**Practical 1 :-** GRAPHICAL METHOD USING R PROGRAMMING.

**Code & Output :-**

#Find a geometrical interpretation and solution as well for the following LP problem

#Max z= 3x1 + 5x2

#subject to constraints:

#x1+2x2<=2000

#x1+x2<=1500

#x2<=600

#x1,x2>=0

#To solve linear programming using R studio, we need to install lpsolve package

install.packages("lpSolve")

require(lpSolve)

C <- c(3, 5)

A <- matrix(c(1, 2,

1, 1,

0, 1), nrow = 3, byrow = TRUE)

B <- c(2000, 1500, 600)

constraints\_direction <- c("<=", "<=", "<=")

plot.new()

plot.window(xlim = c(0, 2000), ylim = c(0, 2000))

axis(1)

axis(2)

title(main = "LPP using Graphical method")

title(xlab = "X axis")

title(ylab = "Y axis")

box()

segments(

x0 = 2000,

y0 = 0,

x1 = 0,

y1 = 1000,

col = "green"

)

segments(

x0 = 1500,

y0 = 0,

x1 = 0,

y1 = 1500,

col = "green"

)

segments(

x0 = 0,

y0 = 0,

x1 = 600,

y1 = 0,

col = "green"

)

optimum <- lp(

direction = "max",

objective.in = C,

const.mat = A,

const.dir = constraints\_direction,

const.rhs = B,

all.int = T

)

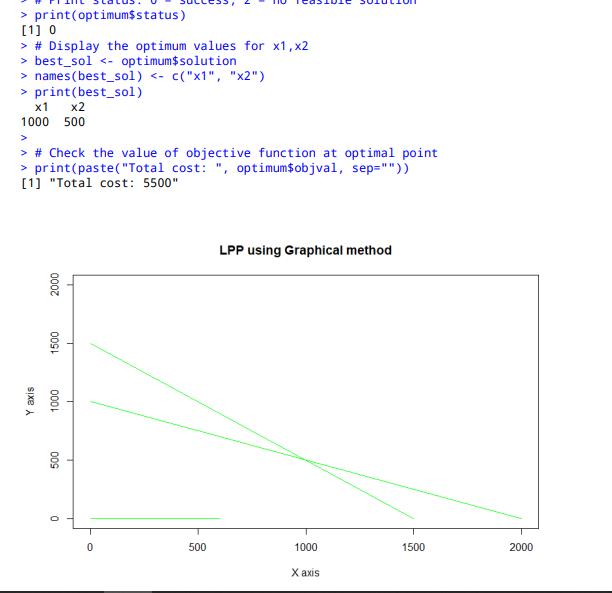
print(optimum$status)

best\_sol <- optimum$solution

names(best\_sol) <- c("x1", "x2")

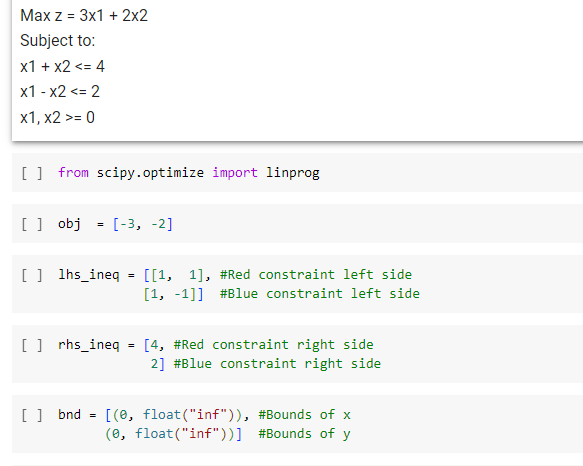
print(best\_sol)

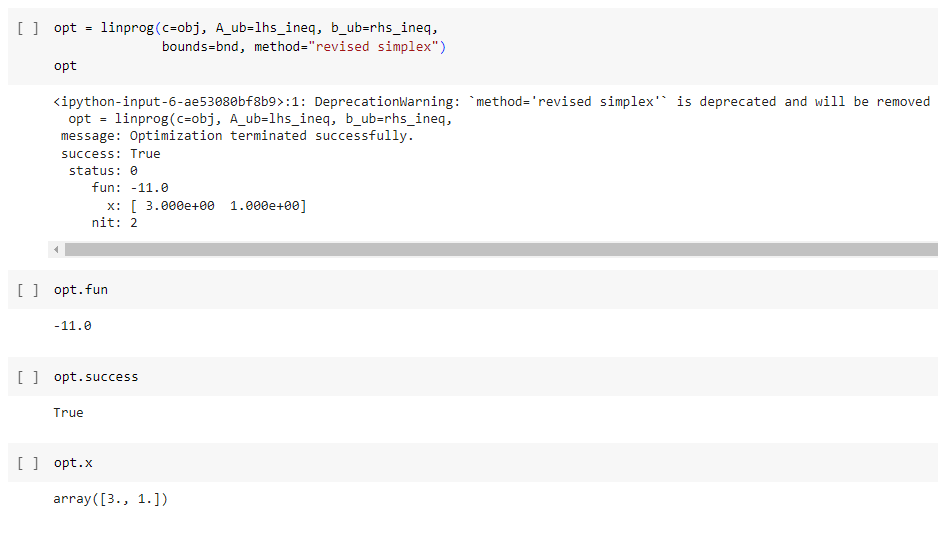
print(paste("Total cost: ", optimum$objval, sep = ""))



**Practical 2 :-** Simplex Method with 2 variables using Python.

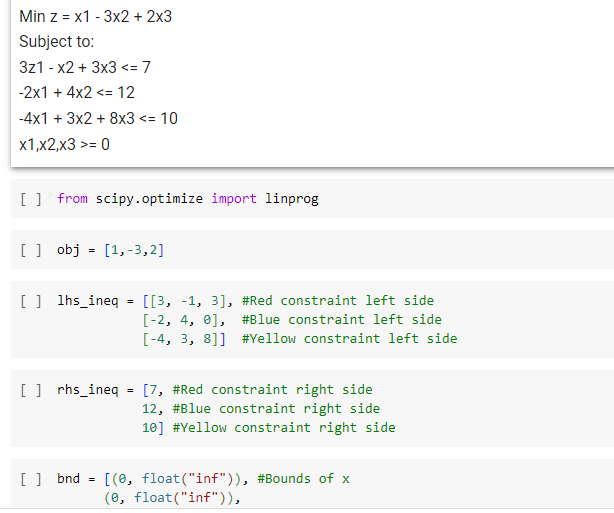
**Code & Output :-**

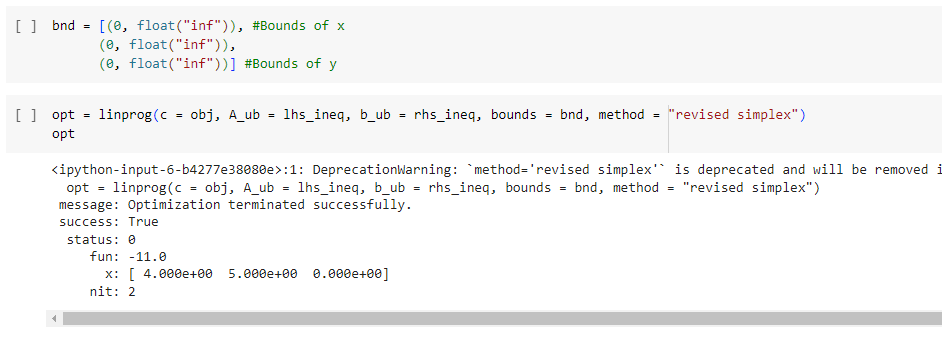
****

****

**Practical 3 :-** Simplex Method with 3 variables using Python.

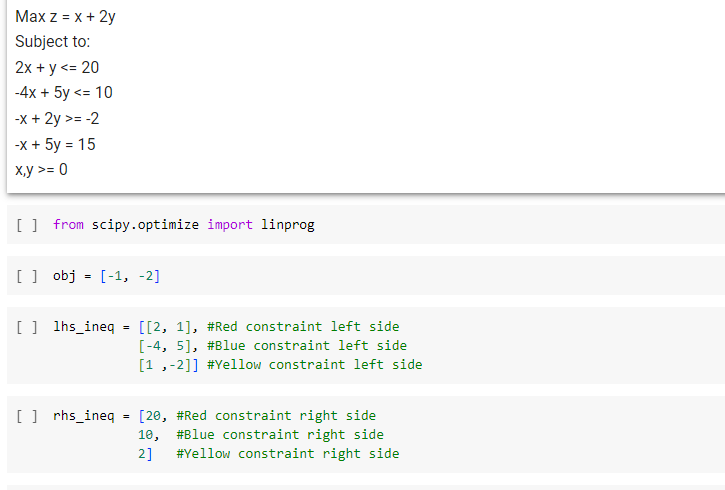
**Code & Output :-**

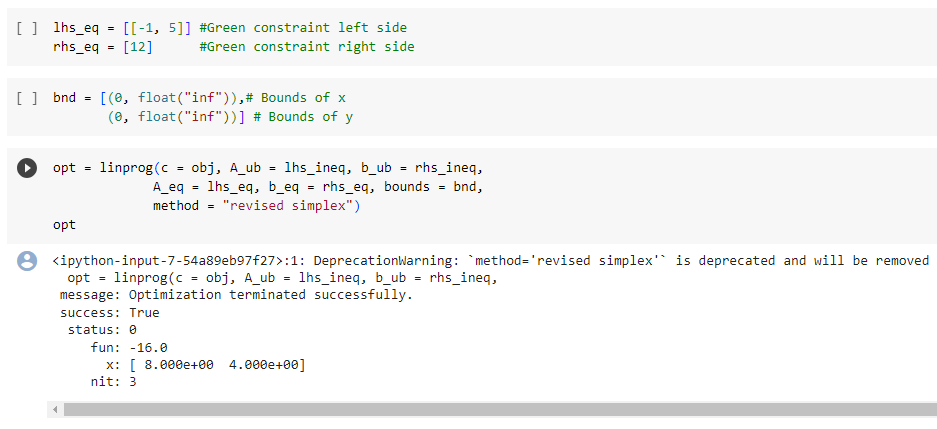




**Practical 4 :-** Simplex Method with Equality Constraints Using Python.

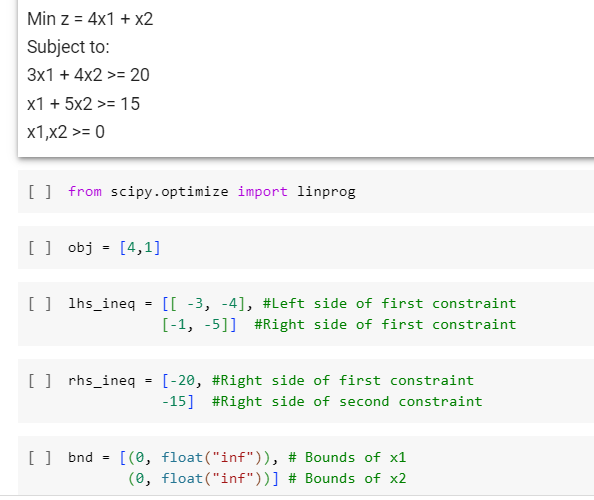
**Code & Output :-**

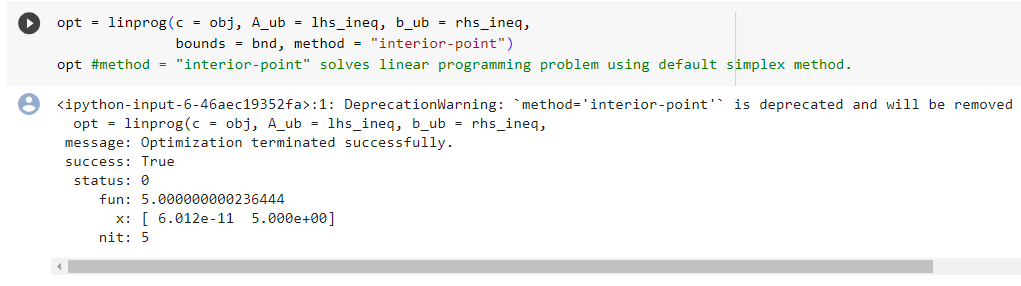




**Practical 5 :-** BigM Simplex Method using Python.

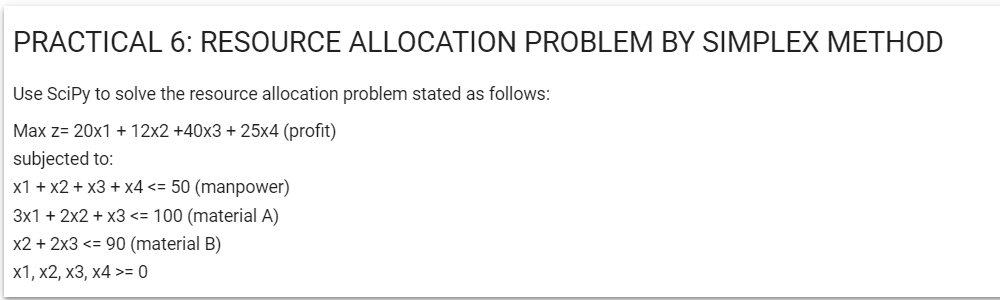
**Code & Output :-**

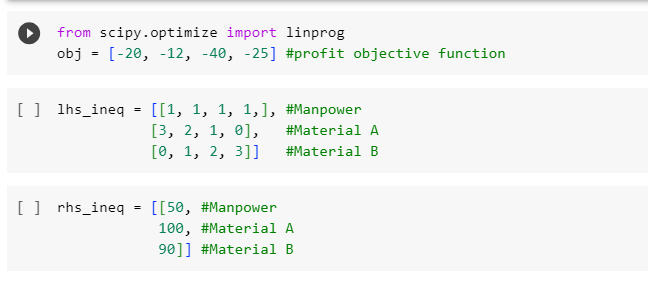


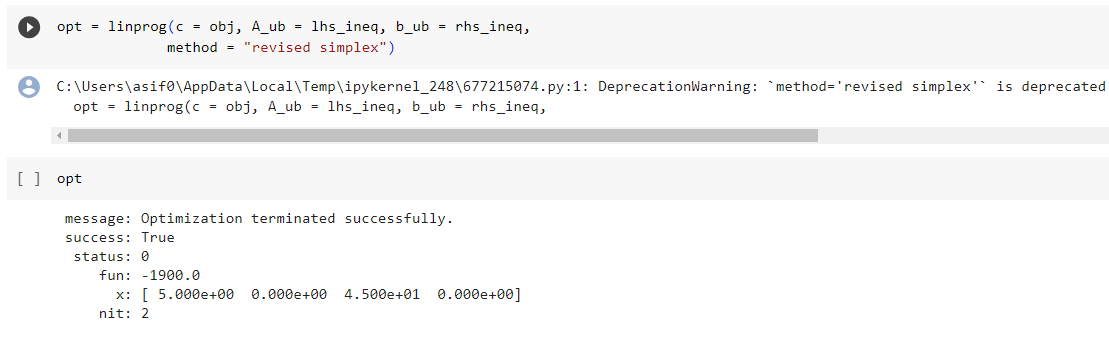


**Practical 6 :-** RESOURCE ALLOCATION PROBLEM BY SIMPLEX METHOD.

**Code & Output :-**

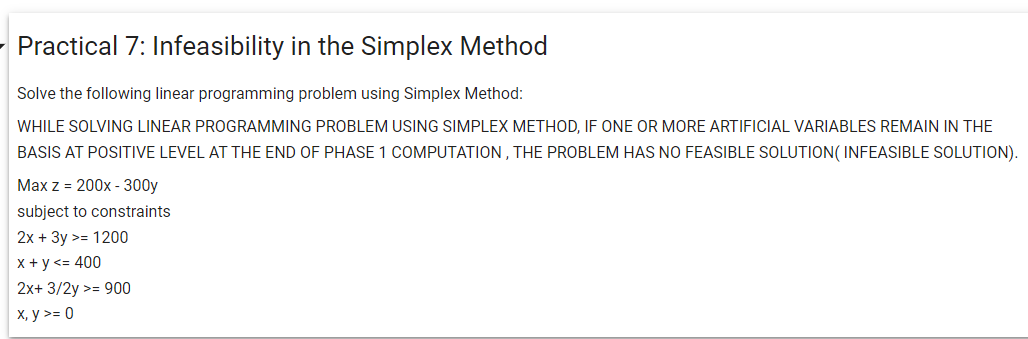
****

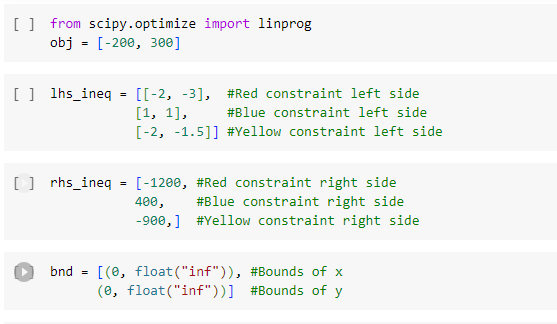
****

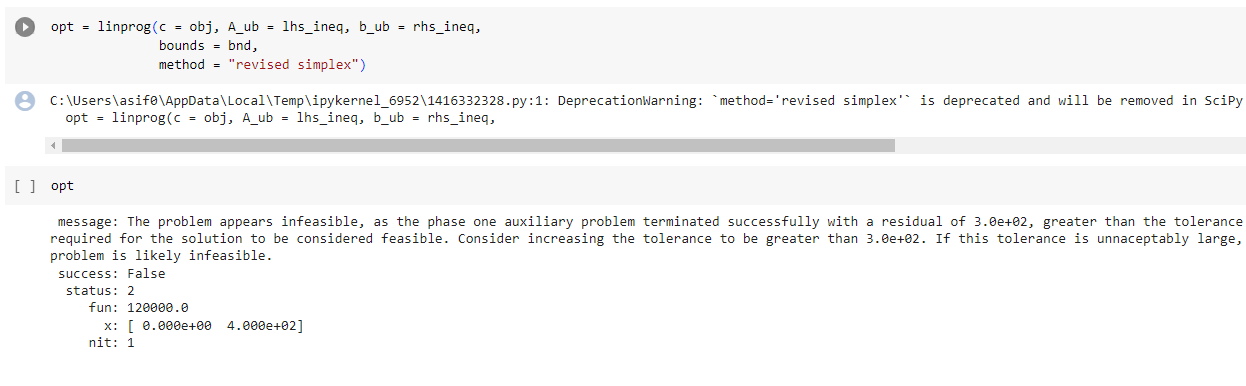
****

**Practical 7:-** INFEASIBILITY IN SIMPLEX METHOD.

**Code & Output :-**

****

****

****

**Practical 8:-** DUAL SIMPLEX METHOD.

**Code & Output :-**

#Solve the following Linear Programming Problem using the Dual Simplex Method in R Programming

#Max Z = 40x1 + 50x2

#Subject to

#2x1 + 3x2 <= 3

#8x1 + 4x2 <= 5

#x1, x2 >= 0

library(lpSolve)

f.obj = c(40, 50)

f.con = matrix(c(2, 3,

8,4), nrow = 2, byrow = TRUE)

f.dir = c("<=",

"<=")

f.rhs = c(3,

5)

lp("max", f.obj, f.con, f.dir, f.rhs)

lp("max", f.obj, f.con, f.dir, f.rhs)$solution

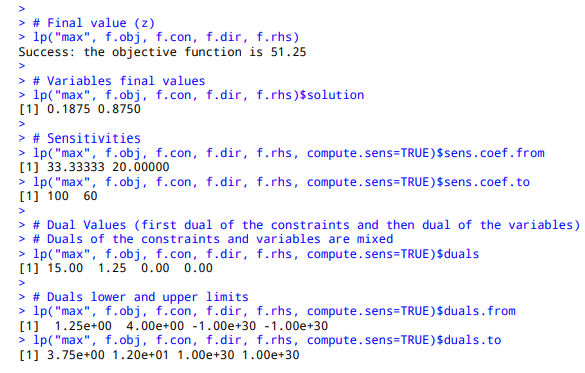
lp("max", f.obj, f.con, f.dir, f.rhs, compute.sens=TRUE)$sens.coef.from

lp("max", f.obj, f.con, f.dir, f.rhs, compute.sens=TRUE)$sens.coef.to

lp("max", f.obj, f.con, f.dir, f.rhs, compute.sens=TRUE)$duals

lp("max", f.obj, f.con, f.dir, f.rhs, compute.sens=TRUE)$duals.from

lp("max", f.obj, f.con, f.dir, f.rhs, compute.sens=TRUE)$duals.to



**Practical 9 :-** TRANSPORTATION PROBLEM

**Code & Output :-**

#Solve the following Transportation Problem in which cell entries represent unit costs using the R programming language

# "Customer 1", "Customer 2", "Customer 3", "Customer 4" SUPPLY

#SUPPLIER 1 10 2 20 11 15

#SUPPLIER 2 12 7 9 20 25

#SUPPLIER 3 4 14 16 18 10

#DEMAND 5 15 15 15 TOTAL = 50

library(lpSolve)

costs <- matrix(c(10, 2, 20, 11,

2, 7, 9, 20,

4, 14 , 16, 18), nrow = 3, byrow = TRUE)

colnames(costs) <- c("Customer 1", "Customer 2", "Customer 3", "Customer 4")

rownames(costs) <- c("Supplier 1", "Supplier 2", "Supplier 3")

row.signs <- rep("<=", 3)

row.rhs <- c(15, 25, 10)

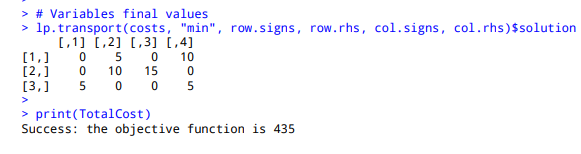
col.signs <- rep(">=", 4)

col.rhs <- c(5, 15, 15, 15)

TotalCost <- lp.transport(costs, "min", row.signs, row.rhs, col.signs, col.rhs)

lp.transport(costs, "min", row.signs, row.rhs, col.signs, col.rhs)$solution

print(TotalCost)



**Practical 10 :-** ASSIGNMENT PROBLEM

**Code & Output :-**

#Solve the following Assignment Problem represented in the following matrix using R Programming

# JOB\_1 JOB\_2 JOB\_3

#W1 15 10 9

#W2 9 15 10

#W3 10 12 8

library(lpSolve)

costs <- matrix(c(15, 10, 9,

9, 15, 10,

10, 12 ,8), nrow = 3, byrow = TRUE)

costs

lp.assign(costs)

lp.assign(costs)$solution

