**PRACTICAL**

1. **Write a program to display student profile**

**Code :**

Firstname <- "Poorna"

Lastname <- "Patel"

Username <- "Poorna18"

cat ("First name:",Firstname,"Lastname:",Lastname,"Username:",Username)

**Output :**

****

1. **Write a program for performing simple calculation ( take value from user)**

**Code :**

a <- as.integer(readline(prompt="Enter the first number: ") )

b <- as.integer(readline(prompt="Enter the second number: ") )

cat("The ADDITION is : ", (a+b),"\n")

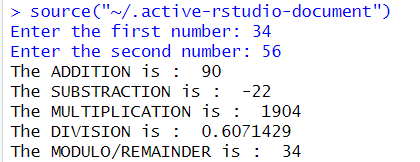
cat("The SUBSTRACTION is : ", (a-b),"\n")

cat("The MULTIPLICATION is : ", (a\*b),"\n")

cat("The DIVISION is : ", (a/b),"\n")

cat("The MODULO/REMAINDER is : ", (a%%b),"\n")

**Output :**

****

1. **Write a program to check whether number is even or odd**

**Code :**

n<-readline(prompt = "Enter a number : ")

n<-as.integer(n)

if(n%%2 == 0){

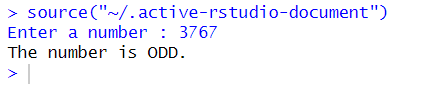
cat("The number is EVEN.")

} else {

cat("The number is ODD.")

}

**Output :**

****

1. **Write a program to check whether number is positive , negative or zero**

**Code :**

n <- as.integer(readline(prompt="Enter a number: ") )

if(n>0){

cat("The number is POSITIVE.")

} else if (n<0) {

cat("The number is NEGATIVE.")

} else {

cat("The number is ZERO.")

}

**Output :**

****

1. **Write a program for finding greatest of two number**

**Code :**

a <- as.integer(readline(prompt="Enter the first number: ") )

b <- as.integer(readline(prompt="Enter the second number: ") )

if(a>b){

cat(a ,"is greater than",b,".\n")

}else if(b>a){

cat(b ,"is greater than",a,".\n")

}else {

cat(a ,"and",b,"booth are equal.\n")

}

**Output :**

****

1. **Write a program using if else ladder for Grading**

**Code :**

print(".....Marks Category.....")

Maths <- as.numeric(readline(prompt = "Enter Marks for Maths : "))

English <- as.numeric(readline(prompt = "Enter Marks for English : "))

Gujarati <- as.numeric(readline(prompt = "Enter Marks for Gujarati : "))

Hindi <- as.numeric(readline(prompt = "Enter Marks for Hindi : "))

Science <- as.numeric(readline(prompt = "Enter Marks for Science : "))

fail <- ""

total <- (Maths+English+Gujarati+Hindi+Science)

if(is.numeric(Maths) && is.numeric(English) && is.numeric(Gujarati) && is.numeric(Hindi) && is.numeric(Science)) {

if(total > 500) {

print("Marks should not more than 100")

} else {

if(Maths < 32) {

fail <- paste(fail, "Maths", sep =",")

}

if(English < 32) {

fail <- paste(fail, "English", sep =",")

}

if(Gujarati < 32) {

fail <- paste(fail, "Gujarati", sep =",")

}

if(Hindi < 32) {

fail <- paste(fail, "Hindi", sep =",")

}

if(Science < 32) {

fail <- paste(fail, "Science", sep =",")

}

if(fail != "") {

cat ("You Fail in", fail)

} else {

cat ("Total => ", total, "\n")

avg <- total / 5

cat ("Percentage => ", avg, "\n")

if(avg <= 100 && avg > 65) {

print("First Class")

} else if(avg <= 65 && avg > 50) {

print("Second Class")

} else if(avg <= 50 && avg > 33) {

print("Pass Class")

} else if(avg <= 33 && avg >= 0) {

print("You Fail! Better luck next time")

} else {

print("Invalid Input")

}

}

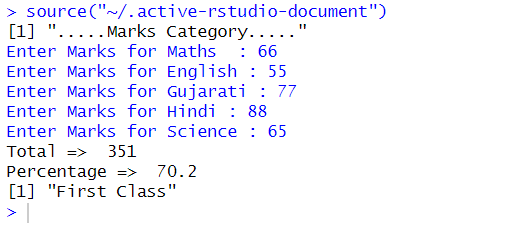
}

} else {

print("Invalid Entry! Please enter marks in number only")

}

**Output :**

****

1. **Write a program to print Fibonacci**  **series**

**Code :**

number <- as.integer(readline(prompt="Enter the numbers till you want Fibonacci  of = "));

a=0;

b=1;

cat(a,"",b ,"")

while(number!=0){

c=a+b;

a=b;

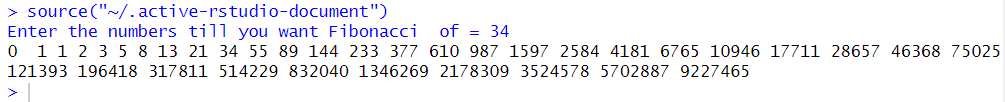
b=c;

cat(c,"");

number=number-1;

}

**Output :**

****

1. **Write a program to find factorial of a number using loop**

**Code :**

n <- as.integer(readline(prompt="Enter a number: ") )

factorial <- 1

if(n < 0) {

cat("factorial does not exist for negative numbers")

} else if(n == 0) {

cat("The factorial of 0 is 1")

} else if(n == 1) {

cat("The factorial of 1 is 1")

} else {

for(i in 2:n)

{

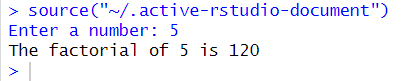
factorial = factorial \* i

}

cat("The factorial of", n ,"is",factorial)

}

**Output :**

****

1. **Write a program to check whether number Palidrome or not**

**Code :**

n <- as.integer(readline(prompt="Enter a number: ") )

org = n

reverse <- 0L

while (n != 0){

remainder <- (n%%10)

reverse <- (reverse\*10 + remainder)

n <- as.integer(n/10)

}

if(org == reverse){

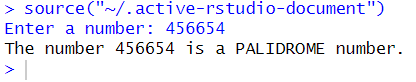
cat("The number",org,"is a PALIDROME number.")

} else {

cat ("The number",org,"is not a PALIDROME number.")

}

**Output :**

****

1. **Write a program to check whether number Armstrong or not**

**Code :**

n <- as.integer(readline(prompt="Enter a number :: ") )

org <- n

temp <- n

len = 0

sum <- 0L

while(temp != 0){

temp <- floor(temp/10)

len <- len+1

}

while (n != 0){

remainder <- (n%%10)

sum <- (sum + (remainder^len))

n <- as.integer(n/10)

}

if(org == sum){

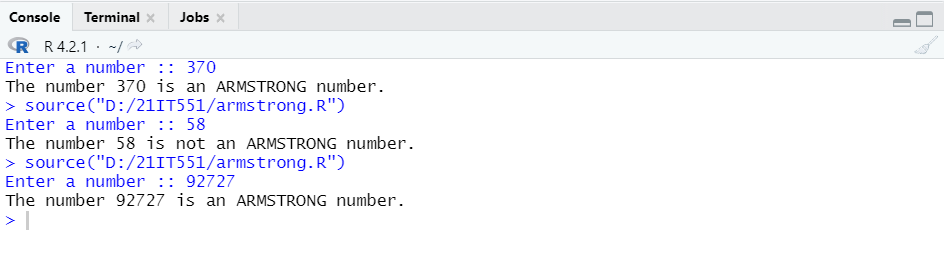
cat("The number",org,"is an ARMSTRONG number.")

} else {

cat ("The number",org,"is not an ARMSTRONG number.")

}

**Output :**

****

1. **Write a program to find factorial of a number using recursion**

**Code :**

n <- as.integer(readline(prompt="Enter a number: ") )

factorial <- function(n) {

if(n < 0) {

cat("factorial does not exist for negative numbers")

} else if(n == 0 || n == 1) {

return (1)

} else {

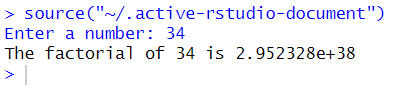
return (n \* factorial(n-1))

}

}

cat("The factorial of", n ,"is",factorial(n))

**Output :**

****

1. **Write a program to print vector**

**Code :**

vec<-c(10.1,10.2,33.2)

print(vec)

**Output :**

****

1. **Write a program to ADD two vector**

**Code :**

vec1<-c(10.1,10.2,33.2)

vec2<-c(10.1,10.2,33.2)

result<-vec1+vec2

cat("---VECTOR 1---\n")

print(vec1)

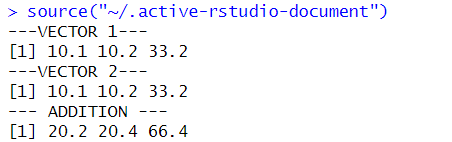
cat("---VECTOR 2---\n")

print(vec2)

cat("--- ADDITION ---\n")

print(result)

**Output :**

****

1. **Write a program to create an array from two vector**

**Code :**

vec1<-c(1,2,3)

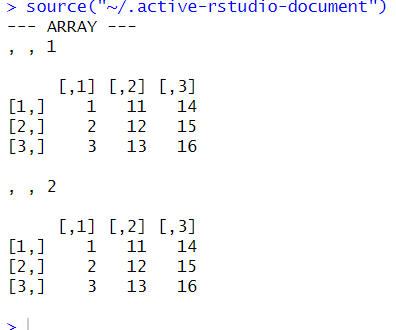
vec2<-c(11,12,13,14,15,16)

res<-array(c(vec1,vec2),dim=c(3,3,2))

cat("--- ARRAY ---\n")

print(res)

**Output :**

****

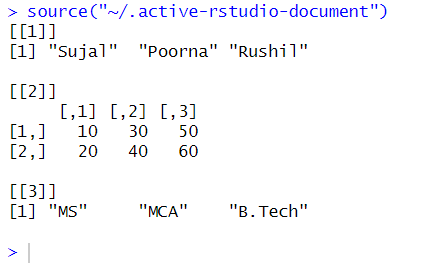
1. **Write a program to create a list containing a vector, a matrix and a list.**

**Code :**

data<-list(c("Sujal", "Poorna", "Rushil"), matrix(c(10, 20, 30, 40, 50, 60), nrow=2), c("MS", "MCA", "B.Tech"))

print(data)

**Output :**

****

1. **Write a program for giving names to the elements in the list.**

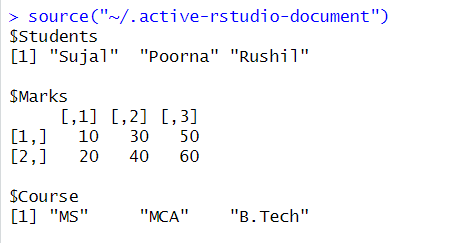
**Code :**

data<-list(c("Sujal", "Poorna", "Rushil"), matrix(c(10, 20, 30, 40, 50, 60), nrow=2), c("MS", "MCA", "B.Tech"))

names(data)<-c("Students","Marks","Course")

print(data)

**Output :**

****

1. **Write a program for accessing the first element of the list.**

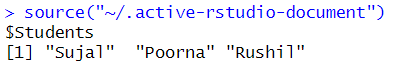
**Code :**

data<-list(c ("Sujal", "Poorna", "Rushil")matrix(c(10, 20, 30, 40, 50, 60), nrow=2), c("MS", "MCA", "B.Tech"))

names(data)<-c("Students","Marks","Course")

print(data[1])

**Output :**

****

1. **Write a program for unlisting a list into vector**

**Code :**

data<-list(c ("Sujal", "Poorna", "Rushil"),matrix(c(10, 20, 30, 40, 50, 60), nrow=2), c("MS", "MCA", "B.Tech"))

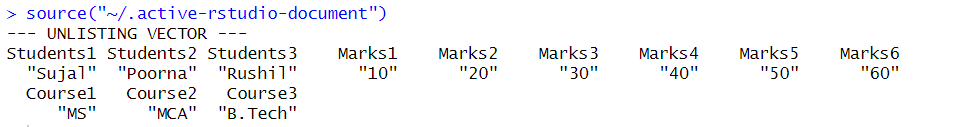
names(data)<-c("Students","Marks","Course")

vec<-unlist(data)

cat("--- UNLISTING VECTOR ---\n")

print(vec)

**Output :**

****

1. **Write a program for converting a vector int a list.**

**Code :**

data1<-list(c ("Sujal", "Poorna", "Rushil"),matrix(c(10,20,30,40,50,60),nrow=2), c("MS","MCA","B.Tech"))

data2<-list(c("Himanshu","Darshak","keertan"), matrix(c(70,80,90,10,20,30),nrow=2), c("MA","BBA","BSc"))

names(data1)<-c("Students","Marks","Course")

names(data2)<-c("Students","Marks","Course")

vec1<-unlist(data1)

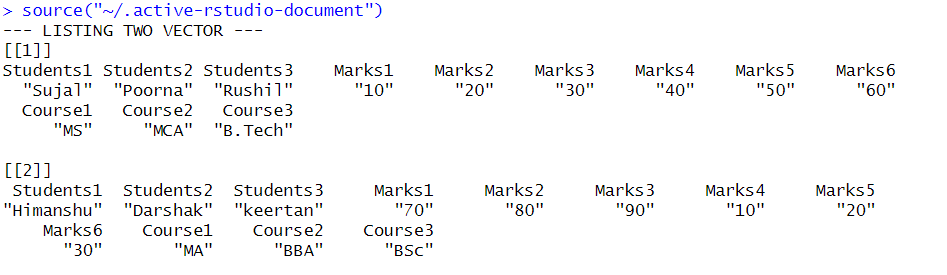
vec2<-unlist(data2)

cat("--- LISTING TWO VECTOR ---\n")

a<-list(vec1,vec2)

print(a)

**Output :**

****

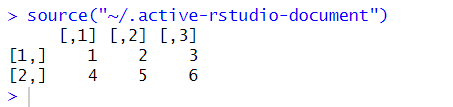
1. **Write a program for create and print a Matrix**

**Code :**

matrix1<-matrix(c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=TRUE)

print(matrix1)

**Output :**

****

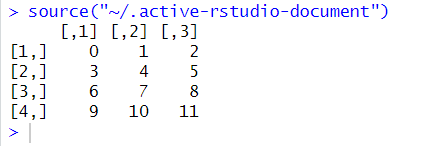
1. **Write a program for arrange a matrix by row**

**Code :**

P<-matrix(c(0:11),nrow=4,byrow=TRUE)

print(P)

**Output :**



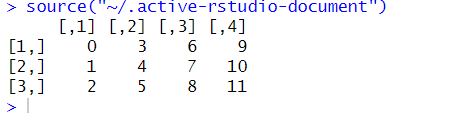
1. **Write a program for arrange a matrix by col**

**Code :**

P<-matrix(c(0:11),ncol=4,byrow=FALSE)

print(P)

**Output :**

****

1. **Write a program for giving name to matrix column and row**

**Code :**

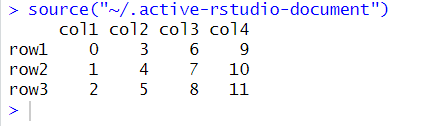
row\_names<-c("row1","row2","row3")

col\_names<-c("col1","col2","col3","col4")

P<-matrix(c(0:11),ncol=4,byrow=FALSE,dimnames=list(row\_names,col\_names))

print(P)

**Output :**

****

1. **Write a program for accessing specific element**

**Code :**

Mat<-matrix(c(0:11),ncol=4,byrow=FALSE)

print(Mat)

cat("Printing 3rd row 2nd column element\n")

print(Mat[3,2])

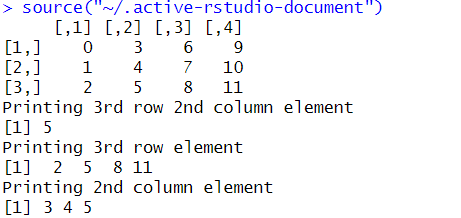
cat("Printing 3rd row element\n")

print(Mat[3,])

cat("Printing 2nd column element\n")

print(Mat[,2])

**Output :**



1. **Write a program for accessing specific element**

**Code :**

Mat<-matrix(c(0:11),ncol=4,byrow=FALSE)

print(Mat)

cat("Making Transpose\n")

print(t(Mat))

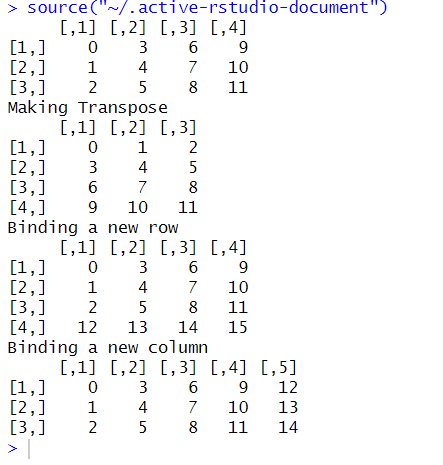
cat("Binding a new row\n")

print(rbind(Mat,c(12:15)))

cat("Binding a new column\n")

print(cbind(Mat,c(12:14)))

**Output :**

****

1. **Write a program for create and print a Data frames**

**Code :**

student.data<-data.frame(

student\_id=c(1:4),

student\_name=c("Jaivik", "Rahul", "Jaimin", "Ravi"),

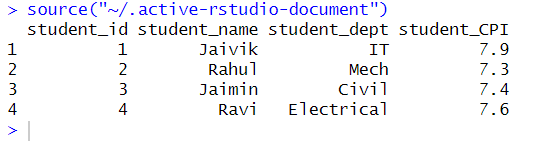
student\_dept=c("IT", "Mech", "Civil", "Electrical"),

student\_CPI=c(7.9,7.3,7.4,7.6)

)

print(student.data)

**Output :**

****

1. **Write a program for print a Data frames table how are having value more than 7.0 CPI**

**Code :**

student.data<-data.frame(

student\_id=c(1:4),

student\_name=c("Sujal", "Rushil", "Jaimin", "Poorna"),

student\_dept=c("IT", "Mech", "Civil", "Electrical"),

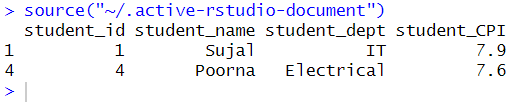
student\_CPI=c(7.9,7.3,7.4,7.6)

)

result<-student.data[student.data$student\_CPI >7.5, ]

print(result)

**Output :**



1. **Write a program for printing and viewing csv/excel**

**Code :**

getwd()

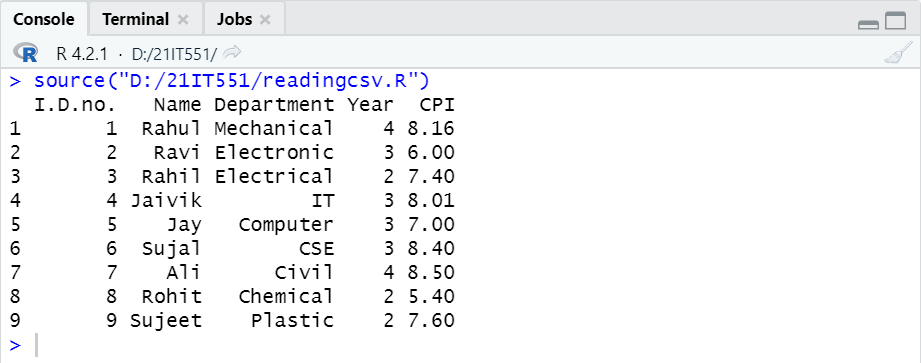
setwd("C:\Documents\22IT607")

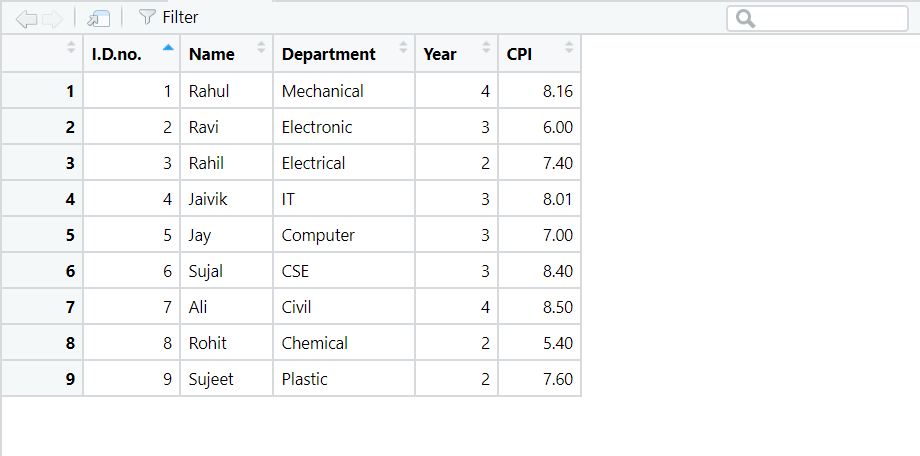
data <- read.csv("Student.csv")

print(data)

View(data)

**Output :**

****

****

1. **Write a program for printing the mean of a column**

**Code :**

getwd()

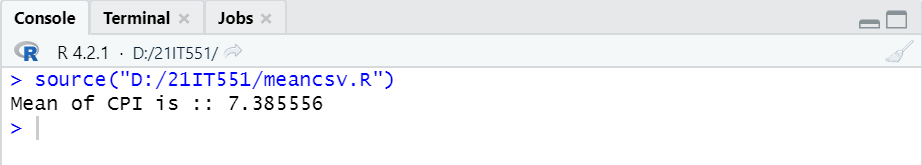
setwd("D:/22IT607")

data <- read.csv("Student.csv")

m<-mean(data$CPI)

cat("Mean of CPI is ::",m)

**Output :**

****

1. **Write a program for printing the number of columns and rows in the table**

**Code :**

getwd()

setwd("D:/22IT607")

data <- read.csv("Student.csv")

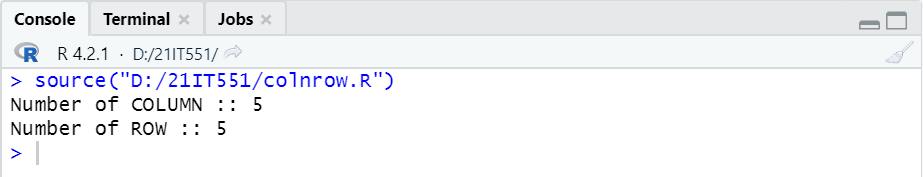
column<-ncol(data)

cat("Number of COLUMN ::",column)

row<-nrow(data)

cat("\nNumber of ROW ::",column)

**Output :**

****

1. **Write a program for printing the maximum of the column(CPI)**

**Code :**

getwd()

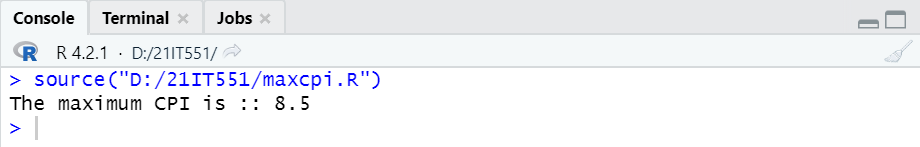
setwd("D:/22IT607")

data <- read.csv("Student.csv")

max\_cpi<- max(data$CPI)

cat("The maximum CPI is ::",max\_cpi)

**Output :**

****

1. **Write a program to print the detail via condition**

**Code :**

getwd()

data <- read.csv("Student.csv")

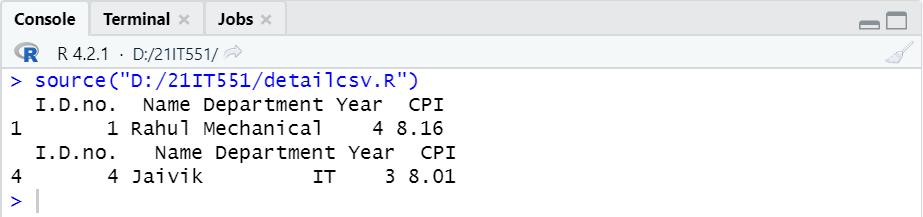
detail1 <- subset(data,Department=="Mechanical")

print(detail1)

detail2 <- subset(data,Department=="IT"&CPI>7.5)

print(detail2)

**Output :**

****

1. **Write a program to print the detail in another file name output.csv**

**Code :**

getwd()

setwd("D:/21IT551")

data <- read.csv("Student.csv")

detail <- subset(data,Department=="Mechanical"&CPI>"8")

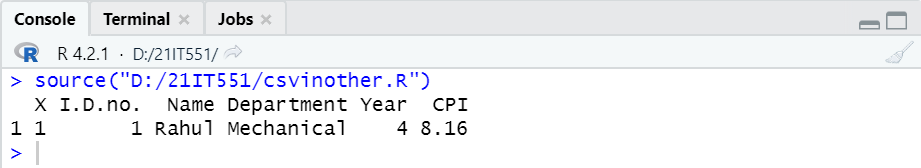
write.csv(detail,"output.csv")

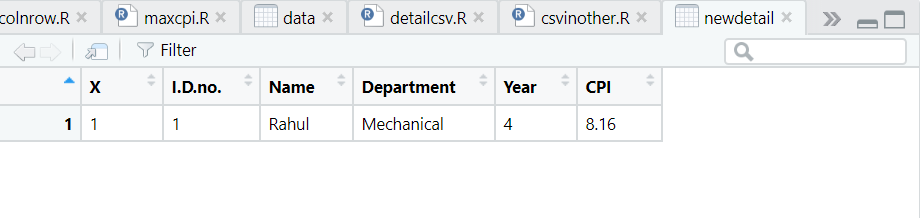
newdetail<- read.csv("output.csv")

View(newdetail)

print(newdetail)

**Output :**

****

****

1. **Write a program to install package “pacman” and view data of mtcars**

**Code :**

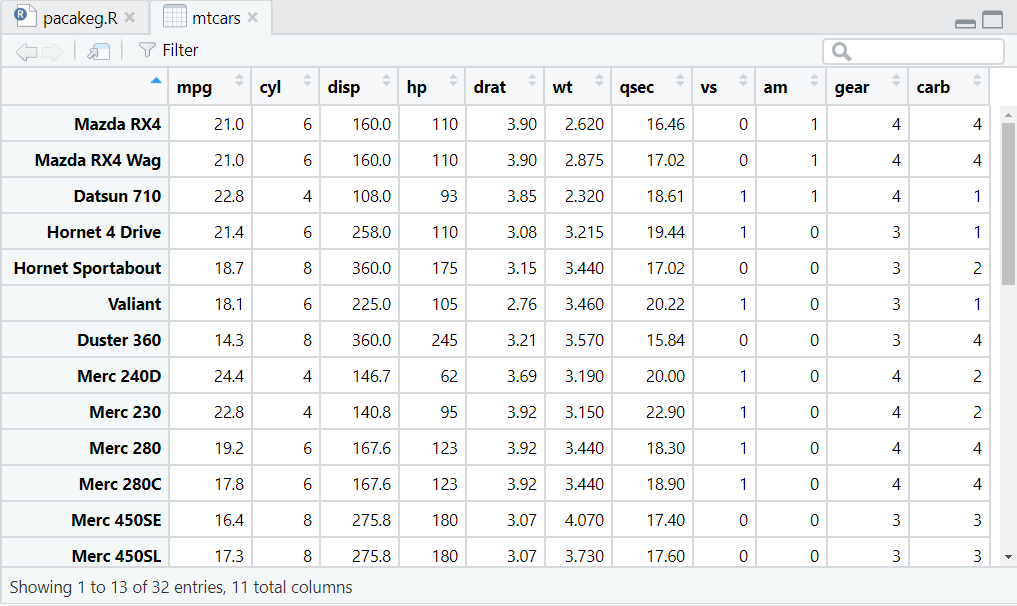
install.packages(“pacman”)

library(pacman)

data()

View(mtcars)

**Output :**

****

1. **Write a program to install package “tidyverse” and view data of mpg**

**Code :**

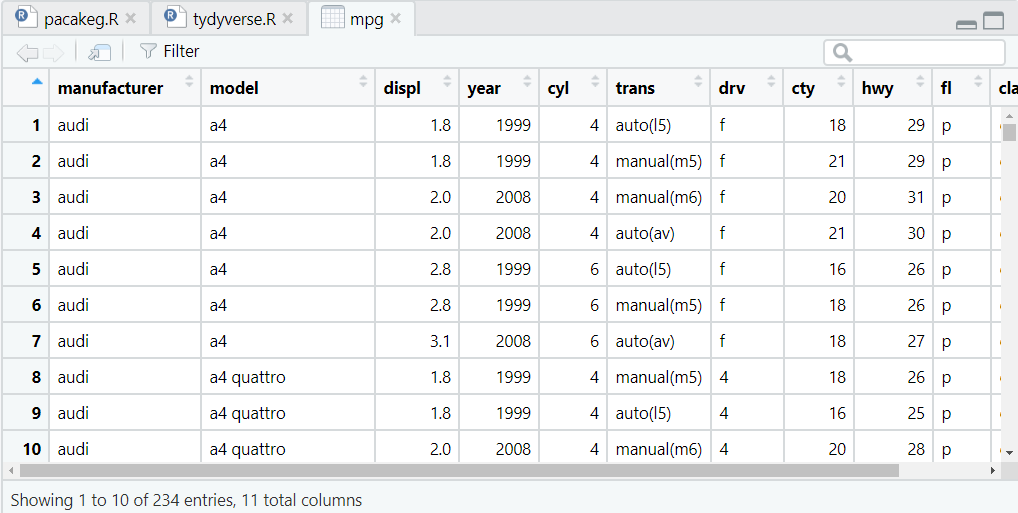
install.packages(“tidyverse”)

library(tidyverse)

data()

View(mtcars)

**Output :**

****

1. **Write a program to view head of package “tidyverse”**

**Code :**

library(tidyverse)

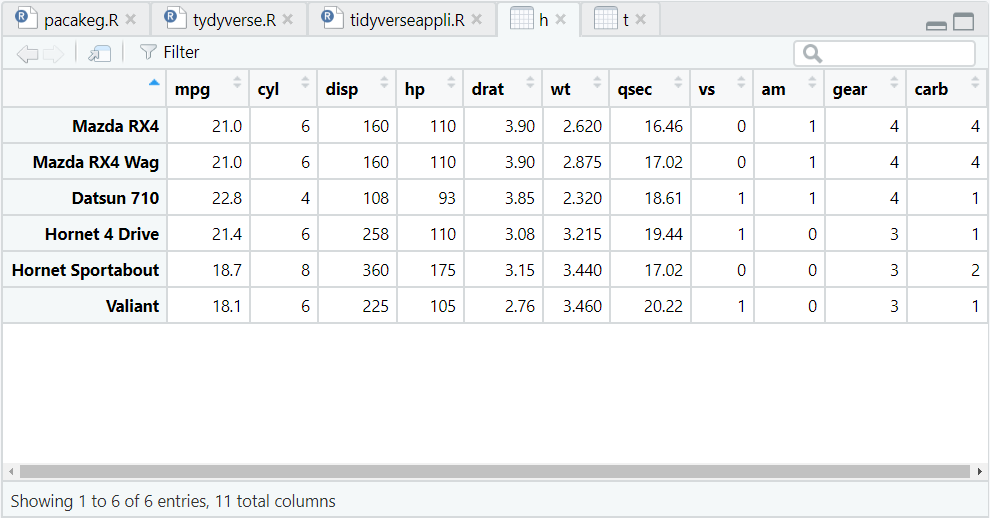
View(mtcars)

#viewing head(top data)

head<-head(mtcars)

View(head)

**Output :**

****

1. **Write a program to view Tail of package “tidyverse”**

**Code :**

library(tidyverse)

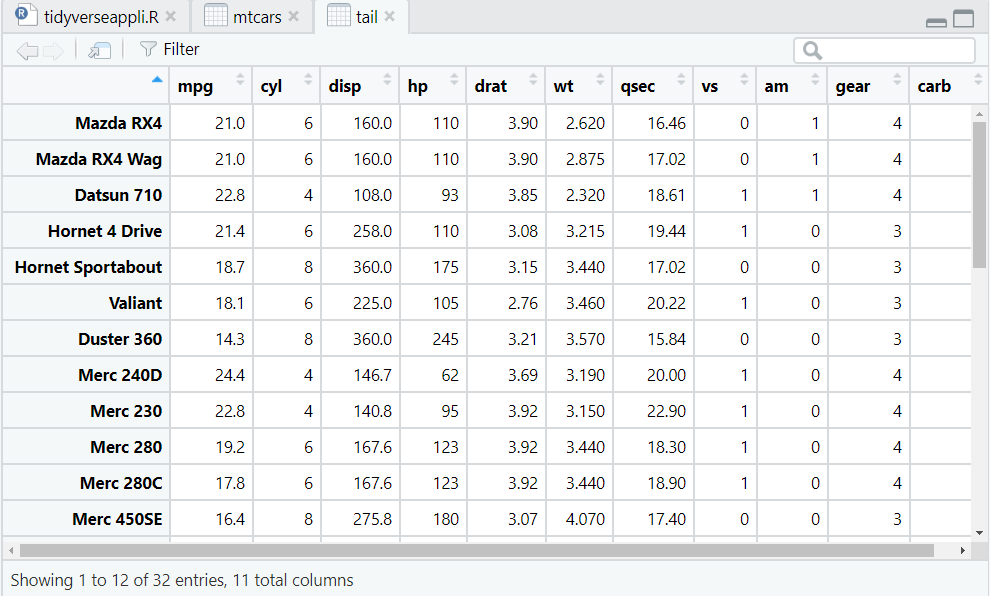
View(mtcars)

#viewing Tail(bottom data)

tail<-head(mtcars)

View(tail)

**Output :**

****

1. **Write a program to view glimpse of mtcar from package “tidyverse”**

**Code :**

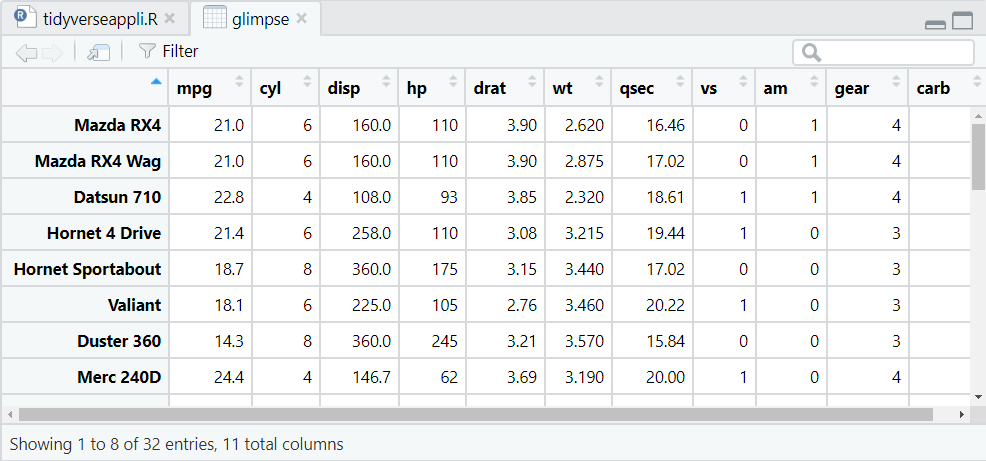
library(tidyverse)

#viewing glimpse

glimpse<-glimpse(mtcars)

View(glimpse)

**Output :**

****

1. **Write a program to create pie chart**

**Code :**

getwd()

setwd("D:/21IT551")

# Create data for the graph.

x <- c(21, 62, 10, 53)

labels <- c("Delhi", "New York", "Singapore", "Mumbai")

png(file = "pie.png")

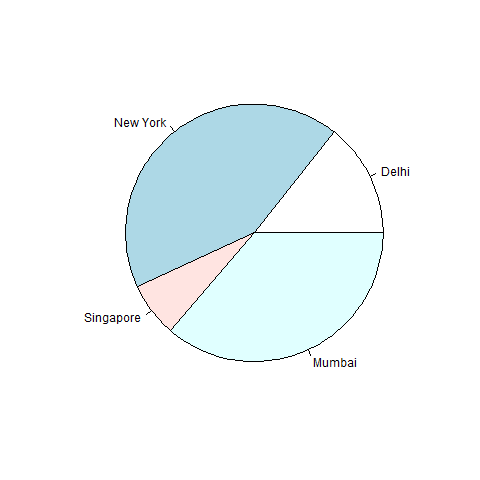
# Plot the chart.

pie(x,labels)

# Save the file.

dev.off()

**Output :**

****

1. **Write a program to create 3D-pie chart**

**Code :**

getwd()

setwd("D:/21IT551")

library(plotrix)

x <- c(21, 62, 10,53)

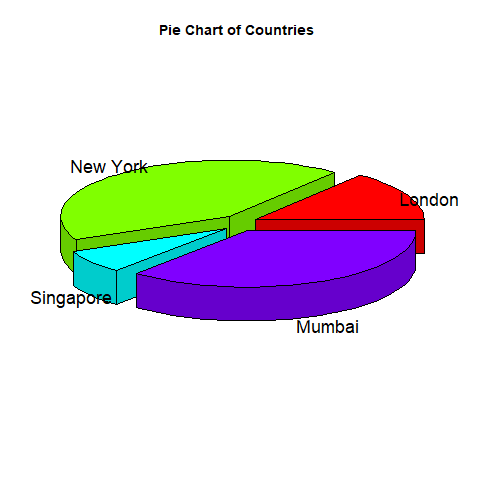
lbl <- c("London","New York","Singapore","Mumbai")

png(file = "3dpie.jpg")

pie3D(x,labels = lbl,explode = 0.1, main = "Pie Chart of Countries ")

dev.off()

**Output :**

****

1. **Write a program to create bar plot chart**

**Code :**

getwd()

setwd("D:/21IT551")

data <- read.csv("Details.csv")

colors <- c("green","orange","brown")

name<-data$Name

age<-data$Age

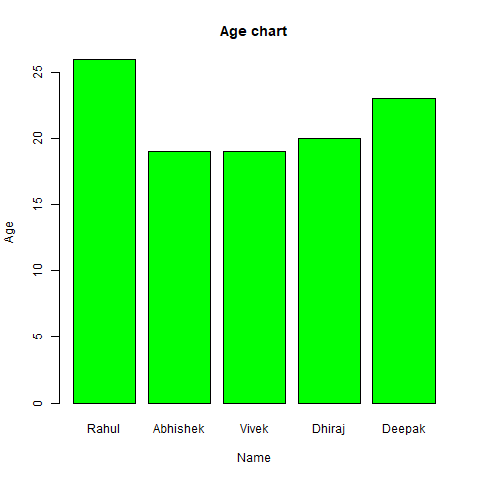
png(file = "barplot.png")

value <- matrix(data$Age, nrow=1,ncol=5,byrow=TRUE)

barplot(value,names.arg=name,xlab="Name",ylab="Age",col=colors,main="Age chart")

dev.off()

**Output :**

****

1. **Write a program to create histogram chart**

**Code :**

getwd()

setwd("D:/21IT551")

v <- c(9,13,21,8,36,22,12,41,31,33,19)

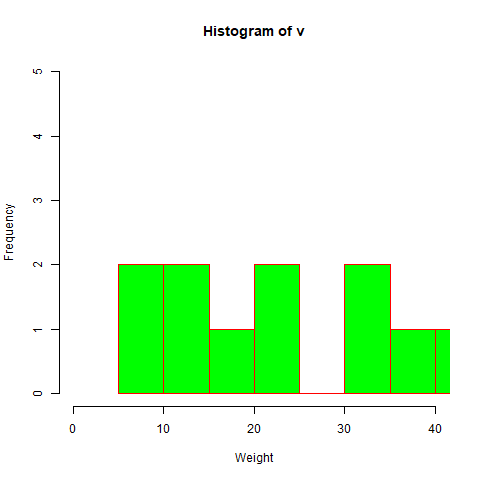
png(file = "histogram.png")

hist(v,xlab = "Weight",col = "green",border = "red", xlim = c(0,40), ylim = c(0,5),

breaks = 5)

dev.off()

**Output :**

****

1. **Write a program to create Scatterplot chart**

**Code :**

getwd()

setwd("D:/21IT551")

input <- mtcars[,c('wt','mpg')]

png(file = "scatterplot.png")

plot(x = input$wt,y = input$mpg,

xlab = "Weight",

ylab = "Milage",

xlim = c(2.5,5),

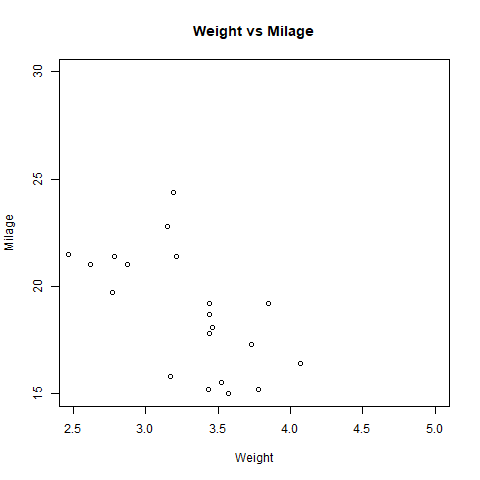
ylim = c(15,30),

main = "Weight vs Milage"

)

dev.off()

**Output :**

****

1. **Write a program to create ggplot chart**

**Code :**

getwd()

setwd("D:/21IT551")

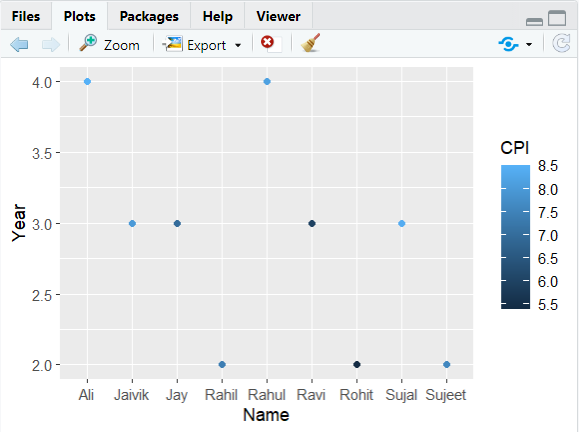
data<-read.csv("Student.csv")

library(ggplot2)

IrisPlot<-ggplot(data, aes(Name, Year, colour=CPI))+ geom\_point()

print(IrisPlot)

**Output :**

****

1. **Write a program to establish data base connectivity with database and print the list of tables**

**Code :**

library(RMySQL)

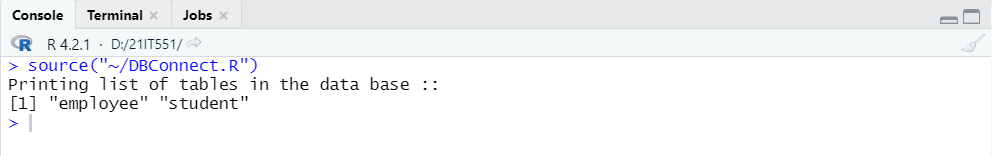
conn <- dbConnect(MySQL(), user = 'root', password = '', dbname = '21it551',

host = 'localhost')

cat("Printing list of tables in the data base :: \n")

print(dbListTables(conn))

**Output :**

****

1. **Write a program to print the tables from the database**

**Code :**

library(RMySQL)

conn <- dbConnect(MySQL(), user = 'root', password = '', dbname = '21it551',

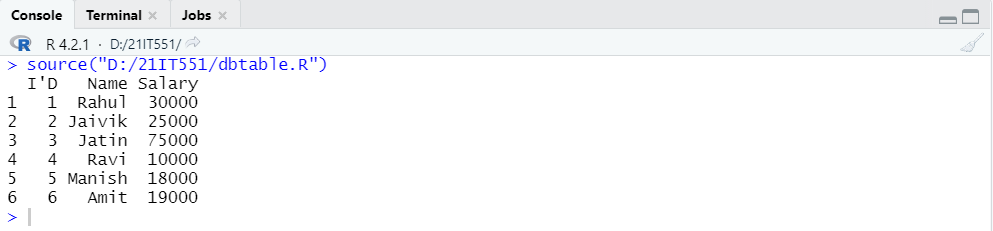
host = 'localhost')

record <- dbSendQuery(conn, "select \* from employee")

data <- fetch(record)

print(data)

**Output :**

****

1. **Write a program to copy data from on table to another in database**

**Code :**

library(RMySQL)

conn <- dbConnect(MySQL(), user = 'root', password = '', dbname = '21it551',

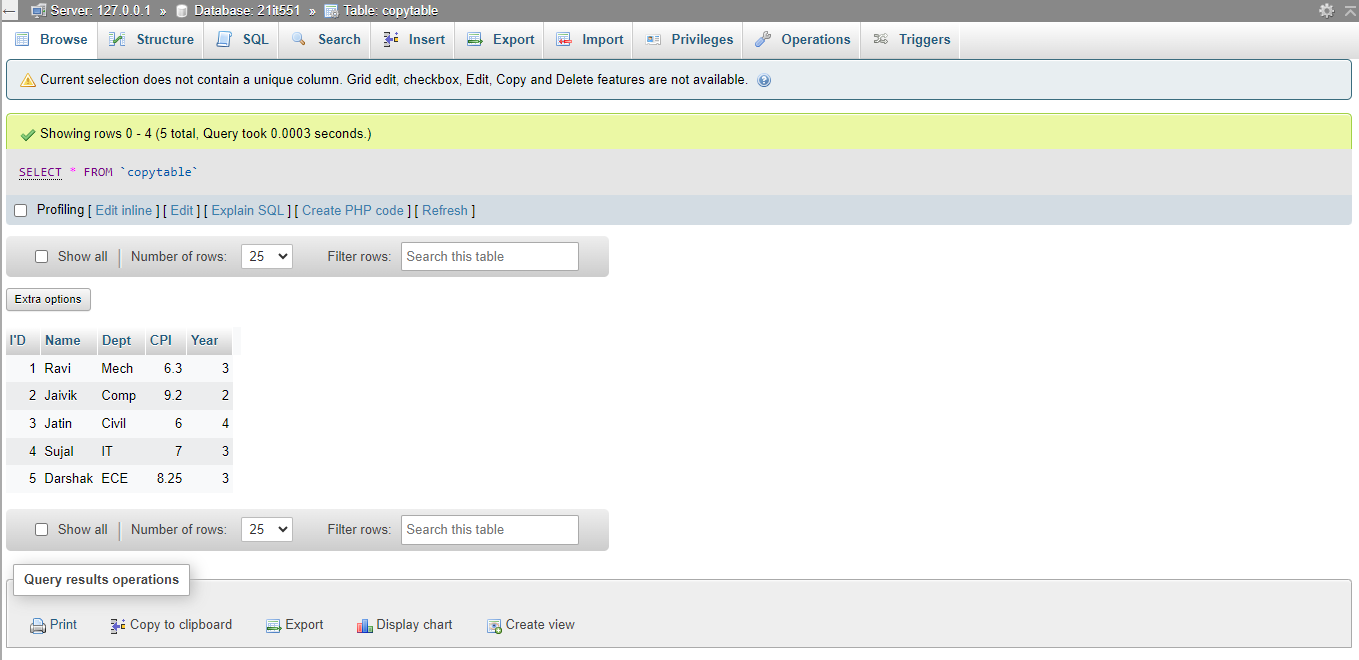
host = 'localhost')

record <- dbSendQuery(conn, "create table copytable select \* from student")

data <- fetch(record)

print(data)

**Output :**

****

1. **Write a program to insert data into data base**

**Code :**

library(RMySQL)

conn <- dbConnect(MySQL(), user = 'root', password = '', dbname = '21it551',

host = 'localhost')

student.data<- data.frame (

Name = c("Ravi","Jaivik","Jatin","Sujal","Darshak"),

Dept = c("Mech","Comp","Civil","IT","ECE"),

CPI = c(6.3,9.2,6.0,7.0,8.25),

Year = c(3,2,4,3,3),

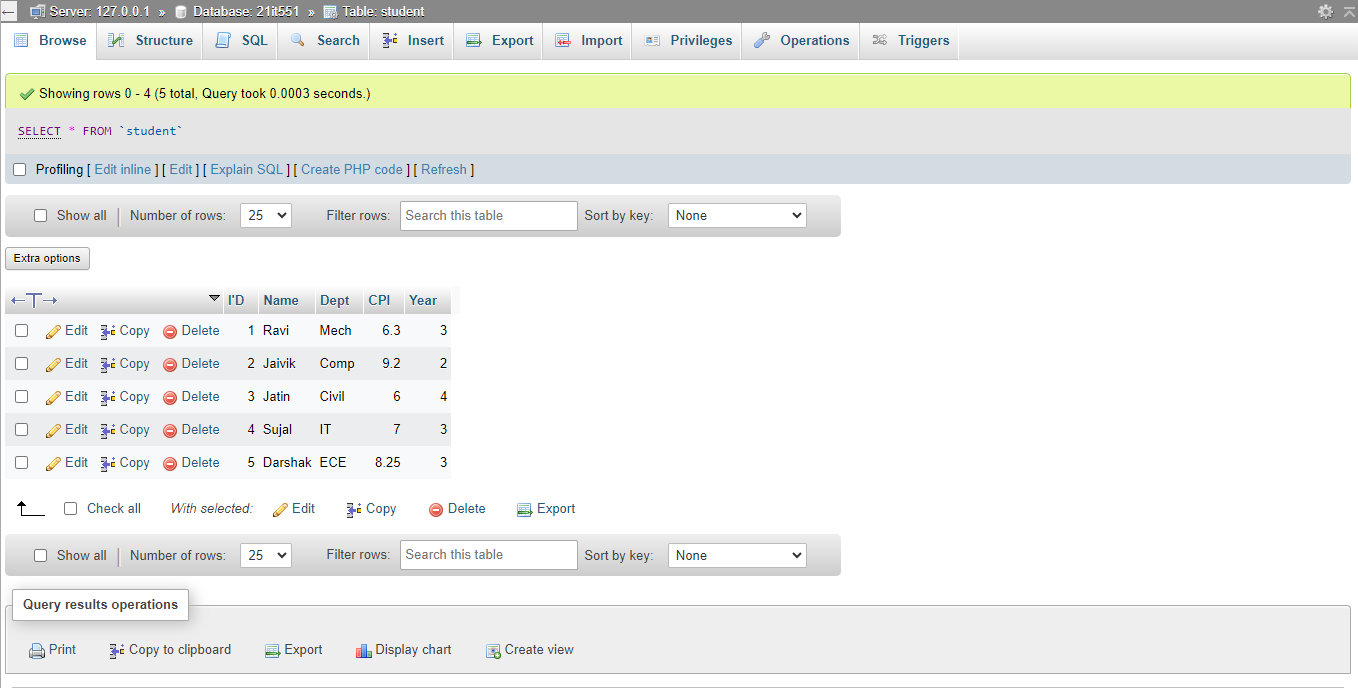
stringsAsFactors = FALSE

)

dbWriteTable(conn, "student", student.data[, ], overwrite = TRUE)

dbDisconnect(conn)

**Output :**

****