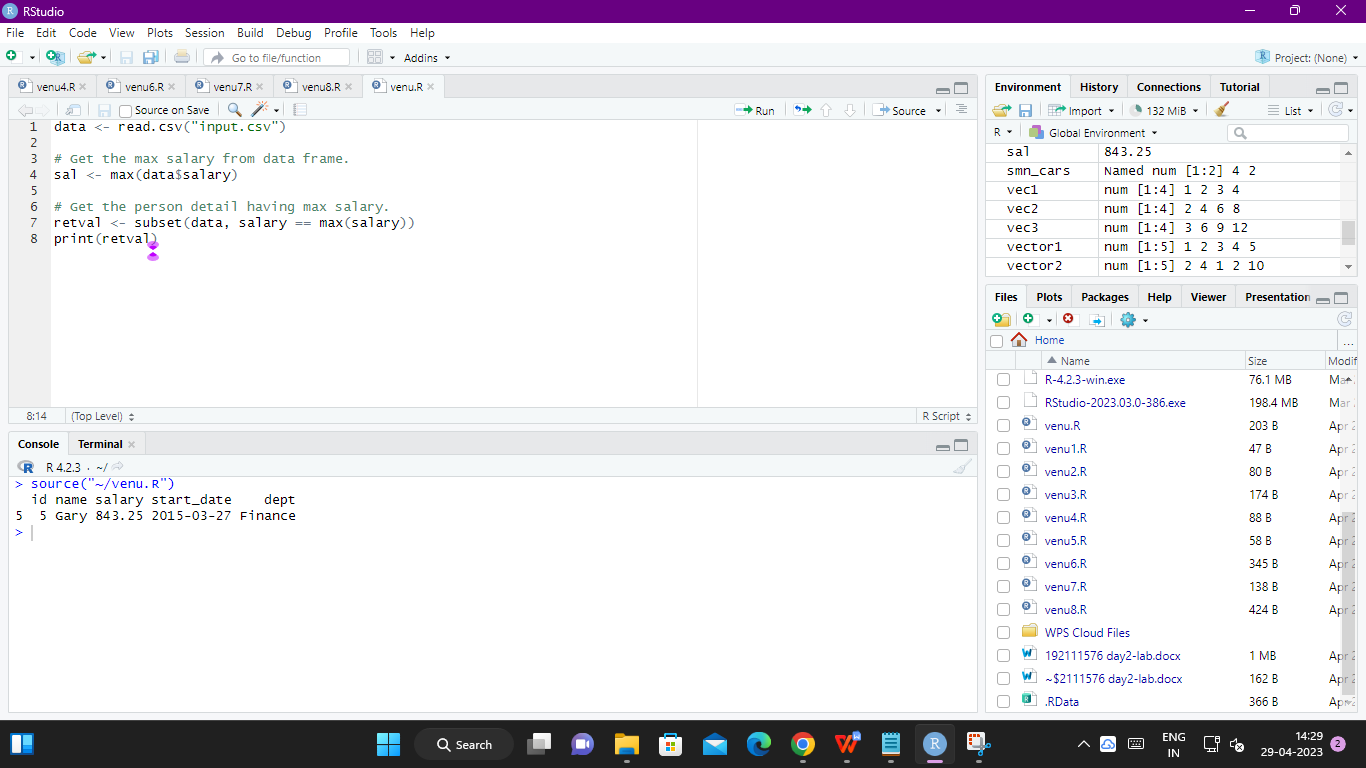
**# Get the max salary from data frame.**

**sal <- max(data$salary)**

**# Get the person detail having max salary.**

**retval <- subset(data, salary == max(salary))**

**print(retval)**

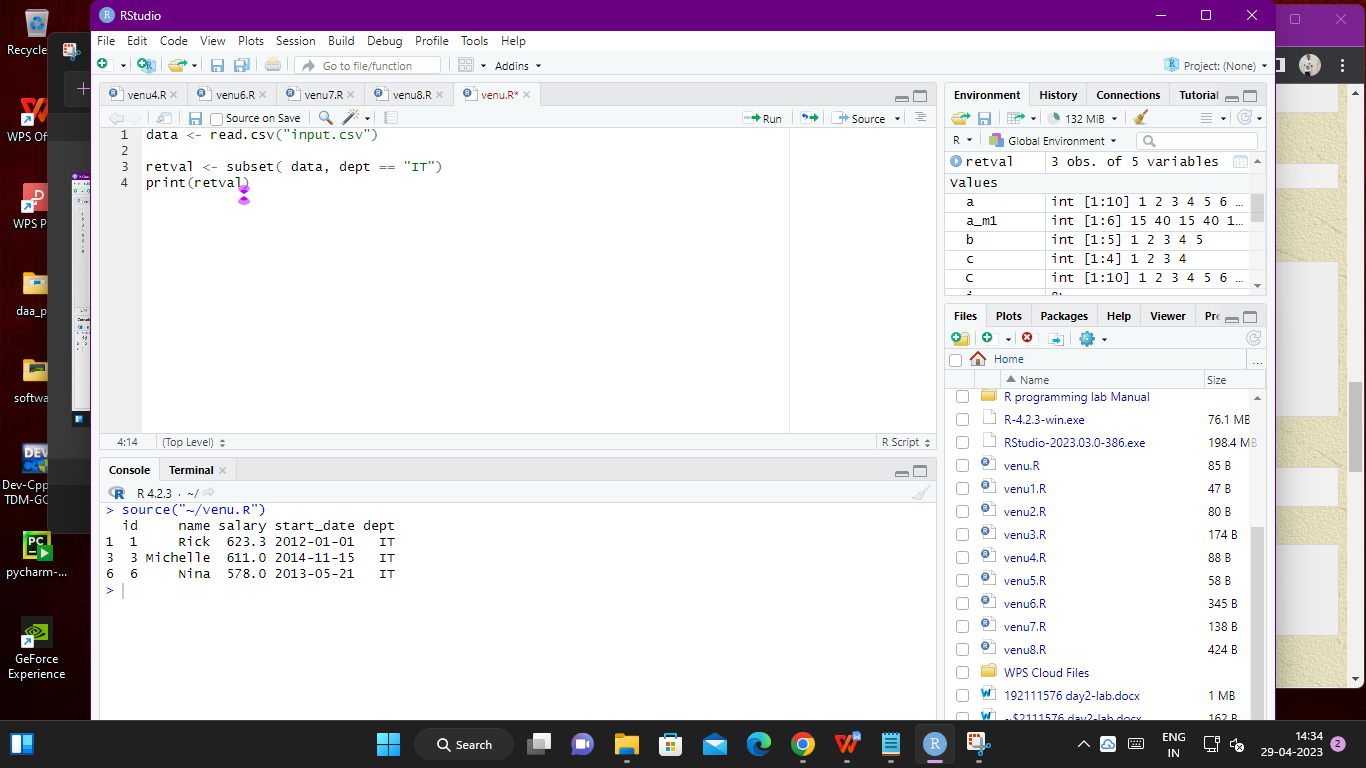


**c. Get all the people working in IT department**

**data <- read.csv("input.csv")**

**retval <- subset( data, dept == "IT")**

**print(retval)**

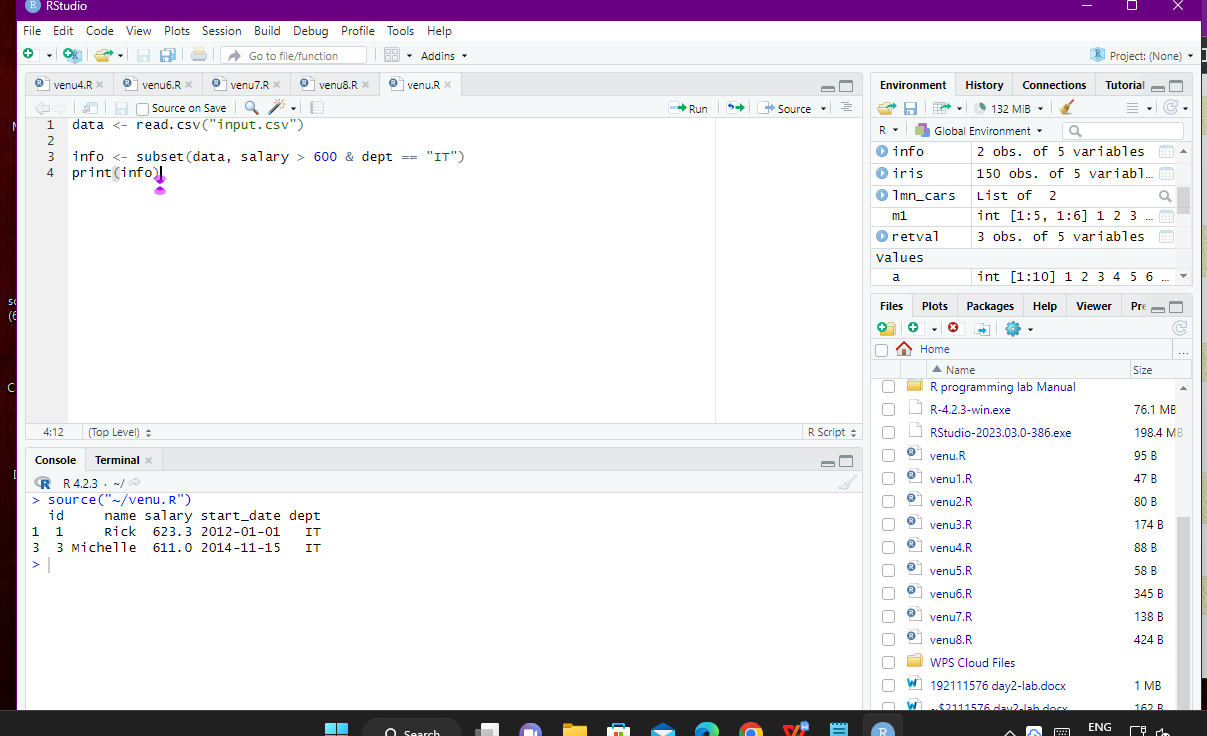


**d. Get the persons in IT department whose salary is greater than 600**

**data <- read.csv("input.csv")**

**info <- subset(data, salary > 600 & dept == "IT")**

**print(info)**

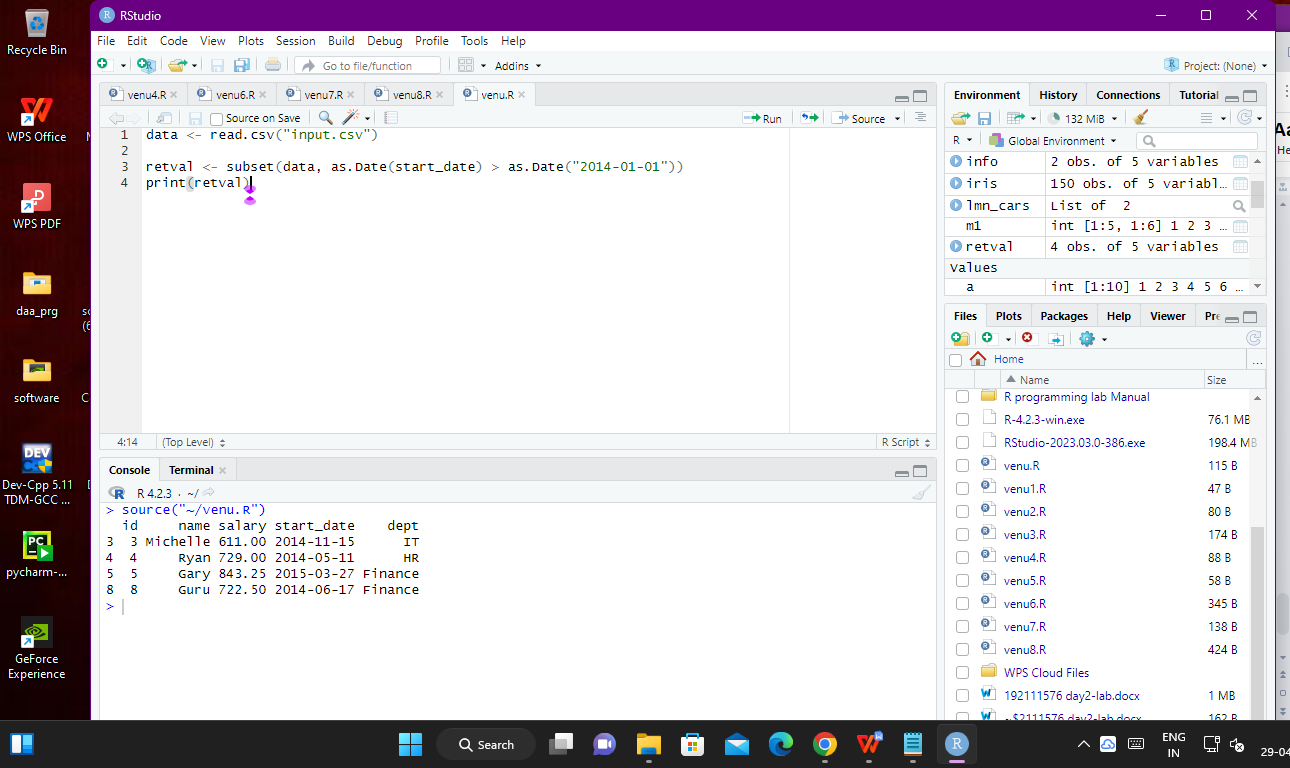


**e. Get the people who joined on or after 2014**

**data <- read.csv("input.csv")**

**retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))**

**print(retval)**



**4. Get the people who joined on or after 2014 and write the output onto a file called**

**output.csv**

**data <- read.csv("input.csv")**

**retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))**

**# Write filtered data into a new file.**

**write.csv(retval,"output.csv", row.names = FALSE)**

**newdata <- read.csv("output.csv")**

**print(newdata)**

