**Brief of R programming, exploring data in R and plotting graphs**

1. What is the difference between package and library? (5 points)

A **package** includes functions, data and belongs to one library whereas a **library** is a directory that contains a set of packages. Also library function() is used to load package.

For example, loading xlsx file**.**

install.packages(“xlsx”)

library(xlsx)

lxl <- read.xlsx(“datafile.xlsx”, 1)

1. Assume there is a CSV file where has header and separated using semicolon. Please write down the functions to load this file with header. Assume there is a xlsx file, write down the functions to load the first sheet of this xlsx file with header. (10 points)

**Loading csv file ‘datafile.csv”:**

lcsv <- read.csv(“datafile.csv”, sep=”;”, header= TRUE)

**Loading xlsx file :**

install.packages(“xlsx”)

library(xlsx)

lxl <- read.xlsx(“datafile.xlsx”, 1)

1. Please create two vectors. One is a vector of 100 normal random variables with mean 5 and standard deviation 2.Another is a vector of 100 normal random variables with mean 15 and standard deviation 3. Please use three different functions to create a scatter plot with two vectors, and write down all functions. Then use three different functions to create a line graph with two vectors, and write down all functions. (20 points)

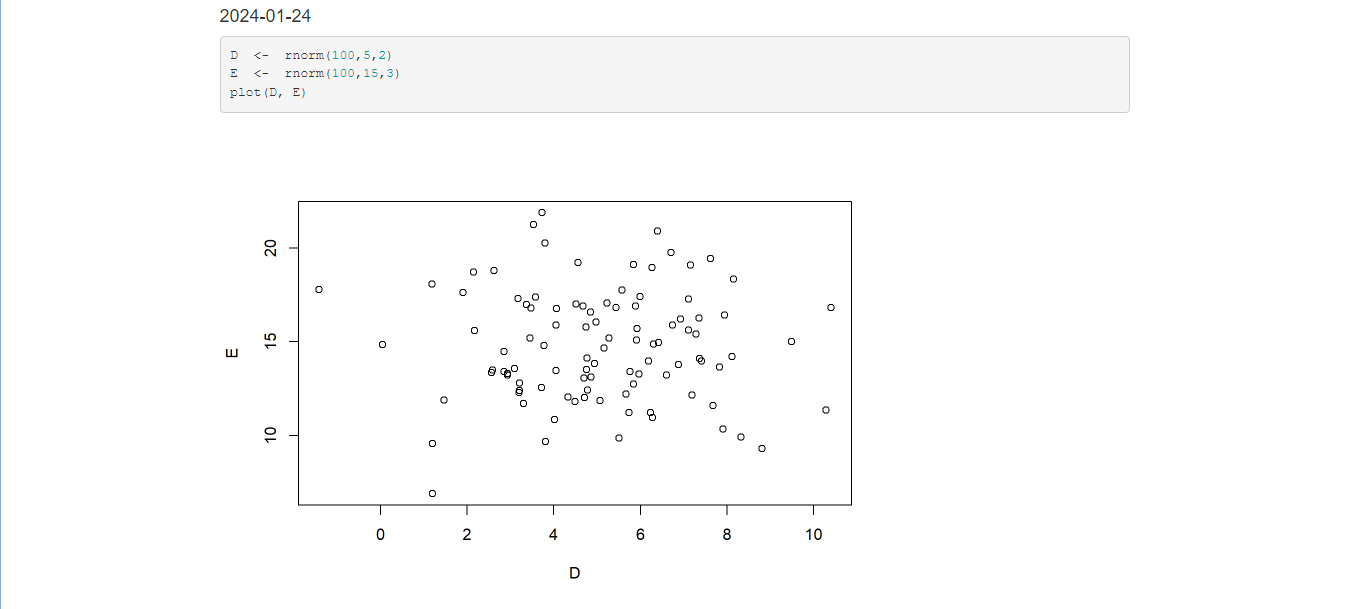
D <- rnorm(100,5,2)

E <- rnorm(100,15,3)

Scatter plot :

*Type-1*:

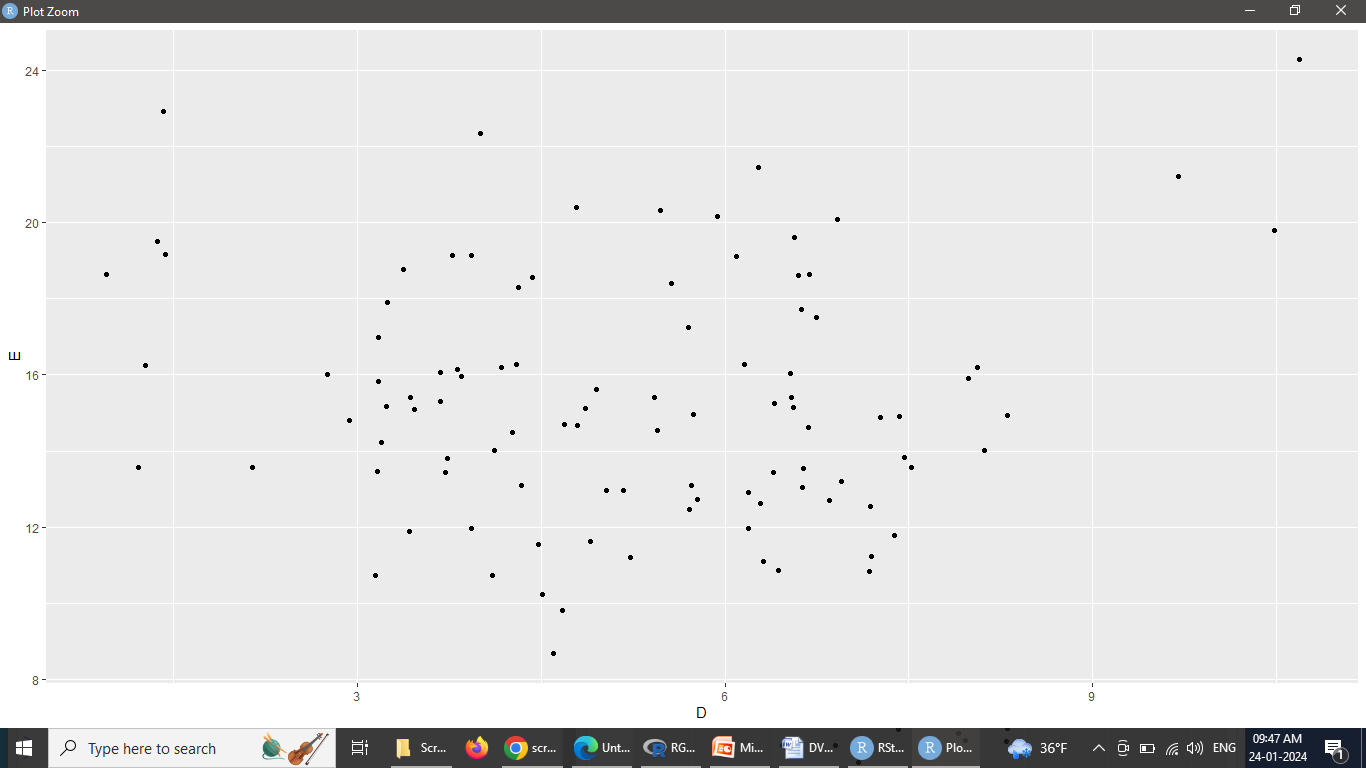
plot(D, E)



*Type-2*:

library(ggplot2)

qplot(D,E)



*Type-3*: If D and E are the vectors in the data frame ‘F’ then the scatter plot is as below:

qplot(D, E, data = F)

ggplot(F, aes(x=D, y=E)) + geom\_point()

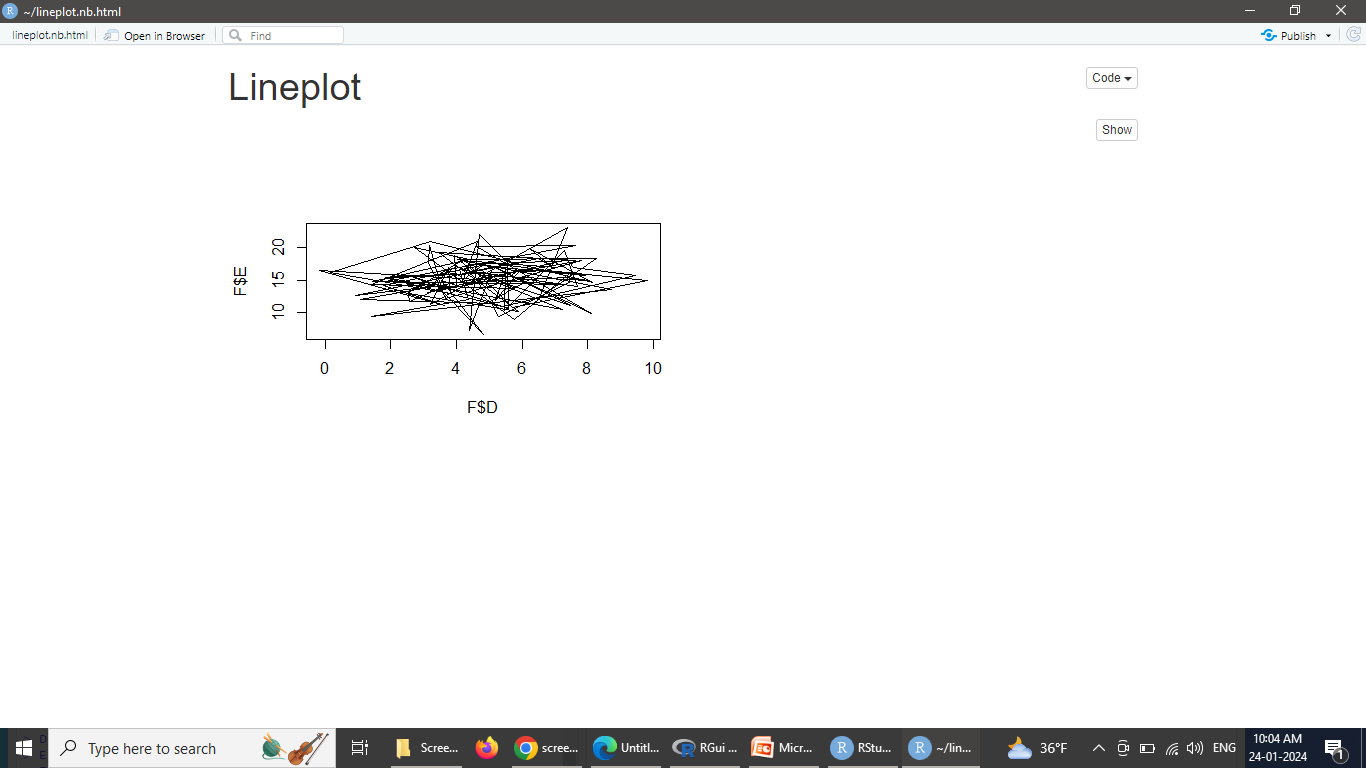
ggplot(NULL,aes(x=D, y=E)) + geom\_point()

F<- data.frame(D,E)

#***Creating a Line Graph using plot() :***

* + Use plot(), pass it a vector of x values and a vector of y values, and use type="l“

plot(F$D, F$E,type="l")

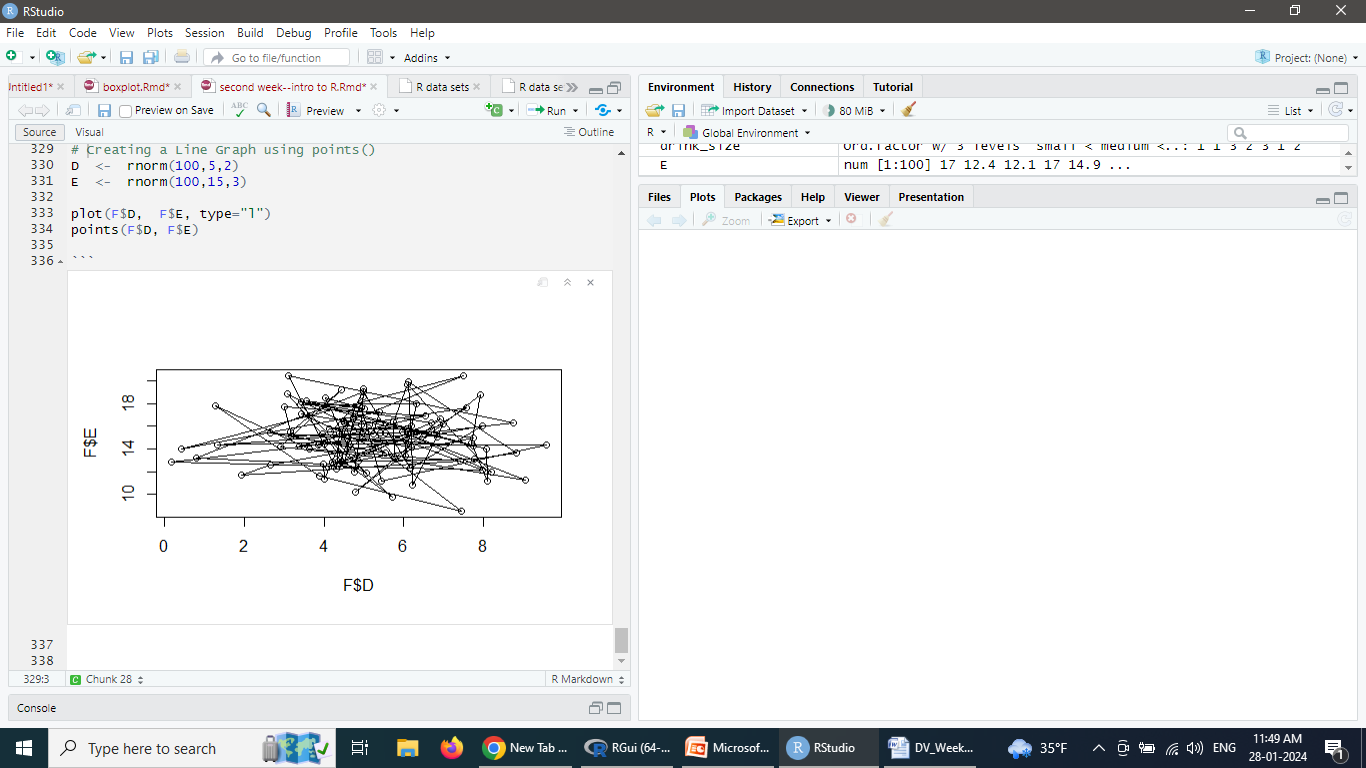


* + To add points and/or multiple lines,
    - first call plot() for the first line,
    - then add points with points() and additional lines with lines():

#***Creating a Line Graph using points()***

plot(F$D, F$E, type="l")

points(F$D, F$E)

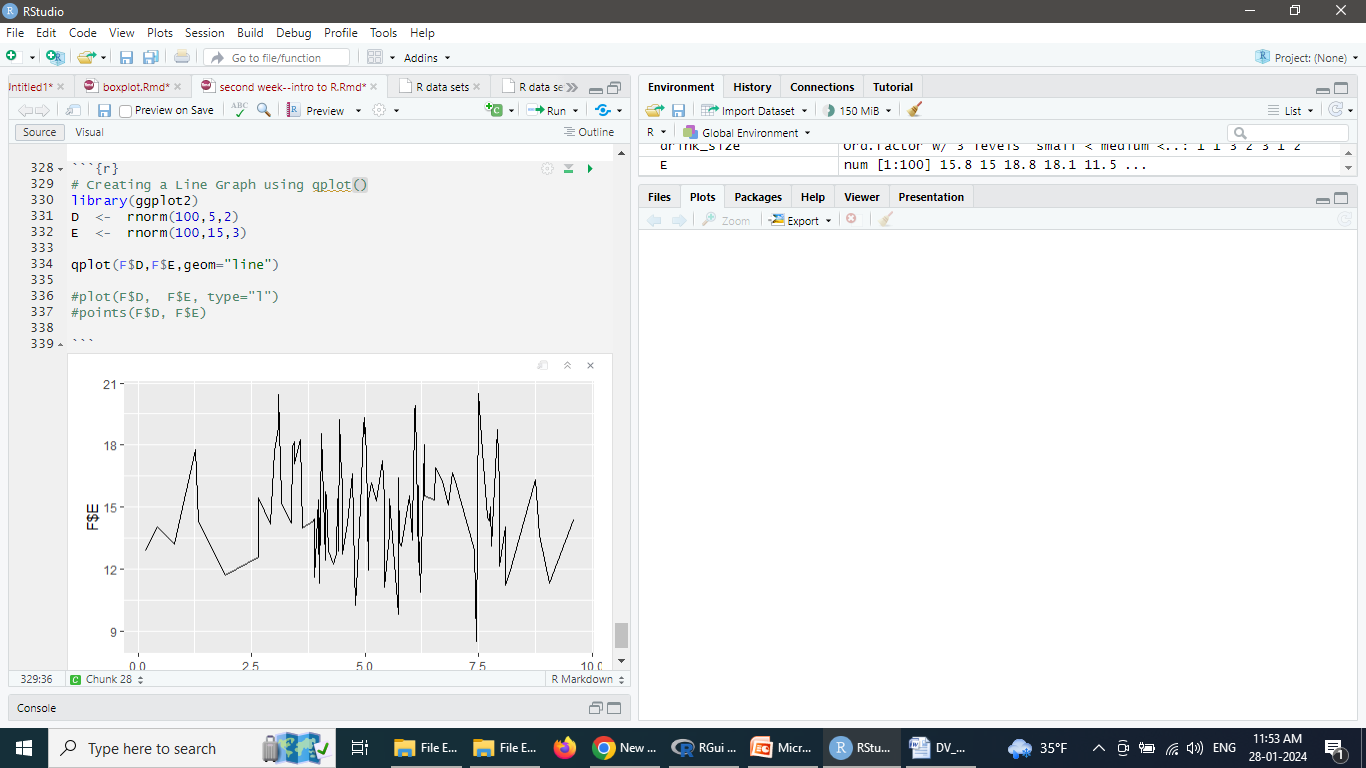


#***Line graph using qplot()***

* In ggplot2, use qplot() with geom="line“

library(ggplot2)

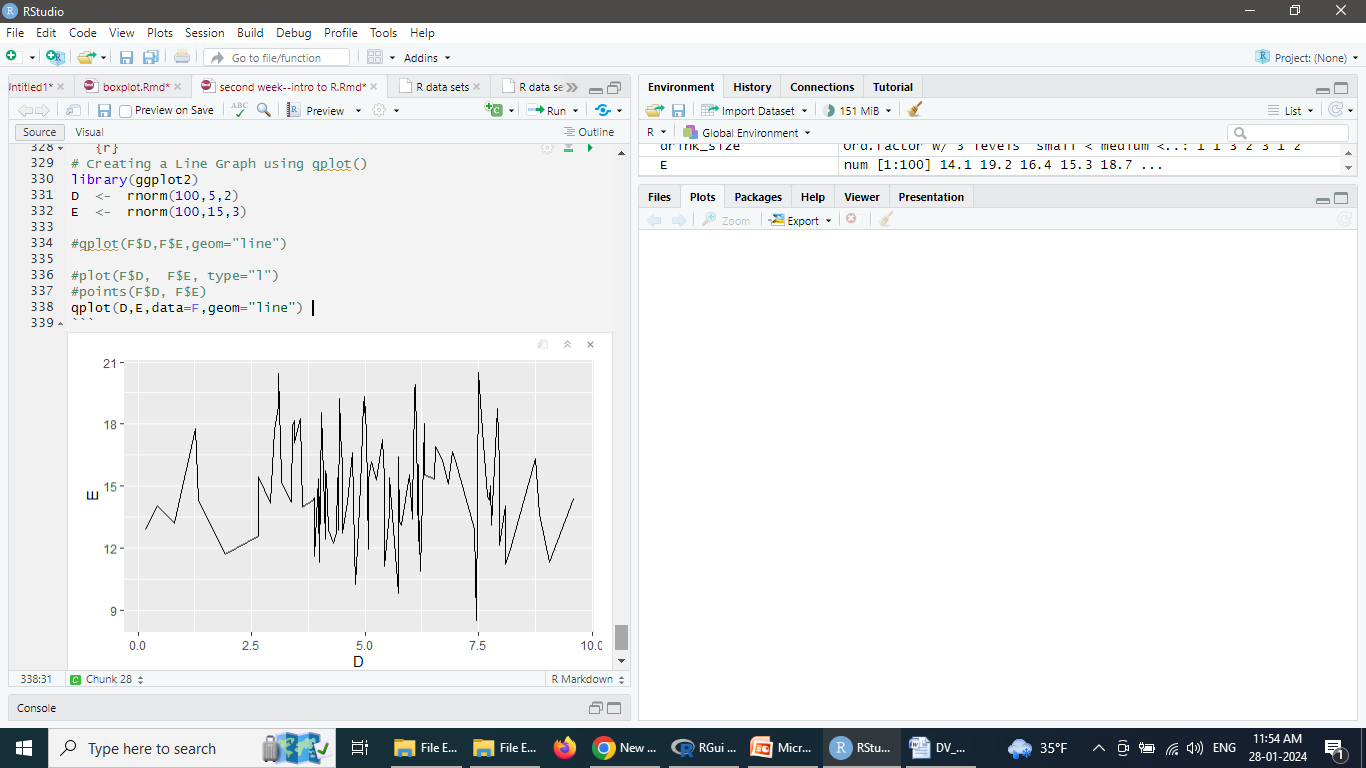
qplot(F$D,F$E,geom="line")



# If the two vectors are already in the same data frame

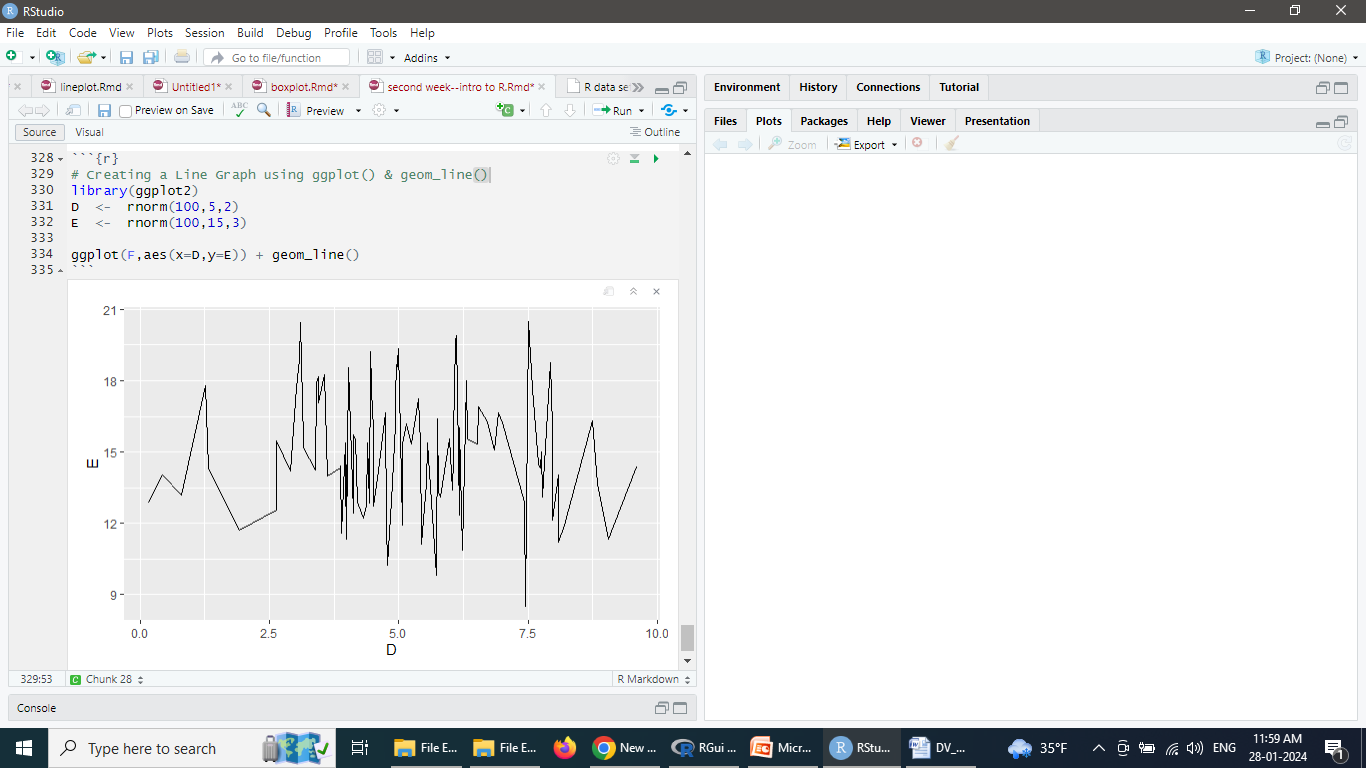
#Type1

qplot(D,E,data=F,geom="line")



#Type2

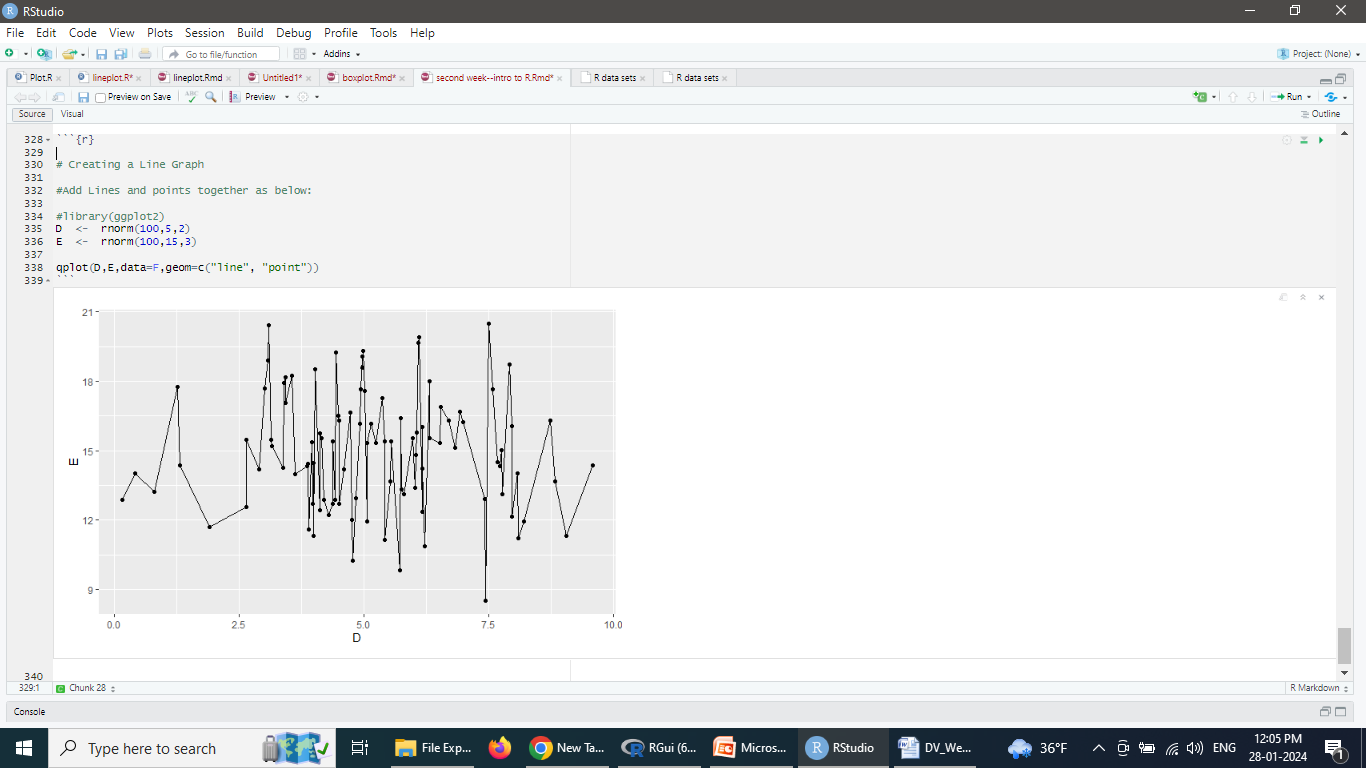
ggplot(F,aes(x=D,y=E)) + geom\_line()



#Add Lines and points together as below:

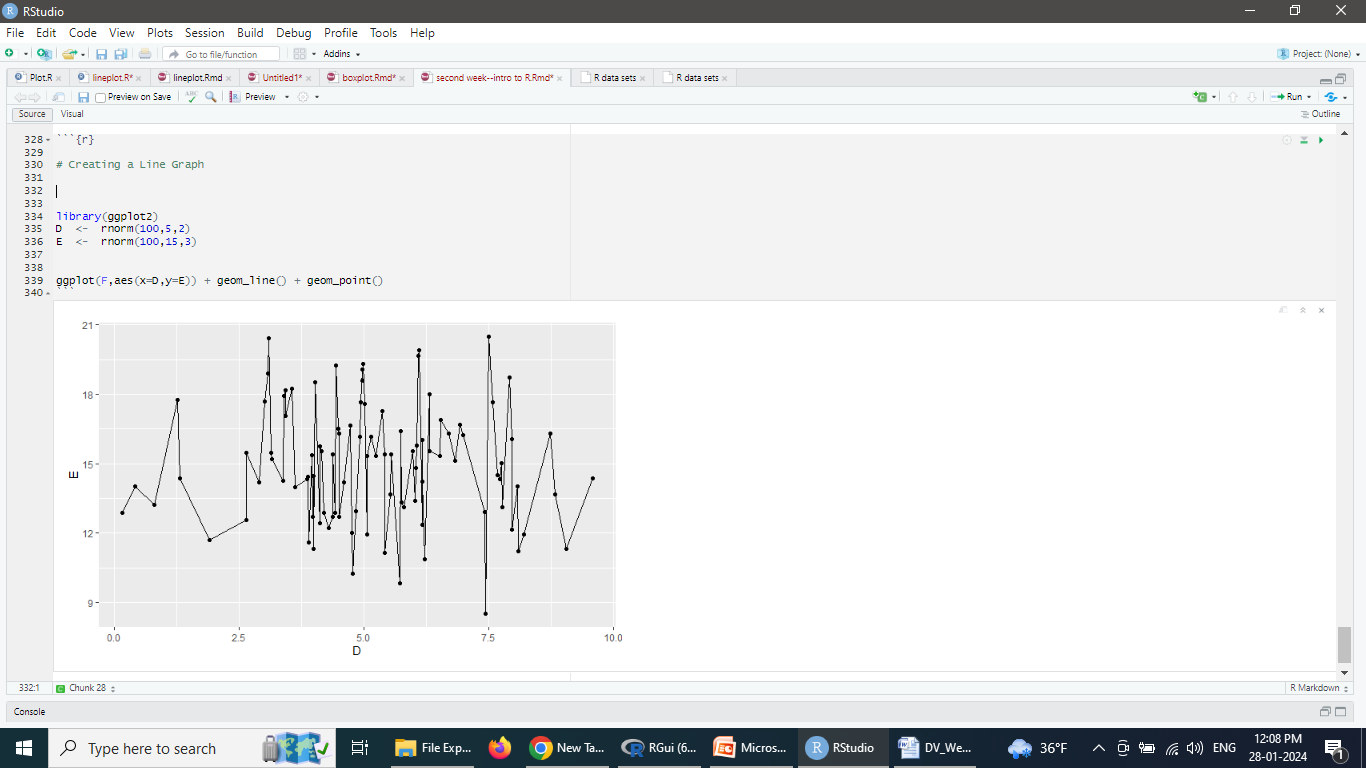
#Type1

qplot(D,E,data=F,geom=c("line", "point"))



#Type2

ggplot(F,aes(x=D,y=E)) + geom\_line() + geom\_point()

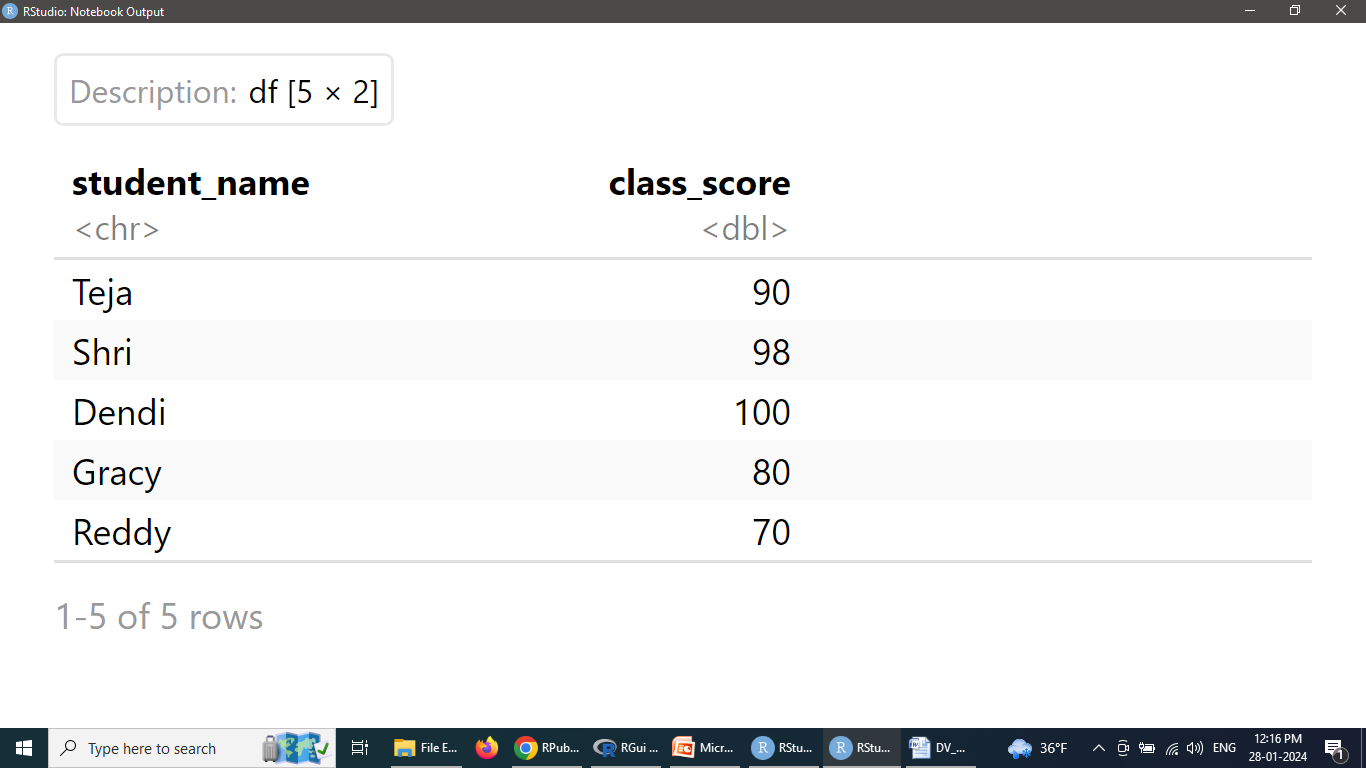


1. Please create a data frame. There are two columns in the frame: one is students’ name and another is class score. There are 5 students totally. (5 points)

student\_ name <- c(“Teja”, “Shri”, “Dendi”, “Gracy”, “Reddy”)

class\_score <- c(90, 98, 100, 80, 70)

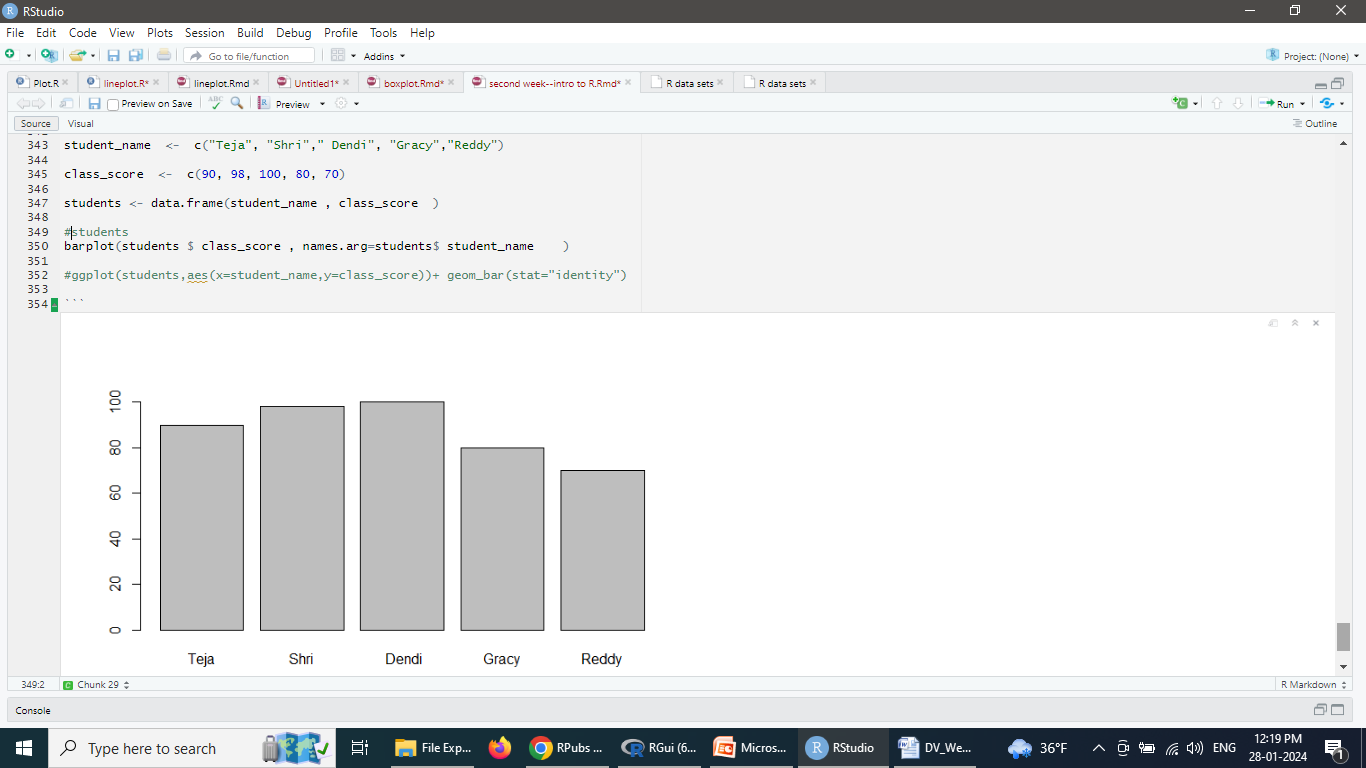
students <- data.frame(student\_ name , class\_score )



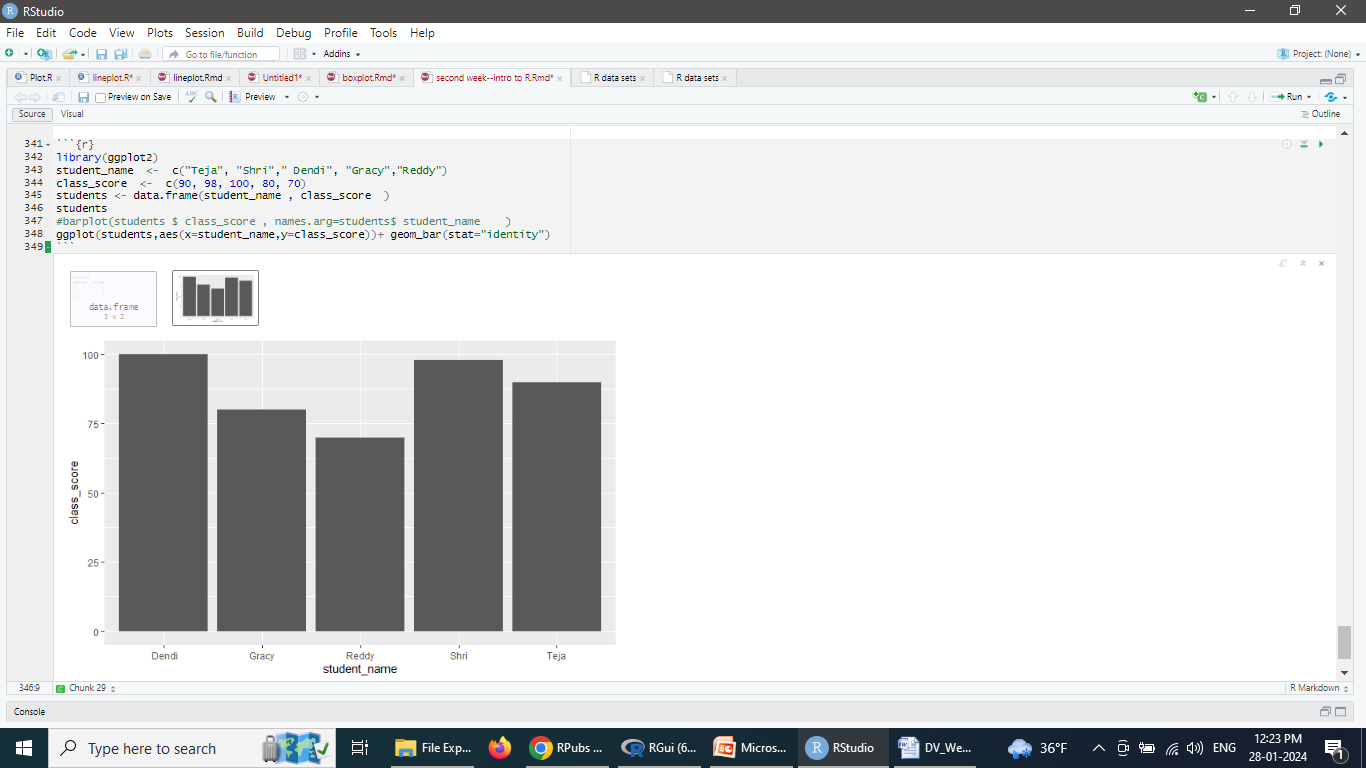
1. Using the above data frame to create a bar graph using two ways. (10 points)

#***Creating a Bar Graph***

barplot(students $ class\_score , names.arg=students$ student\_ name )



ggplot(students,aes(x=student\_name,y=class\_score))+ geom\_bar(stat="identity")

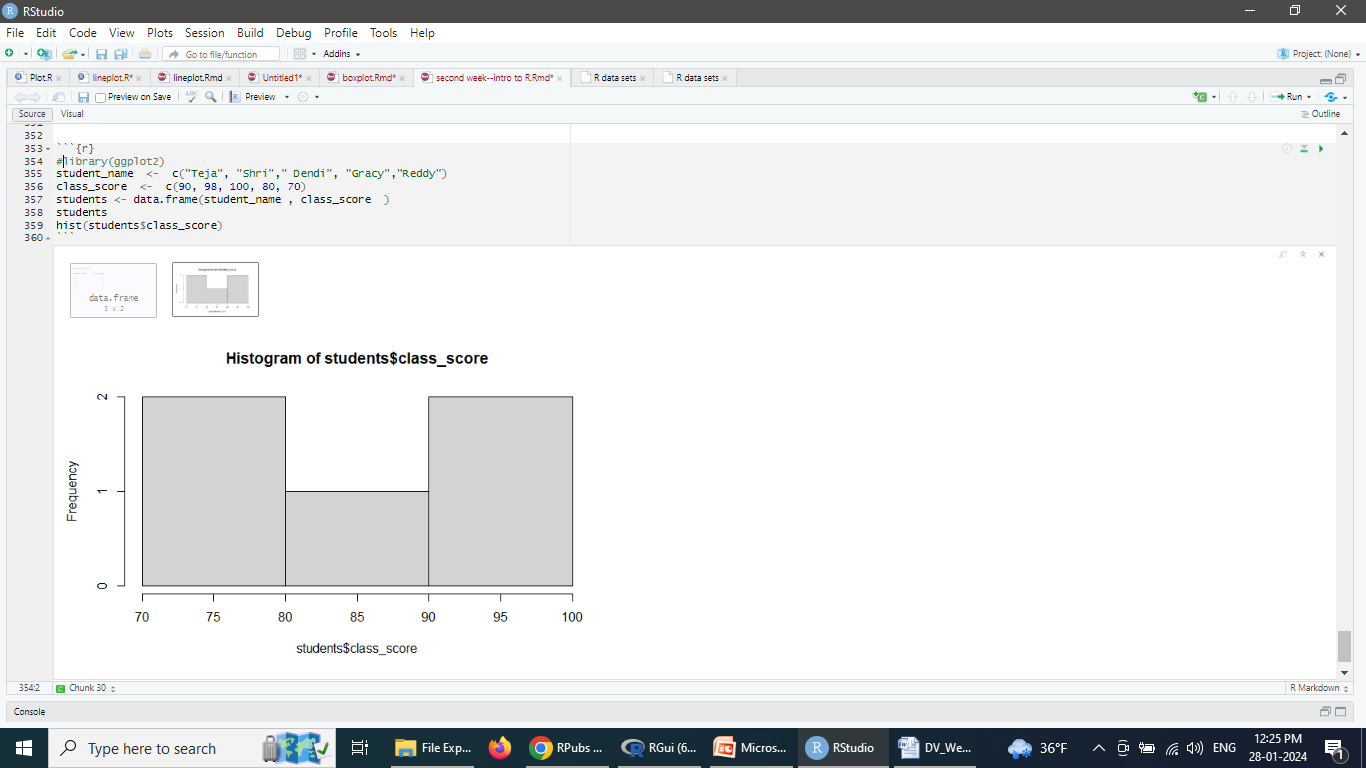


1. Using the above data frame to create a histogram using three ways. (15 points)

#Type1

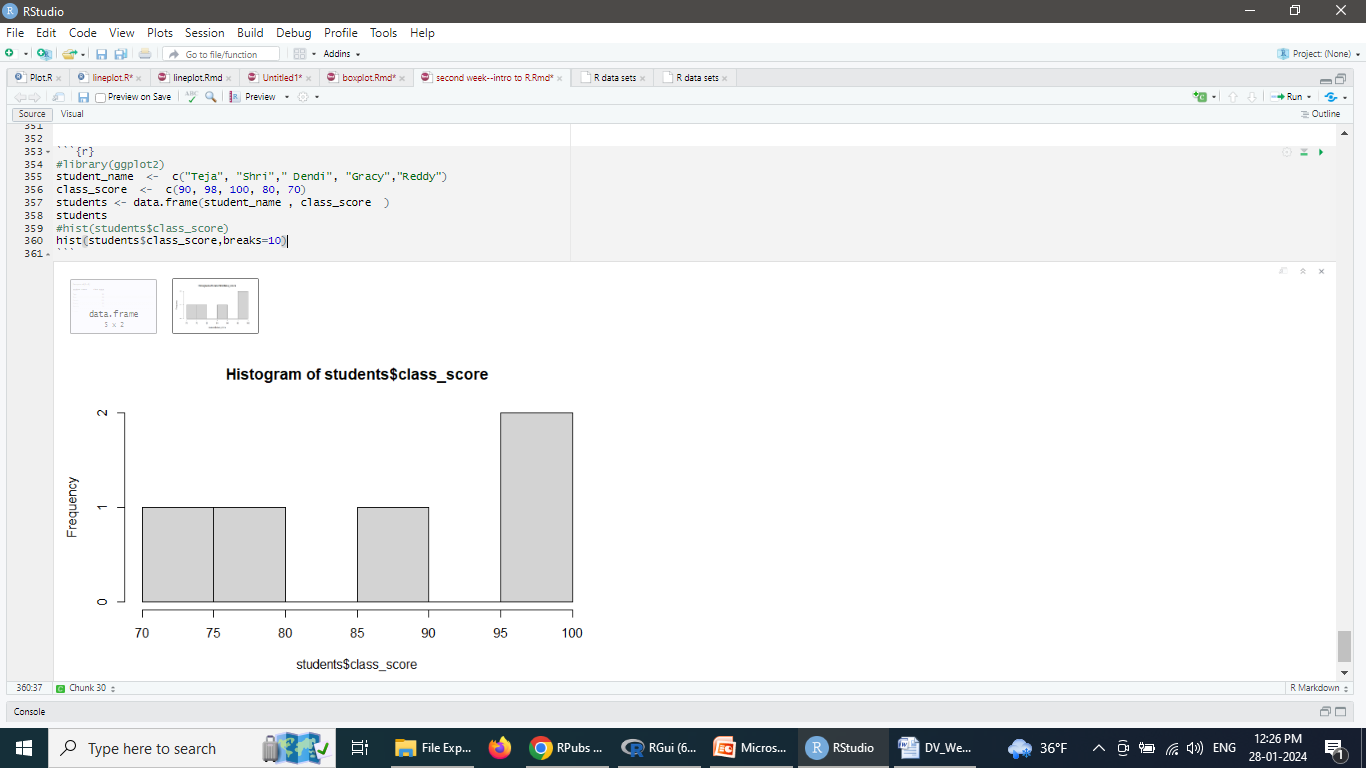
#To make a histogram , use hist() and pass it a vector of values:

hist(students$class\_score)



# Specify approximate number of bins with breaks

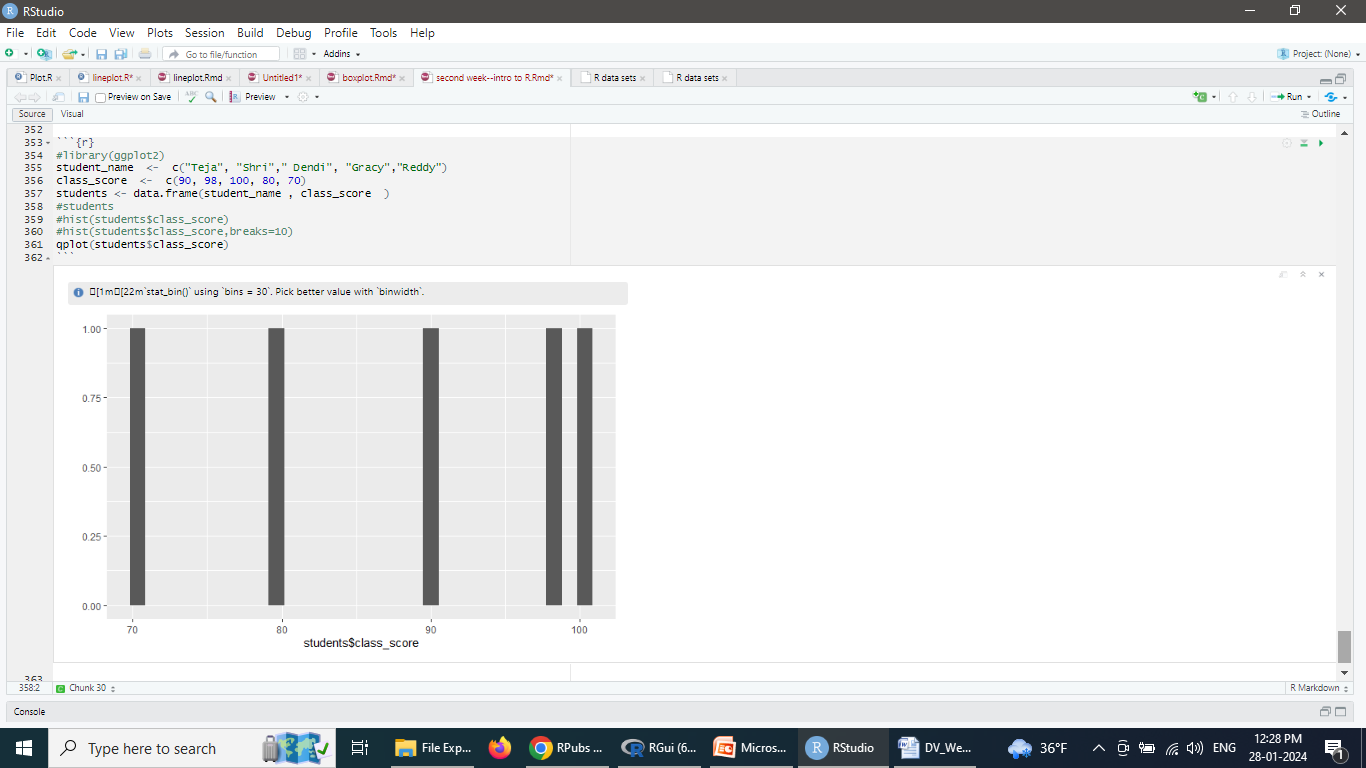
hist(students$class\_score,breaks=10)



#Type2

#With the ggplot2 package, use qplot()

qplot(students$class\_score)

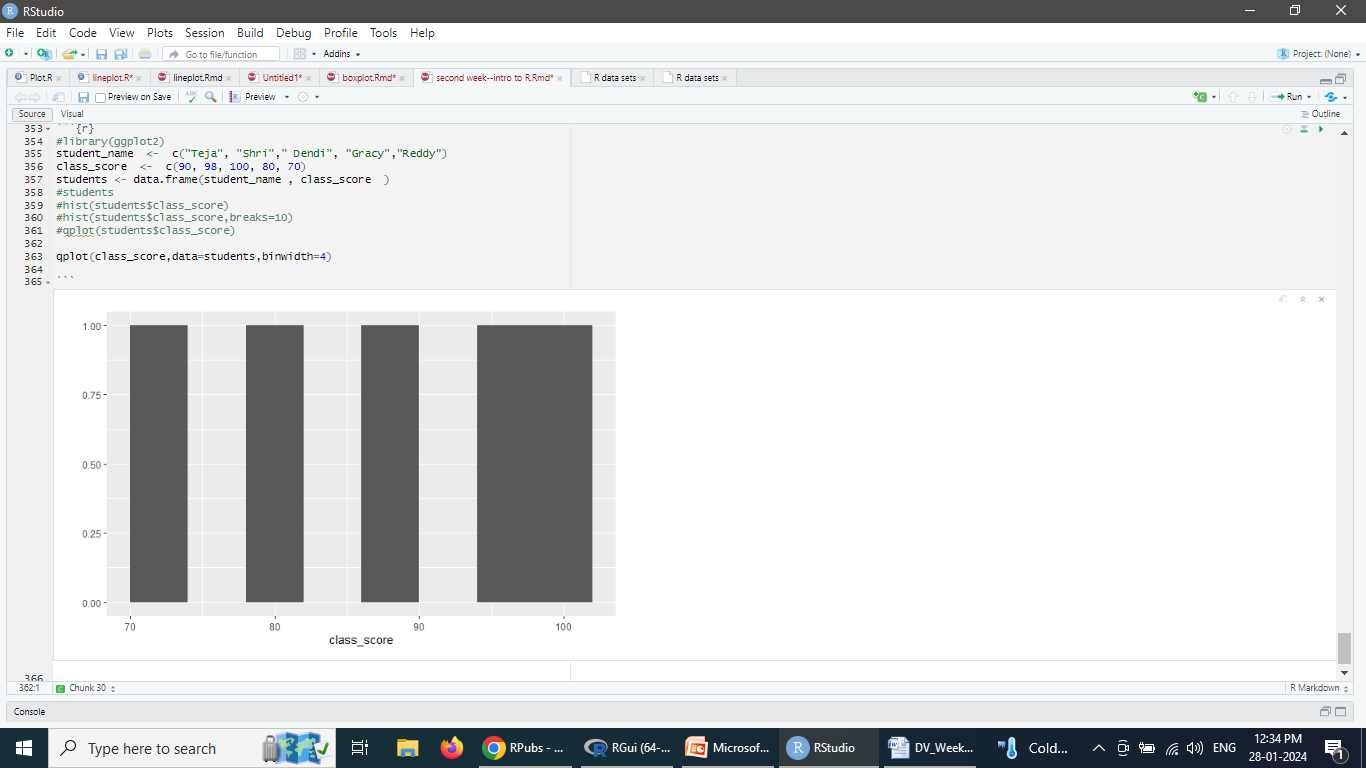


#Type3

#If the vector is in a data frame, you can use the following syntax:

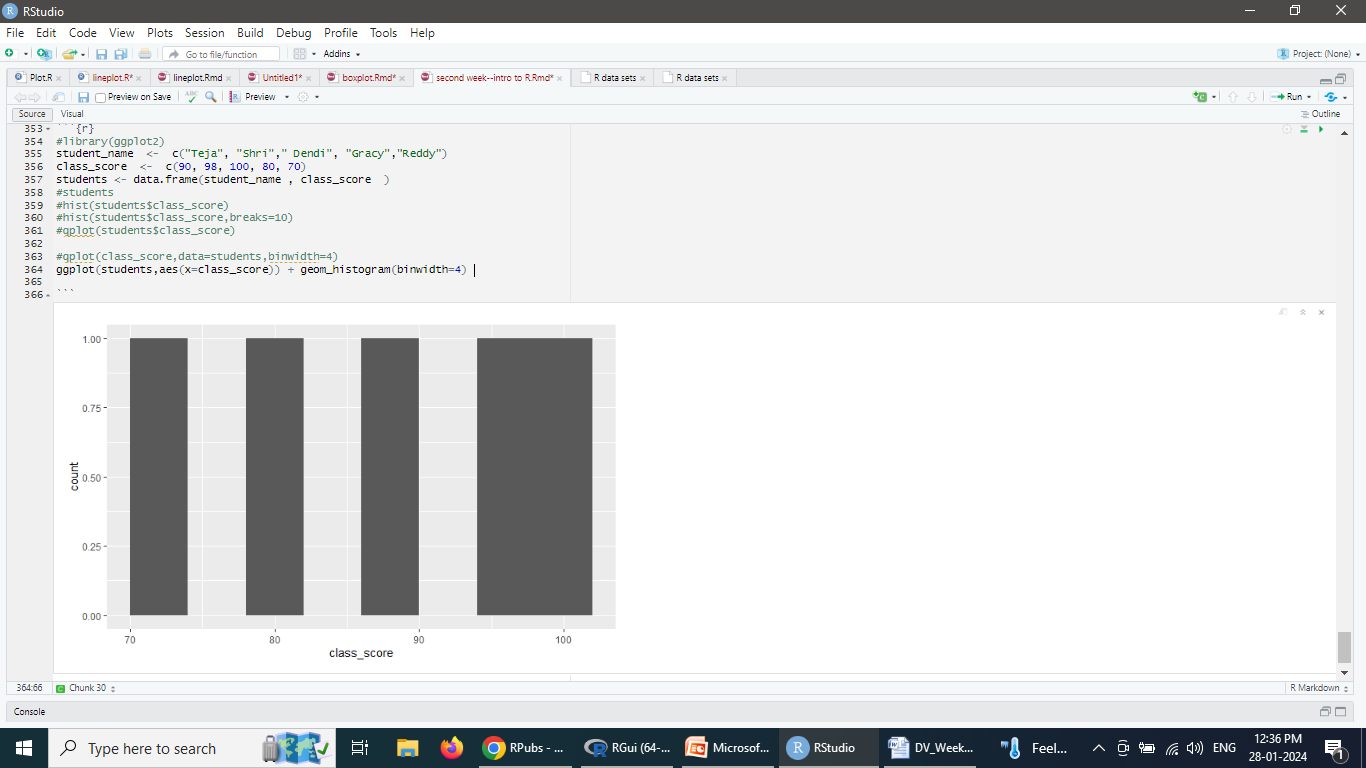
library(ggplot2)

qplot(class\_score,data=students,binwidth=4)



#Or

ggplot(students,aes(x=class\_score)) + geom\_histogram(binwidth=4)



1. Please create a data frame. There two columns in the frame: one is instructors’ names and another is student numbers. There are 5 rows totally, and 2 instructors. Each instructor can have several student numbers and appear several times. (5 points)

instructor<- c(“Elham”, “Sunshine”, “Elham”, “Sunshine”, “Elham”)

student<- c(123,456,756,987,908)

inst\_stu <- data.frame(instructor, student)

1. Use this data frame to draw a boxplot using three ways (15 points)

#***Type1 –plot()***

#To make a box plot, use **plot**() and pass it a factor of x values and a vector of y values. #When x is a factor, it will automatically create a box plot:

plot(inst\_stu $ instructor, inst\_stu $ student)

#***Type2 – boxplot()***

#If the two vectors are in the same data frame, combine two variables on the x-axis

#then the **boxplot** is as follows:

boxplot(student ~ instructor,data = inst\_stu)

# Put interaction of two variables on x-axis

boxplot(len ~ instructor + student, data = inst\_stu)

#***Type3 – qplot()***

#With the ggplot2 package, use **qplot**() with geom="boxplot":

library(ggplot2)

qplot(inst\_stu $ instructor, inst\_stu $ student,geom="boxplot")

#If the two vectors are already in the same data frame,

qplot(instructor, student,data= inst\_stu,geom="boxplot")

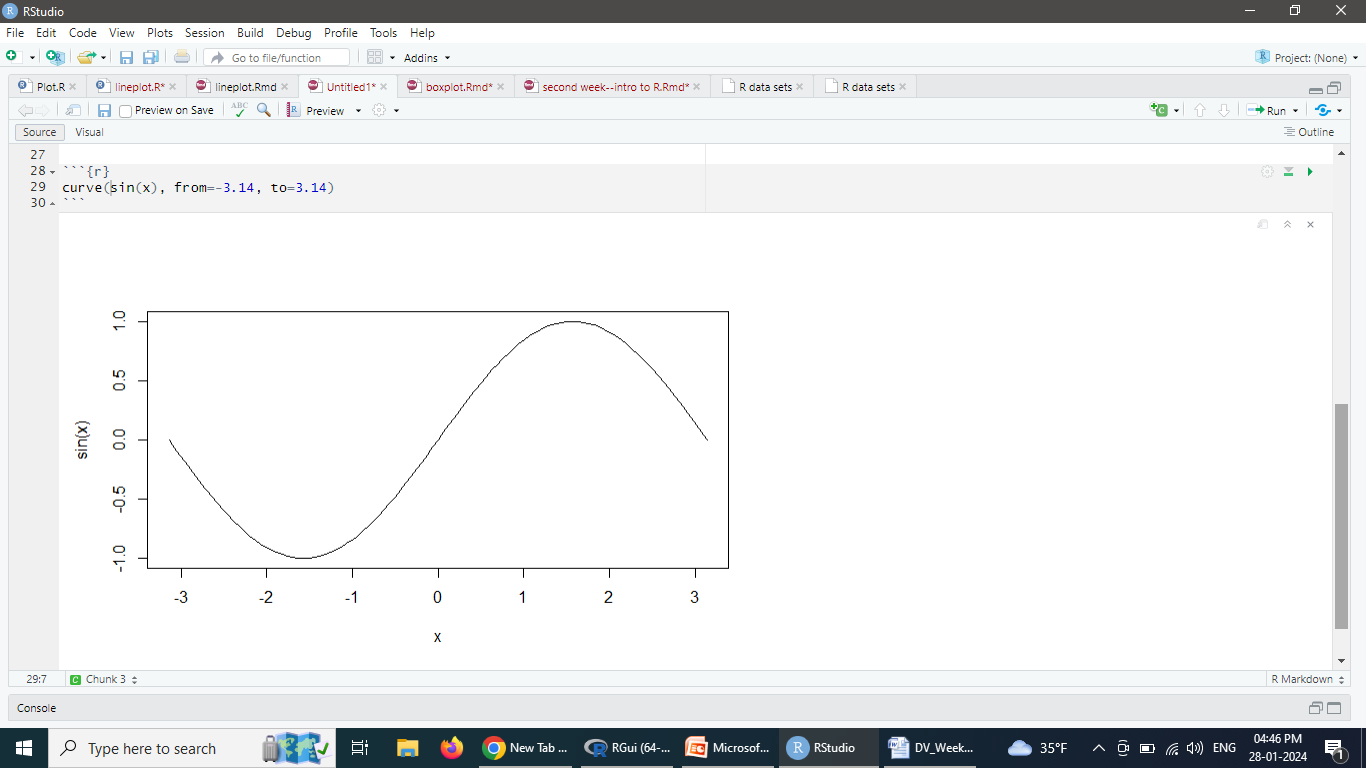
#Or

ggplot(inst\_stu,aes(x= instructor,y= student)) +geom\_boxplot()

1. Write a function as: y=sin(x) and plot this function using two ways. The range is from [-pi, pi]. (15 points)

#use curve()and pass it an expression with the variable x:

curve(sin(x), from=-3.14, to=3.14)



#plot any function. add=TRUE will add a curve to the previously created plot

# Plot a user-defined function

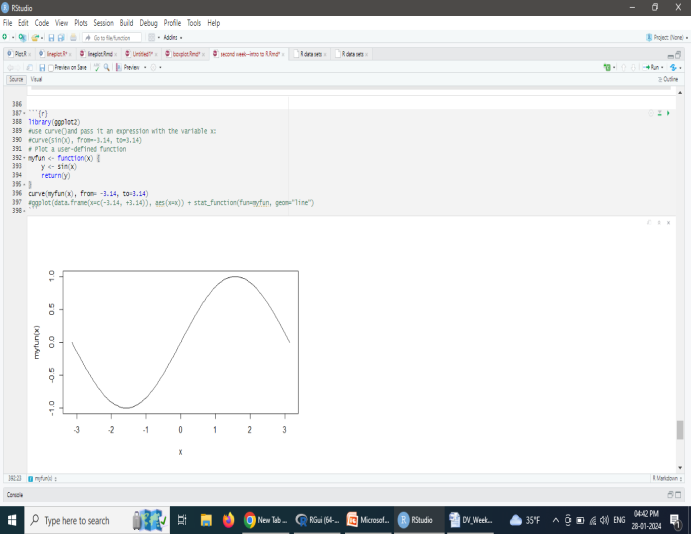
myfun <- function(x) {

y <- sin(x)

return(y)

}

curve(myfun(x), from= -3.14, to=3.14)



# Add a line:

curve(1-myfun(x),add = TRUE, col = "red")

#With ggplot2 package, use stat="function“ & geom="line“ and passing it to a function

library(ggplot2)

ggplot(data.frame(x=c(-3.14, +3.14)), aes(x=x)) + stat\_function(fun=myfun, geom="line")

