Optimization Methods for Civil Engineering

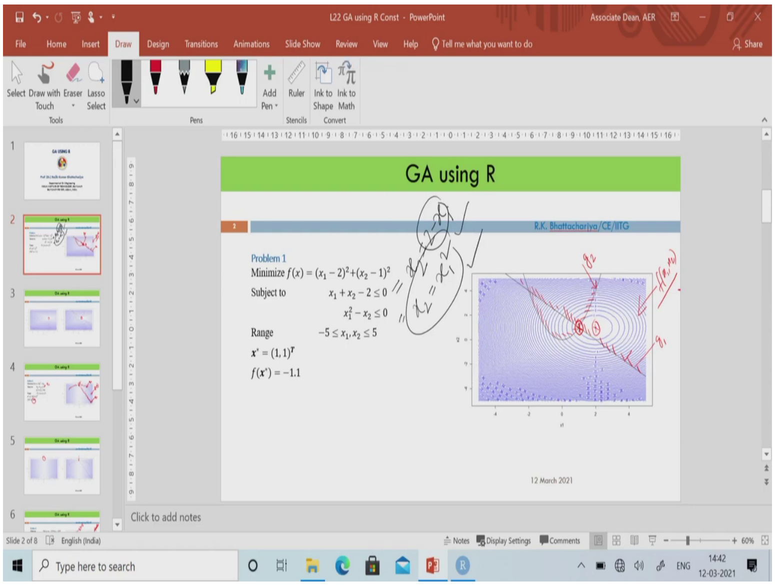
Prof. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 24
GA using R (Constrained problem)

Welcome back student. In the last class, we have solved some problem, mainly the unconstrained optimization problem. So, all are non-linear optimization problem and we have solve using GA, Genetic Algorithm. So, we have solved this problem using GA in our platform.

So, today we will solve some problem, mainly constraint optimization problem. So, we will also discuss how we can handle a constraint in genetic algorithm. So, that we will discuss and we will solve few constraint optimization problem using genetic algorithm and in R platform.

(Refer Slide Time: 01:11)



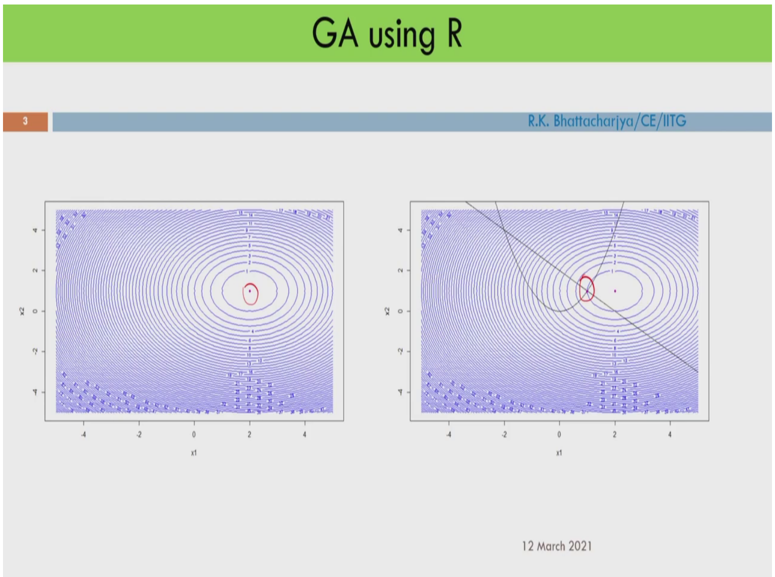
The first problem is, is a minimization problem and it is a function of two variables, that is x 1 and x 2. And the function is x 1 minus 2 whole square plus x 2 minus 1 whole square and we have two constraints; the first constraint is x 1 plus x 2 minus 2 less than equal to 0, and the second one is x 1 square minus x 2 less than equal to 0. And the domain is that x 1 x 2 is between minus 5 and plus 5.

So, now if I solve this problem, this unconstrained problem. So, as you can see I have shown you the contour plot. So, this is the objective function that contour plot I have shown. So, this is the objective function ok, this is the function f of x 1 and x 2. So, this is the plot of this particular function. And the unconstrained solution is somewhere here, unconstrained solution is somewhere here. So, this is the unconstrained solution and the solution is your 2 and 1. So, this is the unconstrained solution.

Now, once you are putting these two constraint; the first one is this particular line and which is basically x 1 plus x 2 minus 2 less than equal to 0. So, therefore, so anything in this side ok; so anything in this side will now be infeasible, ok. So, this is the infeasible region, ok. So, therefore, the earlier solution is not a feasible solution. So, this is your constraint one, so I can say that this is g 1 and similarly if I plot the second constraint, so this is your g 2, ok.

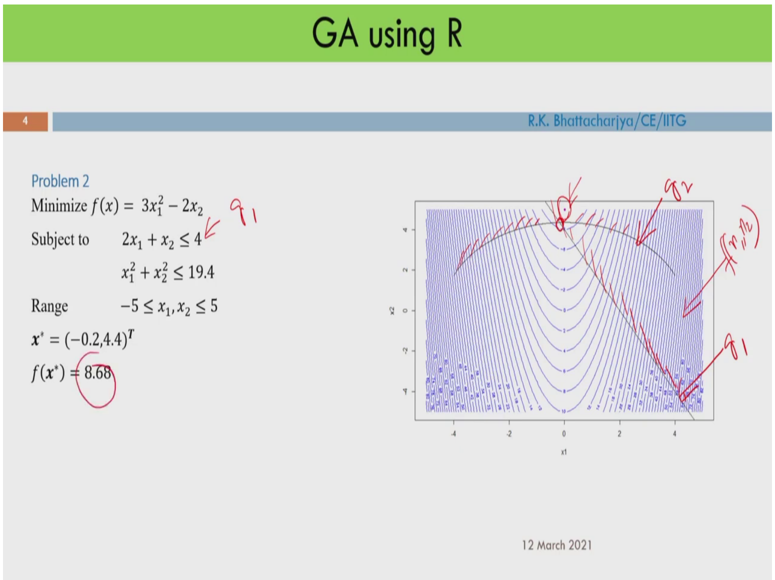
And now you can see, so anything in this particular region will be your infeasible. So, therefore, the feasible solution that the optimal solution should be somewhere here so this is the optimal solution of the constraint problem. So, we will try to see. So, I will show you that how you can find out the unconstrained solution and then how will put constrained in genetic algorithm; let us see whether we are getting this particular solution or not. So, this is the first problem, so I will show you and I should get this particular solution, ok.

(Refer Slide Time: 03:37)



So, you can see, so this is I have shown here. So, this is the unconstrained solution and this is the constrained solution of this particular problem, ok.

(Refer Slide Time: 03:48)



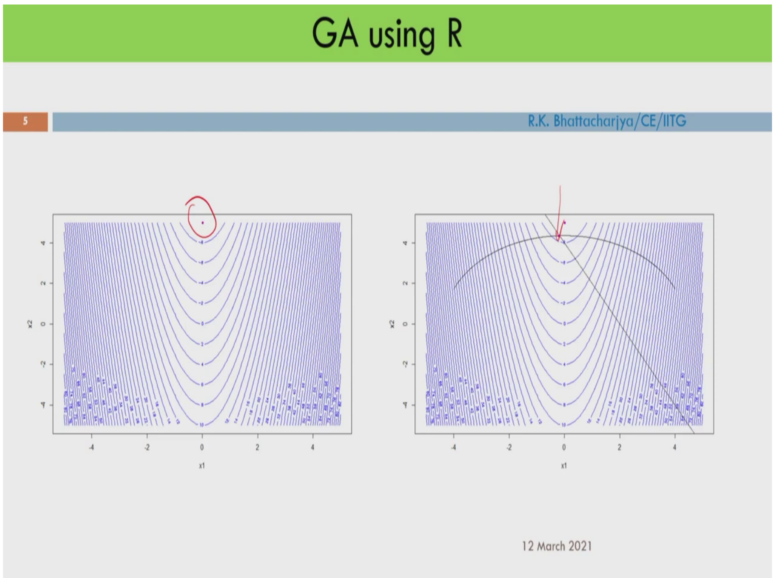
The second problem is again this is a problem of two variables that is x 1 and x 2 the objective function is thrice x 1 square minus twice x 2. And so, this is the contour plot of this particular objective function, so this is a function of x 1 and x 2.

And if I solve this problem; because I am minimizing this problem, so therefore, I should get these particular solution. So, this is the unconstrained solution of this particular function; but there are two constraint, the first constraint is twice x 1 plus x 2 less than equal to 4. So, this is the constraint. So, if I say this is g 1, so g 1 is somewhere here. So, this is your g 1.

And the second one is x 1 square plus x 2 square less than equal to 19.4. So, this is the second constraint g 2, ok. So, any solution in this particular region will be infeasible region. So, therefore, the earlier solution is not a feasible solution, that is an infeasible solution; because it has it will violate that constraint and so, I need to find out the constrained solution of this particular problem, the constrained solution should be somewhere here.

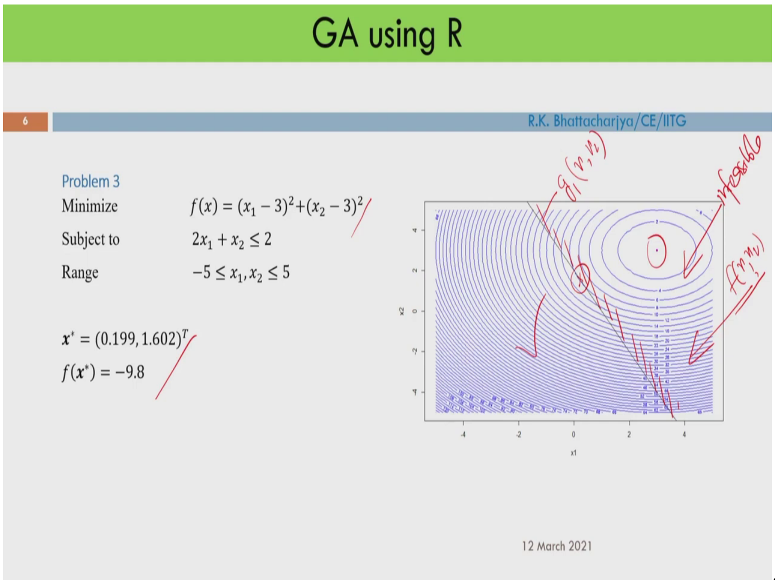
So, if I get the constrained solution, so solution is minus 0.2 and 4.4 and objective function value is 8.68. So, that is the solution.

(Refer Slide Time: 05:22)



So, I have shown here. So, this is the unconstrained solution of this problem and this is the constrained solution of this particular problem. So, we will try to see and we will basically apply genetic algorithm and try to see whether we are getting this particular solution or not.

(Refer Slide Time: 05:38)

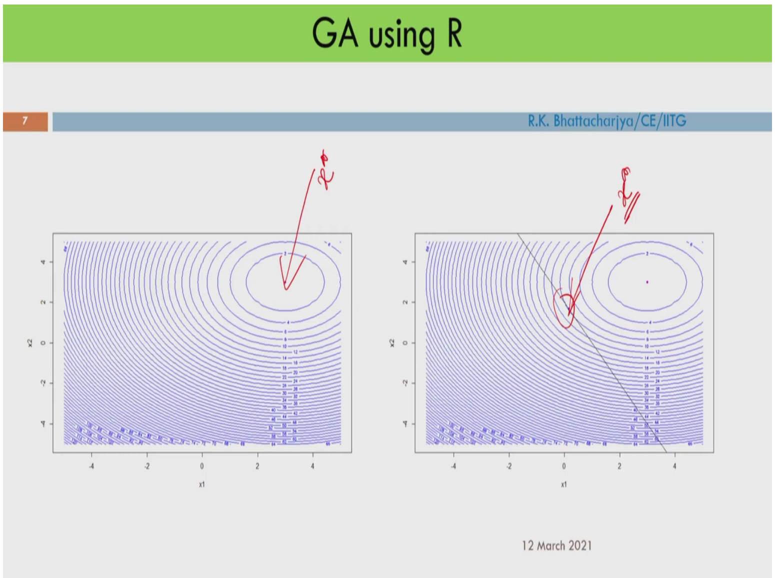


The third problem is also a minimization problem and it is a two variable function and the function is x 1 minus 3 whole square plus x 2 minus 3 whole square and this is the contour plot of this particular function ok, so x 1 and x 2, so this is the function. And unconstrained solution of this problem minimum point is these and that is 3 and 3; however there is a constraint and the constraint is twice x 1 plus x 2 less than equal to 2. And if I plot this particular line, so this is the line. So, we have only one constraint, suppose g 1 and it is the function of x 1 and x 2.

And now here this side is infeasible side, because twice x 1 plus x 2 less than equal to 2; so therefore, this is an infeasible side, so this side is infeasible. And this is your feasible side, so therefore, the solution. So, if I find out the minimum point, so it should be somewhere here, the solution should be somewhere here. So, I have shown the solution here, the solution is

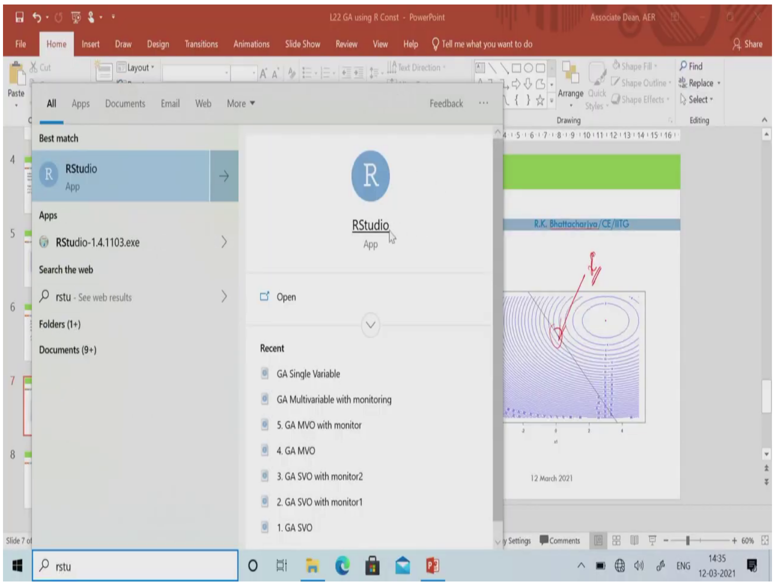
0.199 and 1.602; that means x 1 equal to 0.199 and x 2 equal to 1.602 and the function value is minus 9.8. So, that is the minimum function value.

(Refer Slide Time: 07:09)



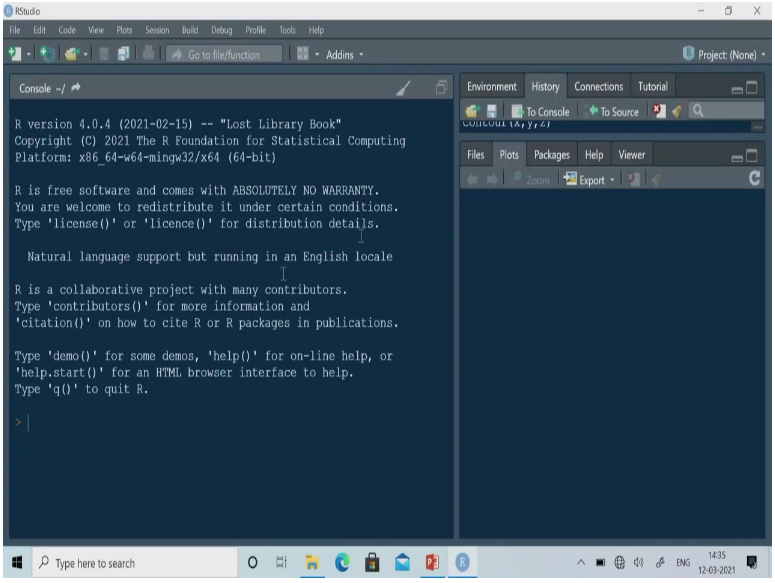
So, I have shown you the solution here. So, this is the unconstrained solution of this particular. So, this solution that is your x star ok; but constrained solution is somewhere here. So, this is the x star of the constrained solution. So, we will apply genetic algorithm and try to get the constrained solution. So, let us solve this problem using R. So, I will open R studio.

(Refer Slide Time: 07:35)



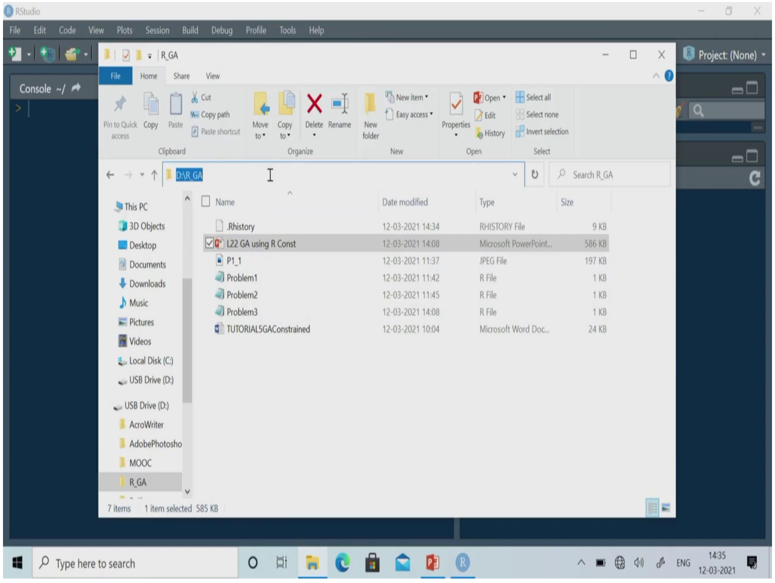
So, you just write here R studio.

(Refer Slide Time: 07:42)



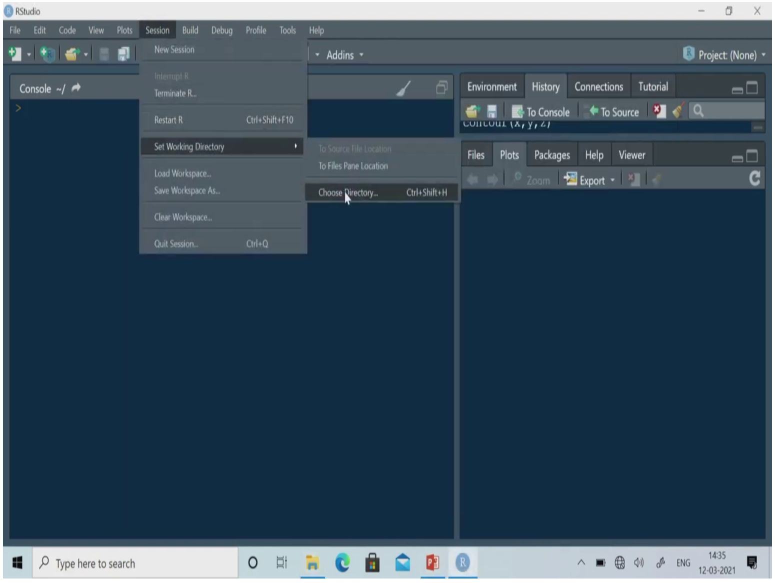
So, I will open that one. So, this is the R studio.

(Refer Slide Time: 07:48)



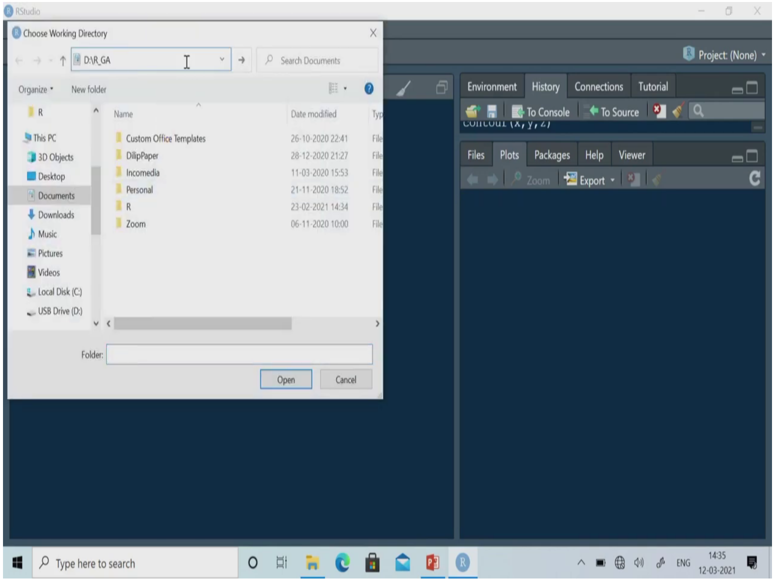
And my files are somewhere here. So, this is the file I have, actually I already wrote this code. So, I will open. So, this is the directory I should put.

(Refer Slide Time: 07:58)



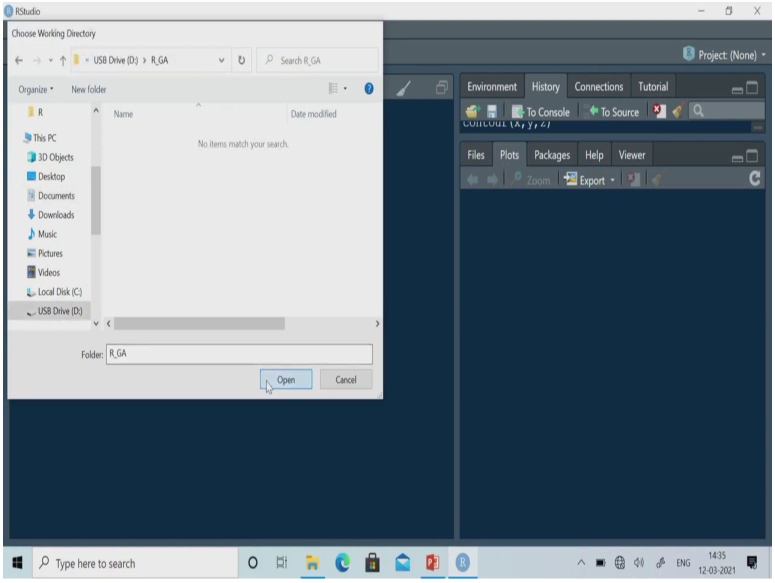
So, I am just choosing the working directory.

(Refer Slide Time: 08:02)



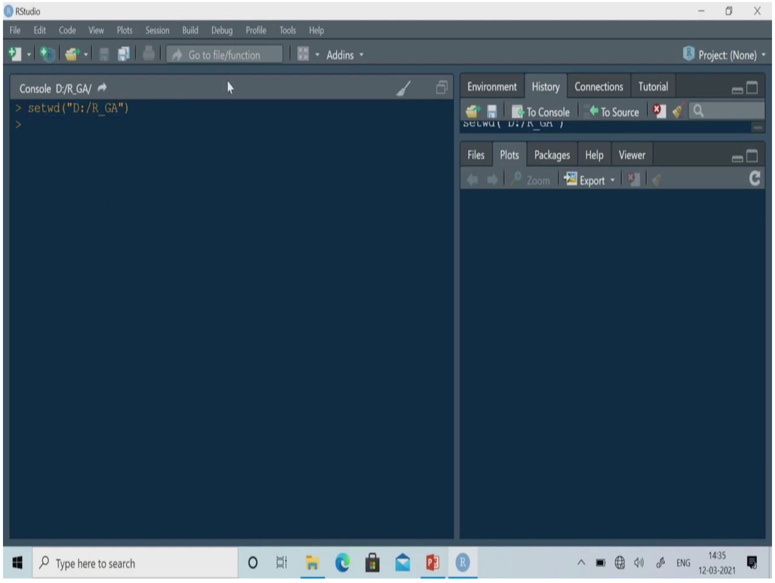
So, working directory is somewhere here.

(Refer Slide Time: 08:07)



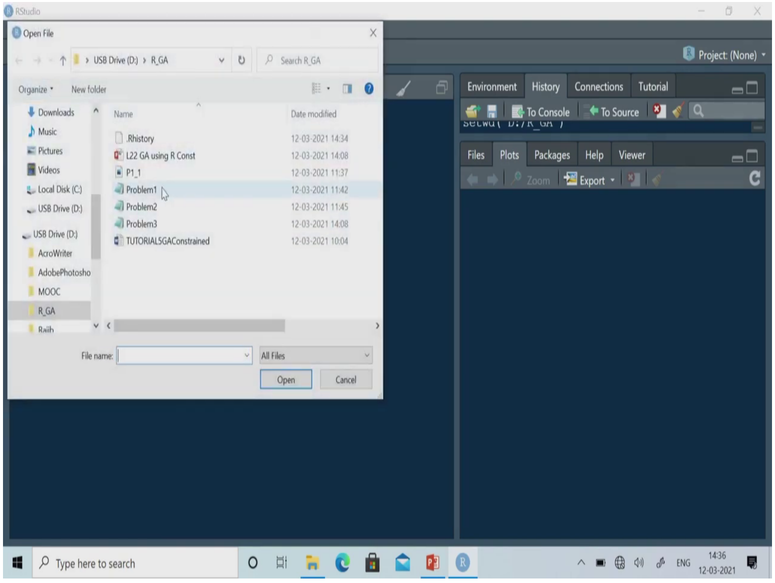
So, this is the working directory, ok.

(Refer Slide Time: 08:09)



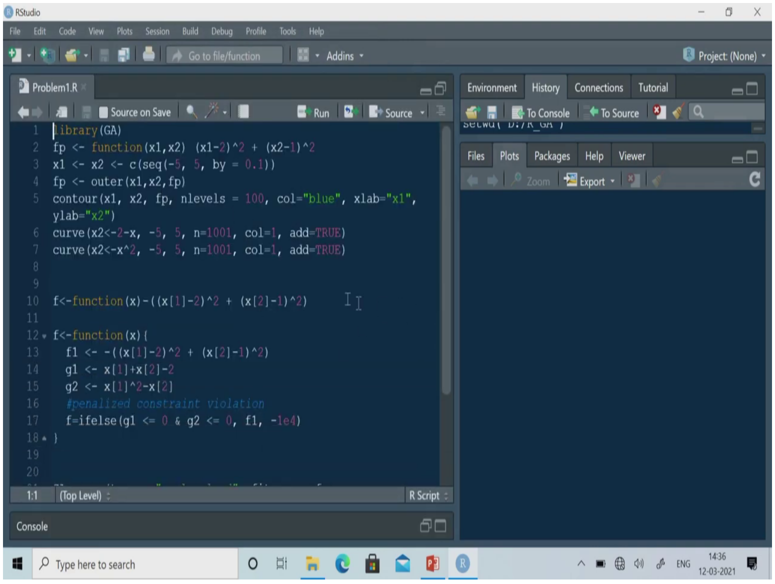
So, I can also set the working directory using this particular command.

(Refer Slide Time: 08:18)



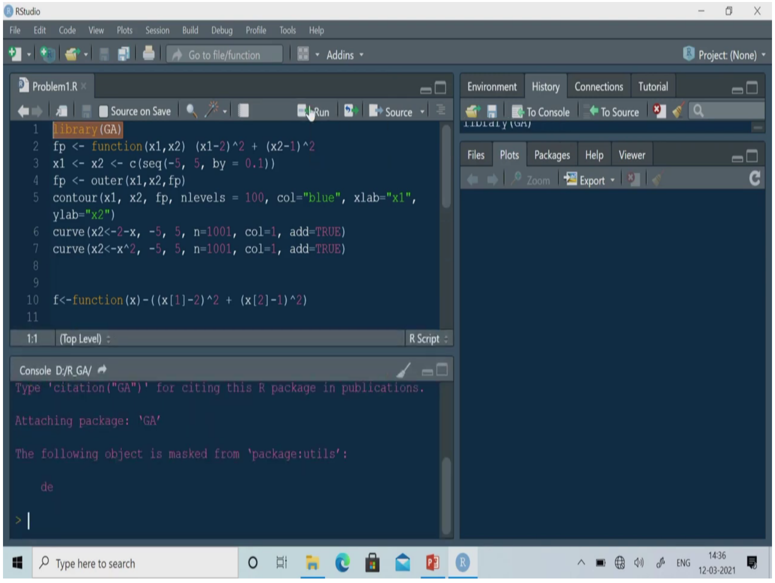
Now, let us obtain the problem 1.

(Refer Slide Time: 08:22)



So, this is your problem 1, the function is x 1 minus 2 whole square plus x 2 minus 1 whole square. So, this is the function. So, I think you are now you can write the function. So, this part again, the first part is just to plot this particular function and constraint. So, I will show you; you have to include the GA library, already you have install GA. So, next time you need not install just include the GA library. So, I can run this particular command GA library.

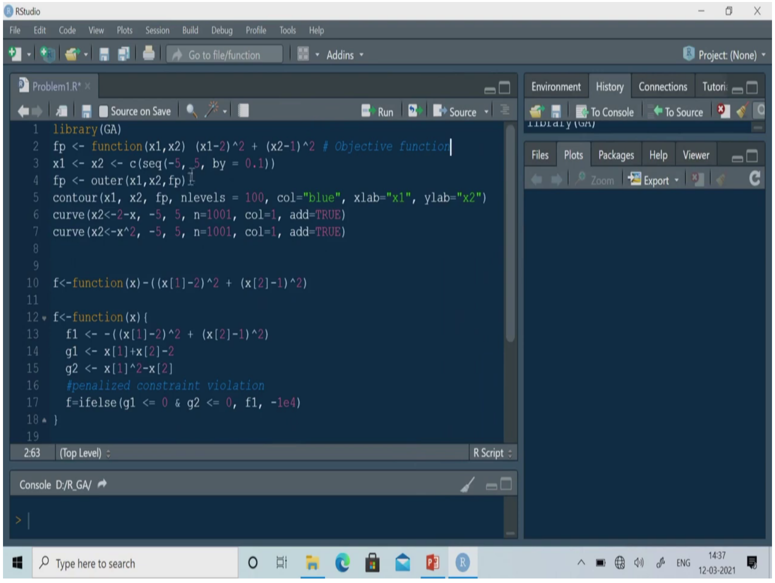
(Refer Slide Time: 08:55)



So, it will be loaded, ok. So, GA library is included here.

Now, next is I have to define this function, objective function.

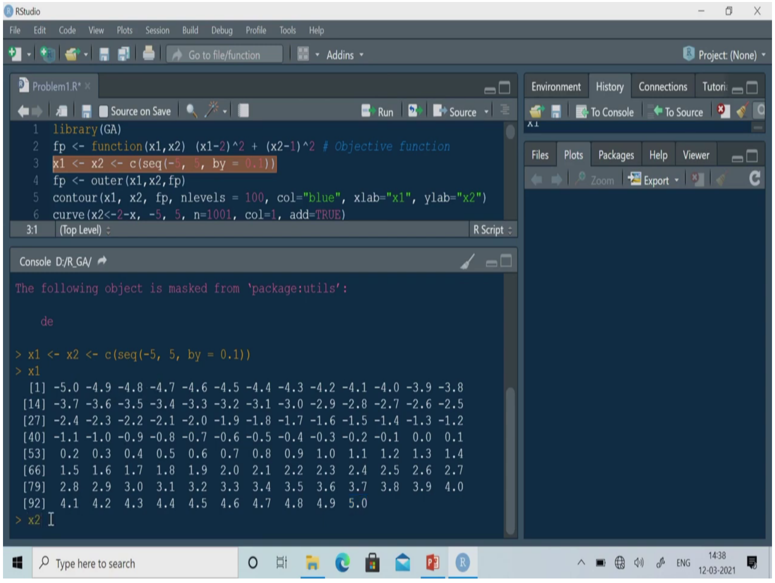
(Refer Slide Time: 09:04)



So, I can write that. So, this is your objective, objective function, ok. So, objective function is the objective function is that, it is a function of two variable x 1 and x 2. So, that is x 1 minus 2 whole square plus x 2 minus 1 whole square. So, this is the objective function, I think you can write it.

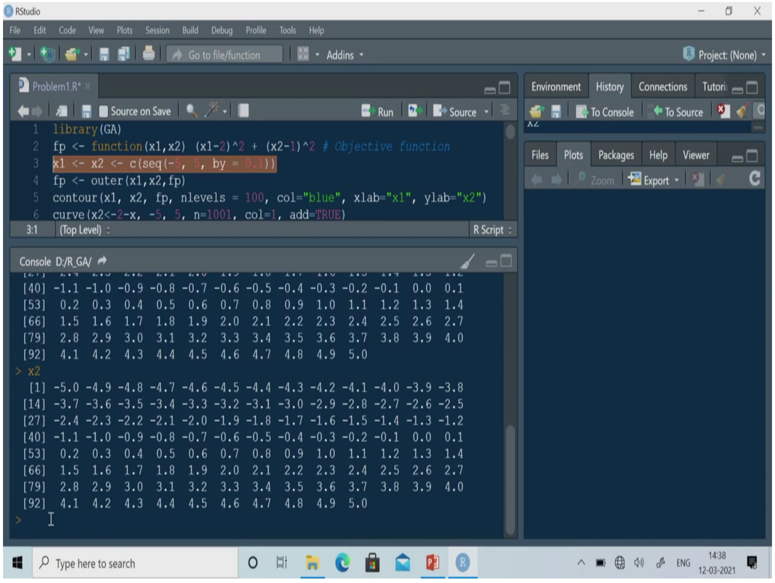
And then I am defining the sequence, so basically the value of x 1 and x 2; so x 1 is varying from minus 5 to plus 5 and I need a division of 0.1, that means it is varying from minus 5 and the interval between the two values is 0.1, ok. So, now, with this I can actually generate the data for x 1 grid, I generate the data for x 2 and x 1. So, this is x 1 and x 2 executed.

(Refer Slide Time: 09:58)



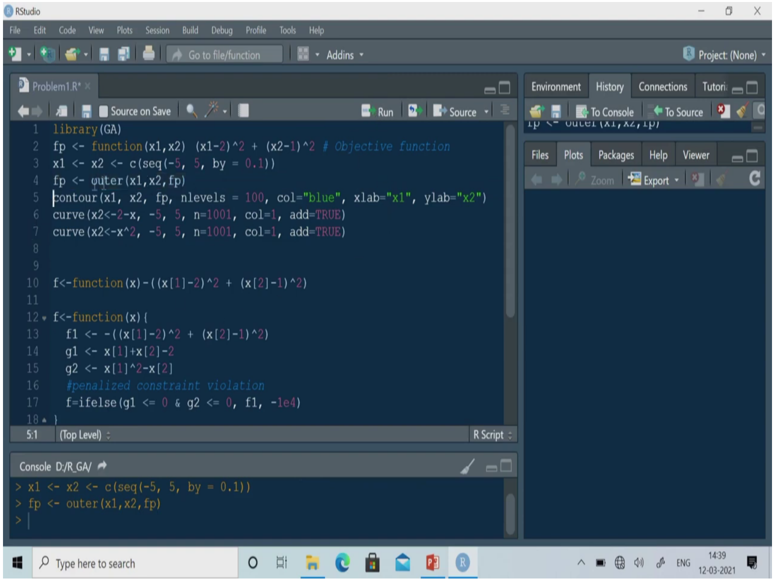
So, you can also see actually here by typing this. So, I am getting the x 1 value. So, this is the x 1 value you can see. So, this is from minus 5 to plus 5 with an interval 0.1, that is minus 5.0, then x is minus 4.9, then x is minus 4.8 something like that.

(Refer Slide Time: 10:23)



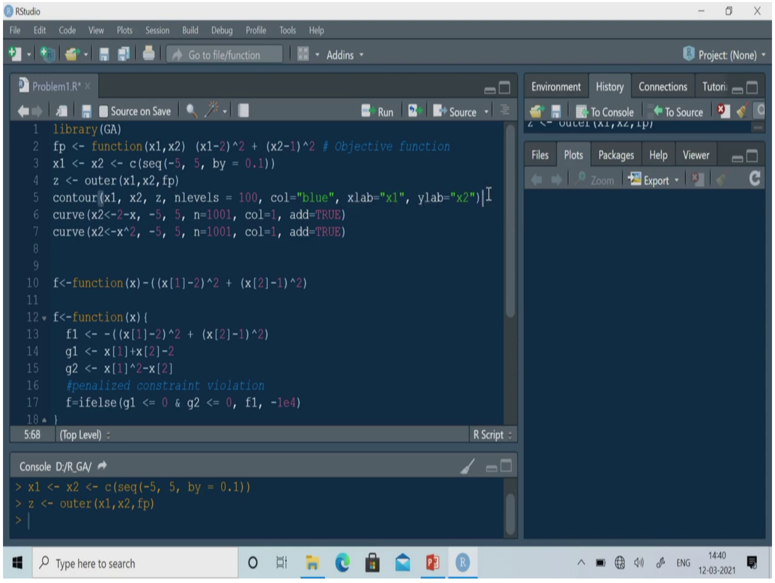
Similarly you can also see that I have the value of x 2 also, ok; so minus 0.5 to plus 0.5. So, by control L I can clean this, I can clean the console, ok. So, next is that, I would like to find out the function value at every x 1 and x 2 points. So, this is the outer function I am using and here you have to define x 1 and x 2. So, I already generated the value for x 1 and x 2 and f p I am that function I am putting here. So, then this will create, this will create the function value at its grid point. So, let us execute this particular line.

(Refer Slide Time: 11:09)



So, let us execute this particular line. So, now, using this contour function, so I can plot the contour. So, here I have to define x 1, then x 2, then the f p, that is the value I have generated.

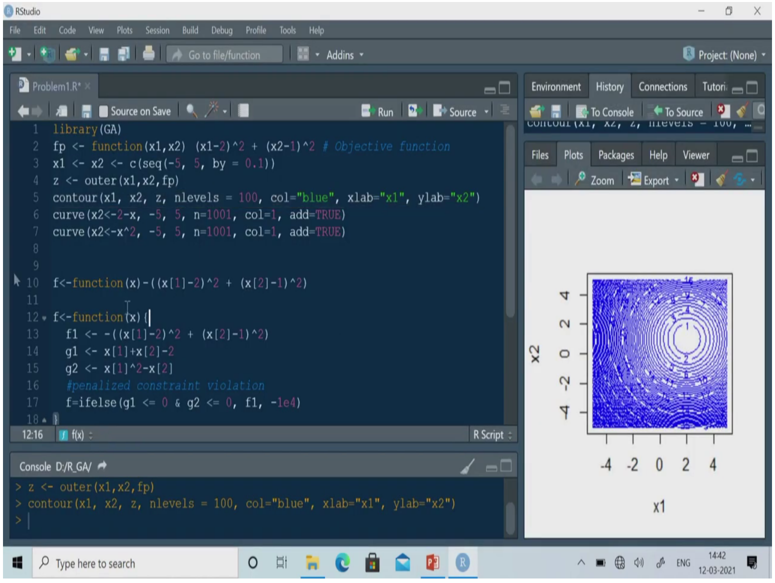
(Refer Slide Time: 11:25)



So, you can also write that these value is basically the z values, ok. So, I can also write is as a z, so x y z. So, let us execute and store in z, that is the x y z value. So, I am putting x 1, x 2 and z; then number of levels, I am putting 100. So, if you want to make more level, so you can do that and if you want to make less level, that also you can do, but I am putting 100. Color I am using blue and x, I am putting x level and y level, ok.

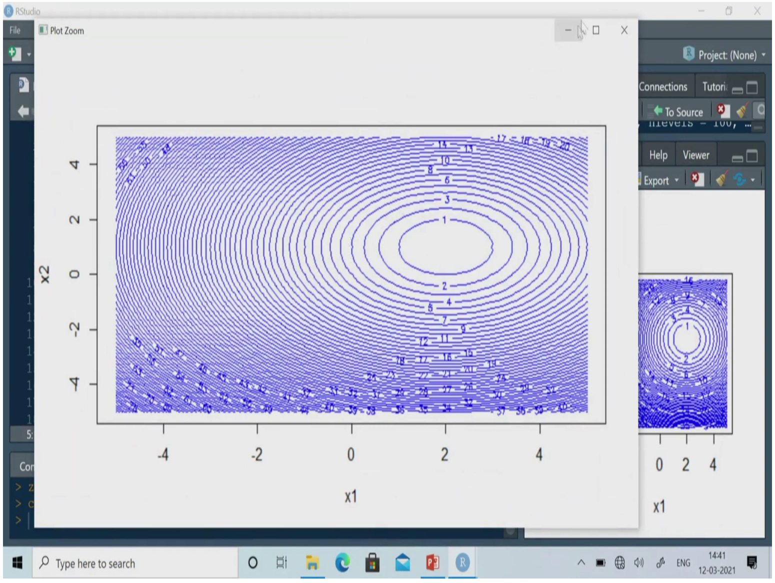
So, if I execute this particular line, then the contour should be plotted.

(Refer Slide Time: 12:01)



Let us see. So, I am getting, I am getting this. This is the contour, ok.

(Refer Slide Time: 12:04)

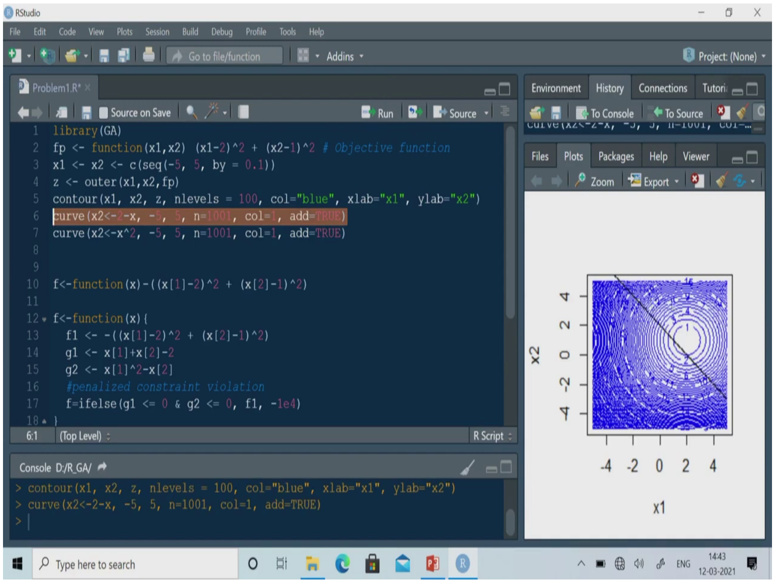


So, you can see that, this is the contour of this particular function I am getting and unconstrained optimal will optimal solution is somewhere here that is 2 and 1. So, somewhere here I should get that solution. Now, let us put the constraint. So, you just see the constraint is that, I am writing in terms of x 2 now.

So, if you go to the constraint, the constraint was something like this. So, this is the constraint that x 1 plus x 2 minus 2 is less than equal to is less than equal to 0. So, what I can do basically? So, I can write this particular constraint something like that x 2 is equal to 2 minus x, ok. So, x 1, but I am writing here as a x ok, so 2 minus x 1. And similarly these particular constraint, so I can write that x 2 equal to; so I would like to plot the line, so therefore I am writing x 2 equal to x 1 square, ok.

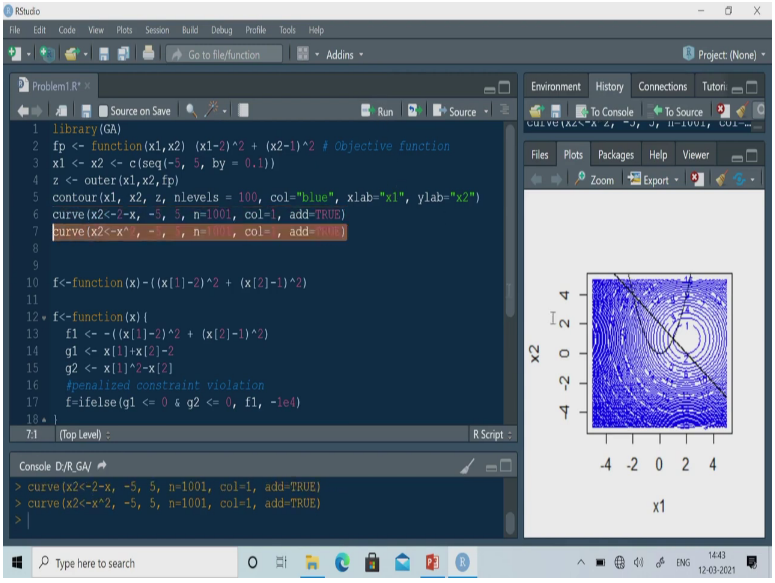
So, let us write this. So, this one is your g 1, this is g 1 and this is g 2. So, therefore, I am writing what is the first constraint is x 2 equal to 2 minus x, ok. So, I am writing x here and similarly the second constraint is x 2 equal to x square. And the range is minus 5 to plus 5, then I am putting the number of interval 1 0 1 in order to get a smooth curve and color I am putting 2, sorry color I am putting 1; that means black color and add tTRUE, that means I would like to add in this particular plot itself over the contour plot.

(Refer Slide Time: 14:00)



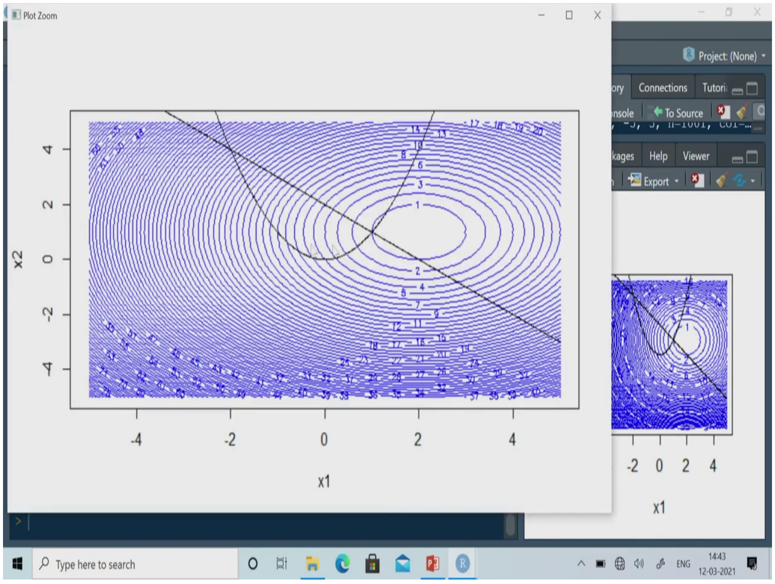
So, let us execute this particular line, then I should get this particular constraint.

(Refer Slide Time: 14:08)



So, now, this is the constraint I am getting and similarly if I run the other constraint, then I should get this particular. So, I am getting this constraint.

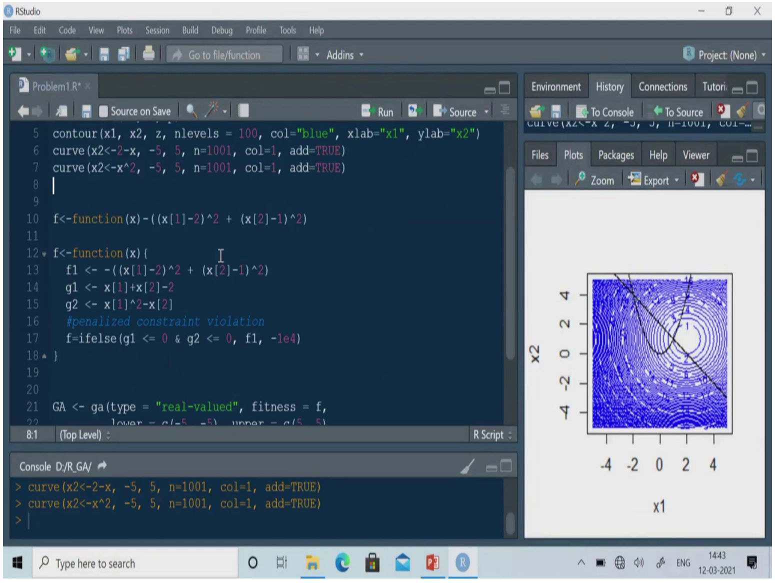
(Refer Slide Time: 14:13)



So, now, if you look at, so if you look at; so this is the original optimal solution. So, original optimal solution is somewhere here 2 and 1; but this is not feasible solution now, the constrained solution is somewhere here, ok. So, you should get this particular solution.

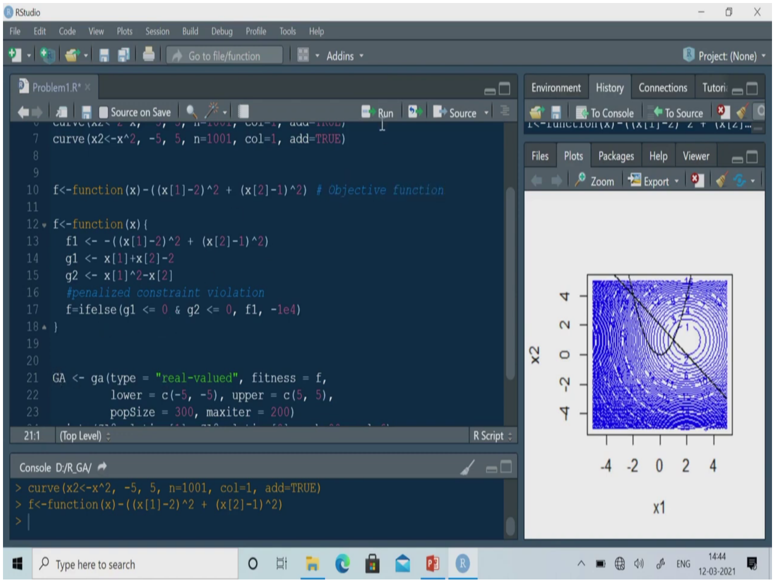
So, what I have done? So, this part whatever I have executed; so just to plot this particular function, just to plot this particular function and function as well as the constraint.

(Refer Slide Time: 14:45)



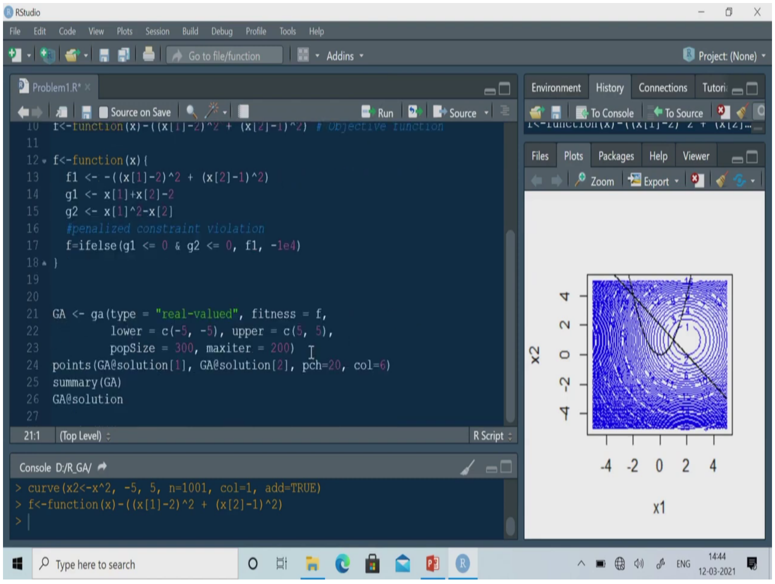
So, this is, now let us solve the unconstrained problem first, ok. So, I am defining my function objective function. So, I am defining it. So, object objective function is x 1 minus 2 whole square plus x 2 minus one whole square the solution is 2 and 1. So, let us run this particular, ok.

(Refer Slide Time: 15:05)



And then let us run the GA, ok. So, this is for the constraint part anyway. So, this is I am doing for unconstrained one. So, this is the objective function. So, you can write this, this is the objective function only, ok. So, I have executed, now you execute the GA, ok.

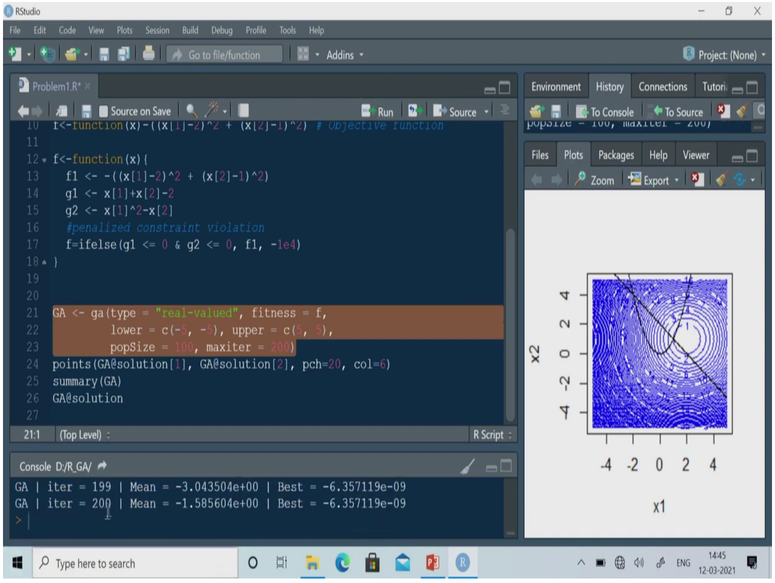
(Refer Slide Time: 15:27)



So, GA I am just running this particular command, this line, so it will execute GA. So, let us see. So, here population size I have put 300, then maximum iteration 200 anyway. So, if you are putting the default value, so default population will take and default maximum iteration will take; but in this case I have taken 300, but I can also take 100 also, there is no issue.

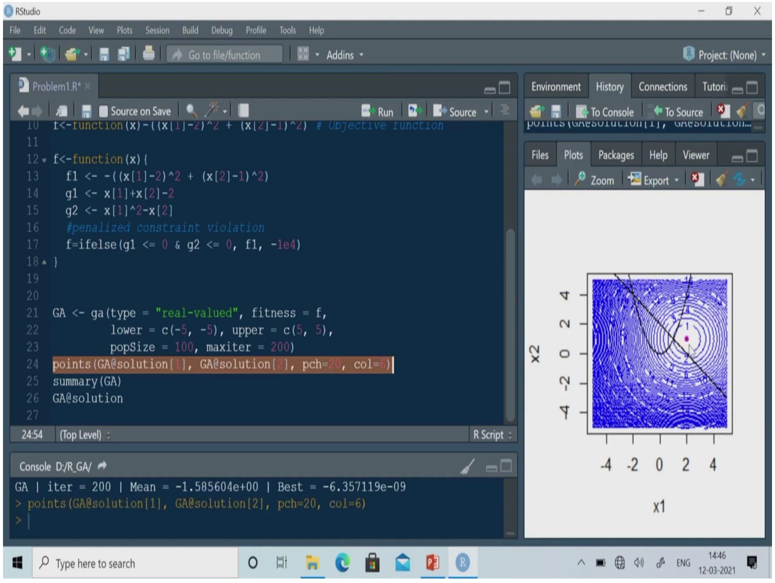
Probably I will get if I do not get the solution, you can increase the population size and iteration, so you can do that. And let us execute. And here what I have done? So, I have use the real valued GA. So, I have not used binary value till now. So, I use real valued and fitness is that f, so this f I have executed. So, I will I have executed this one and then this is lower bound and upper bound I have defined and population size I have defined and maximum iteration 200 I have defined, ok.

(Refer Slide Time: 16:28)



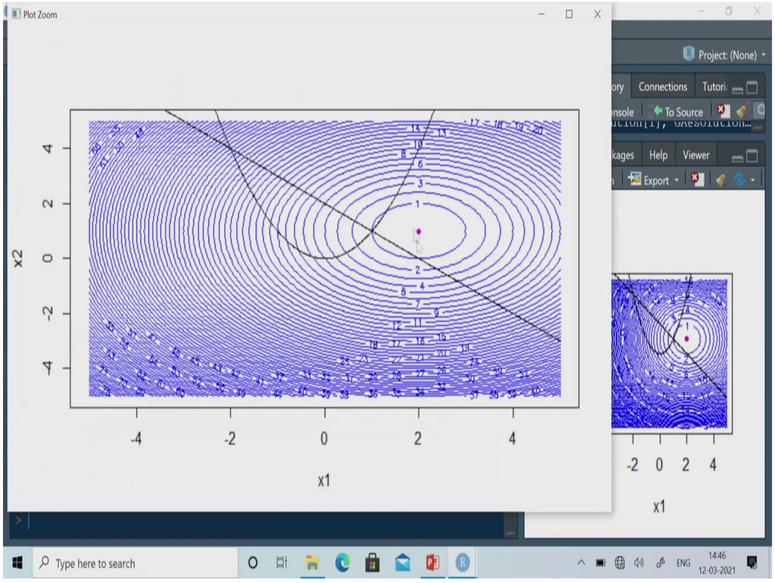
So, let us execute this particular line. So, it will execute GA, ok. So, up to 200 iteration, so I got the solution. Now, I would like to plot the optimal solution over this contour plot and that I can do using the point function. So, this is the function you can use. So, I am just putting the point, that is GA solution 1; that means first variable and GA solution 2, that is the second variable and then color I am putting 6 ok and let us see.

(Refer Slide Time: 16:29)



So, if I run this particular line, I should get the solution, the solution is somewhere here.

(Refer Slide Time: 17:03)



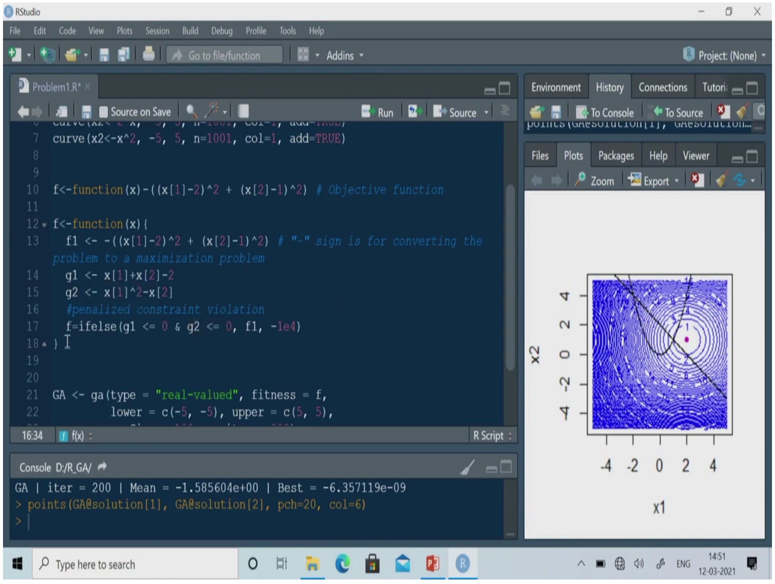
So, I can show you that I am getting the unconstrained optimal solution. So, this is the unconstrained optimal solution; constrained solution is somewhere here, so let us see how I can find out the constrained solution. So, for constrained solution, so I have to add the constraint with the objective function value. So, therefore, I have written a function.

So, if it is more than one line, then you have to start your function with the second bracket and you have to close that with a second bracket; but in one line directly you can write, so it is a one line function. But if you have more than one line, then second bracket you have to use. So, and you have to open it using second bracket and then you have to close it.

Now, this is the objective function, ok. So, in this case objective function is that is x 1 minus 2 whole square plus x 2 minus 1 whole squared. But before that I would like to tell one thing; so GA generally solve the maximization problem, but in your case this problem is a minimization problem. A minimization problem can be converted to a maximization problem by multiplying minus 1. So, therefore, I have used this minus 1 just to convert the problem to a maximization problem, ok.

So, therefore, I have use this minus your 1, ok. So, now, objective function is minus. So, you if you look at the original problem is, there is a minimization problem and minimization of these particular function; but if I want to maximize, so therefore just I have to put minus 1.

(Refer Slide Time: 18:40)



So, this minus 1 I can write; so this minus sign is for converting the problem to a maximization problem, ok. So, minus sign is for converting the problem to a maximization problem; then you write g 1, so g 1 is x 1 plus x 2 minus 2. So, that is your g 1 and g 2 is x 1 square minus x 2, ok.

So, this is the constraint and this g 1 and g 2 value should be less than equal to 0, ok. So, therefore, what we have done here, so I have used if else function. Now, question is that, if g 1 and g 2 are less than 0; that means that is not a violation, so whatever solution you are getting that is on the feasible side.

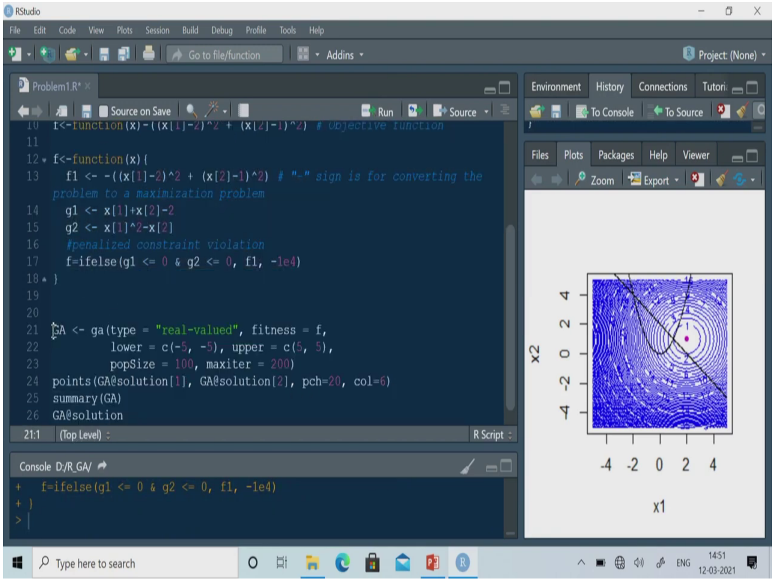
So, therefore, if g 1 is less than 0 and I am using n function, because both has to be true, that means g 1 is less than 0 and g 2 is also less than 0; then what this f value will be? The f 1, f 1 means the objective function value. Otherwise, what will happen; because I am maximizing this problem, so otherwise. So, I am putting a negative value, negative means large negative value ok, 1 e to the power 4, ok. So, 10 to the power 4, so it is a large value I am putting.

So, therefore, if there is a violation, then this f value will be very less ok, 10 to the power minus your 4. So, in that case what will happen; when the solution will go through the selection operator, then because the objective function value is very less in terms of 10 to the power minus 4, that in the tournament selection this solution will not go to the next generation.

So, therefore, I am just using this if else functions; if there is a violation that is g 1 is less than 0 and g 2 is less than 0, then f will be negative value, that is a very less value and we are maximizing that one. So, in the selection operator will not select this particular solution; but if there is no violation, then it will return the objective function value.

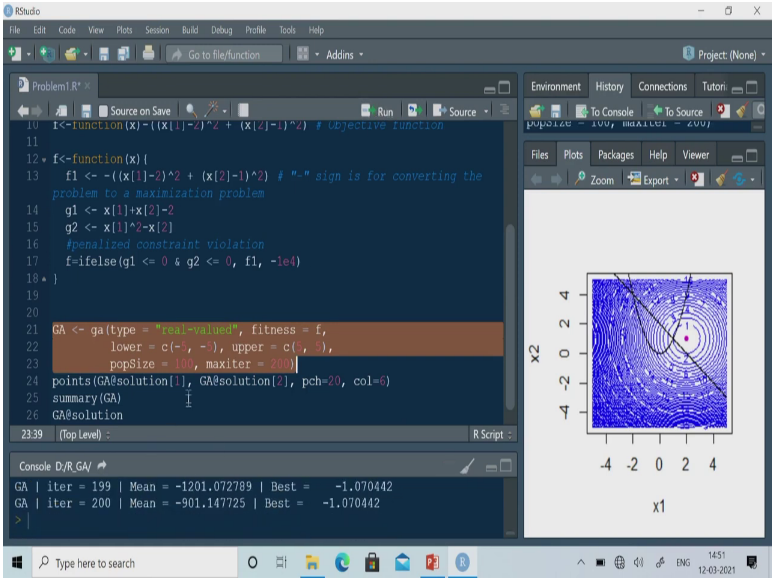
So, therefore, that way I can implement constraint here. So, I have implement constraint something like that. So, let us execute this particular function by using run, ok.

(Refer Slide Time: 21:19)



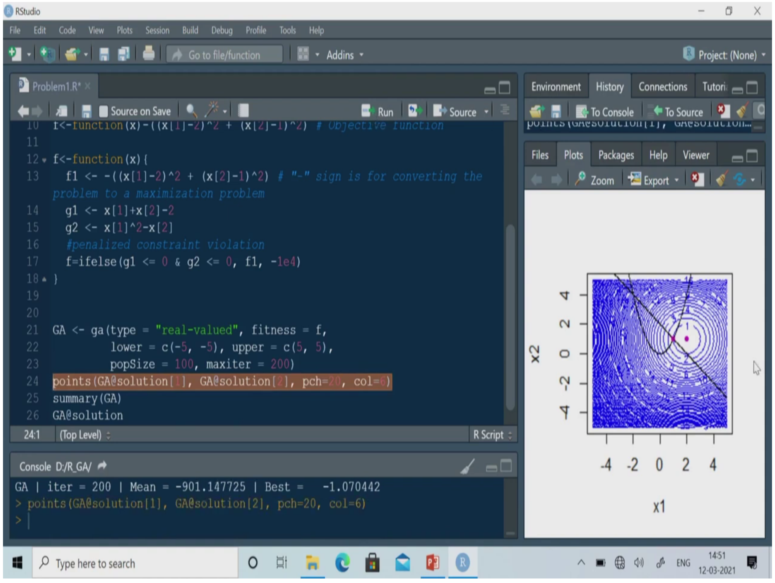
So, I have done that one. So, now, let us run this particular line to implement GA, to execute GA.

(Refer Slide Time: 21:32)



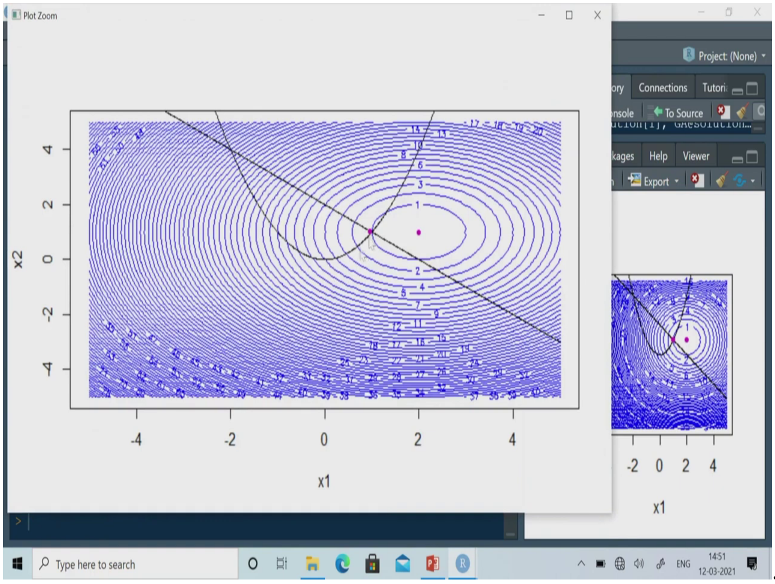
So, I am executing this one, ok. So, I got the solution and let us plot the solution now using this function. So, I can plot it.

(Refer Slide Time: 21:42)



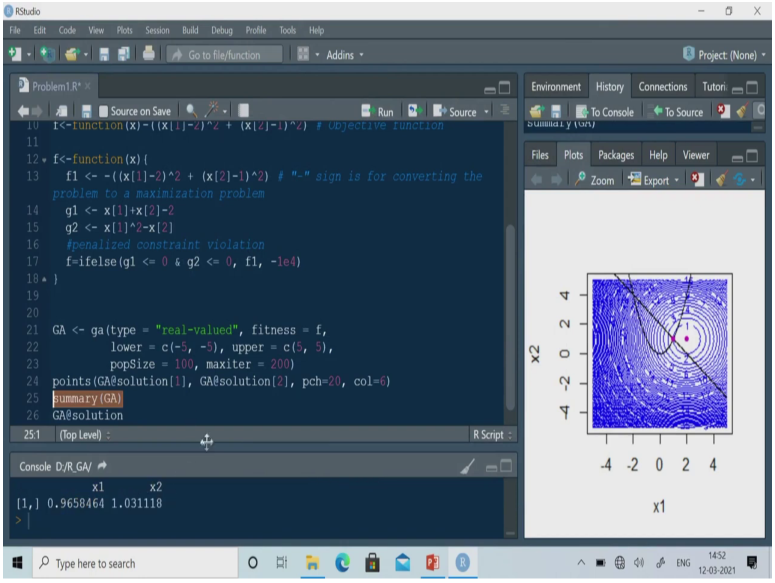
So, you can see that I have plotted and you can see from here.

(Refer Slide Time: 21:45)



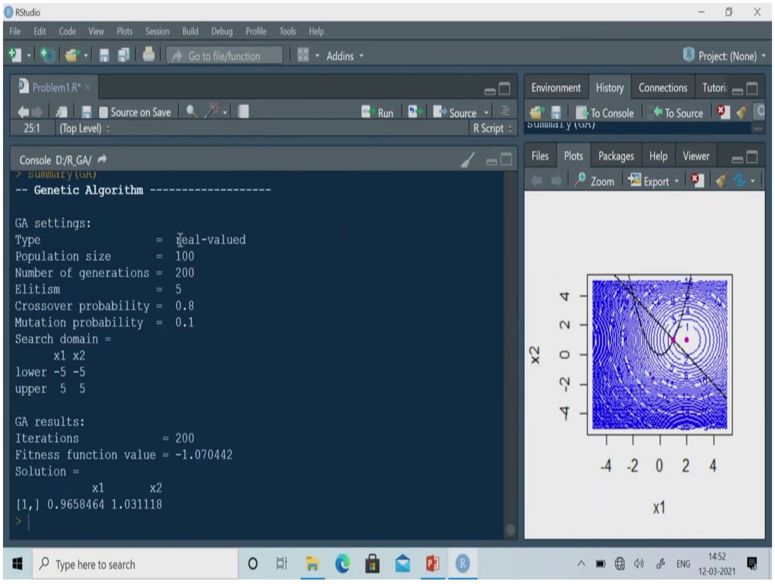
Now, I am getting this solution, ok. So, it may not be the exact optimal solution, but if you are increasing your iteration and population; so you may get the exact optimal solution, but you are getting a near optimal solution.

(Refer Slide Time: 22:06)



So, you can see by summary you can see actually the summary of GA.

(Refer Slide Time: 22:08)

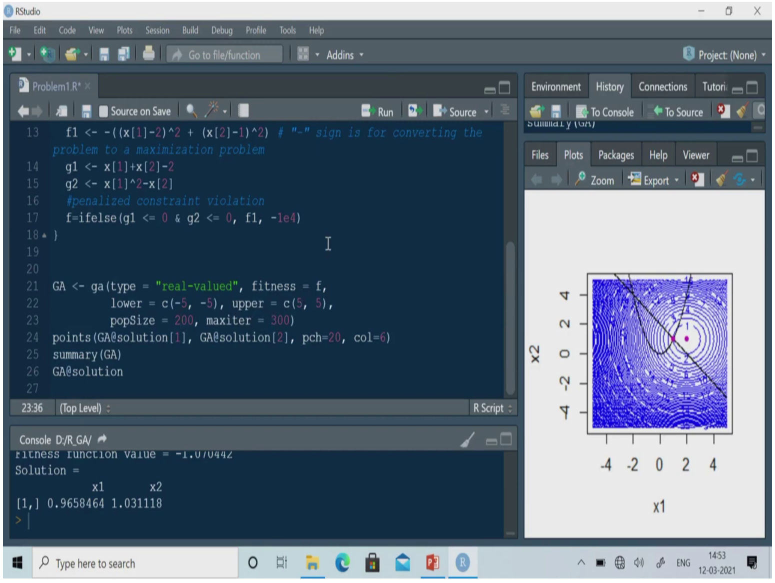


So, here everything is mentioned that, I have used real valued, population size is 100, number of generation is 200, elitism 5, crossover probability is 0.8, mutation probability 0.1; then upper bound lower bound is defined, iteration is 200 and fitness function value I got this one and this is the solution 0.9658464 and 1.031118. But this is not the optimal solution, but it is just near to the optimal solution.

When you are applying GA so you may not get the exact optimal solution because we are not using the gradient information so you may get a near optimal solution. So, in order to get the exact optimal solution, so from here I can apply a classical method, gradient based method and I can find out the exact optimal solution. And moreover, so if you are increasing our population size, number of generation, so I may get the exact optimal solution. So, I may try that one.

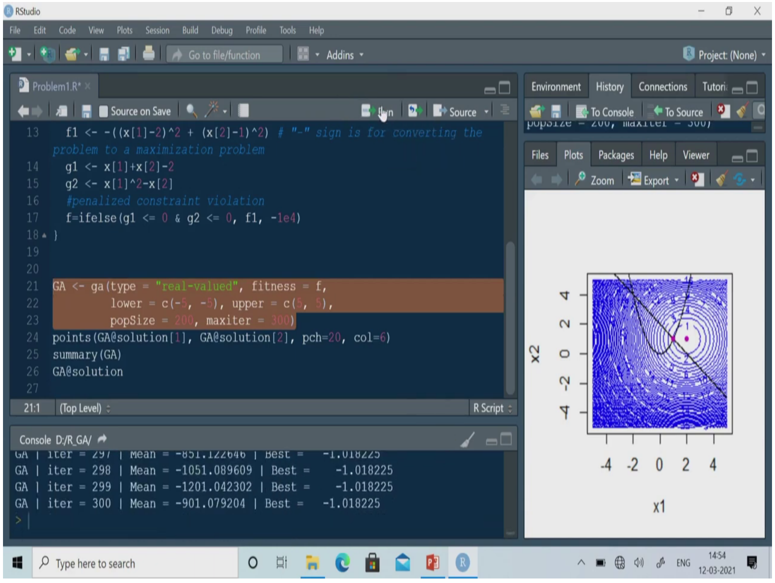
So, let us see what are the solution. Solution is 1 and 1; but I am not getting 1 and 1, I am getting 0.96 and 1.03. So, let us see if I increase my population size, then what will happen?

(Refer Slide Time: 23:37)



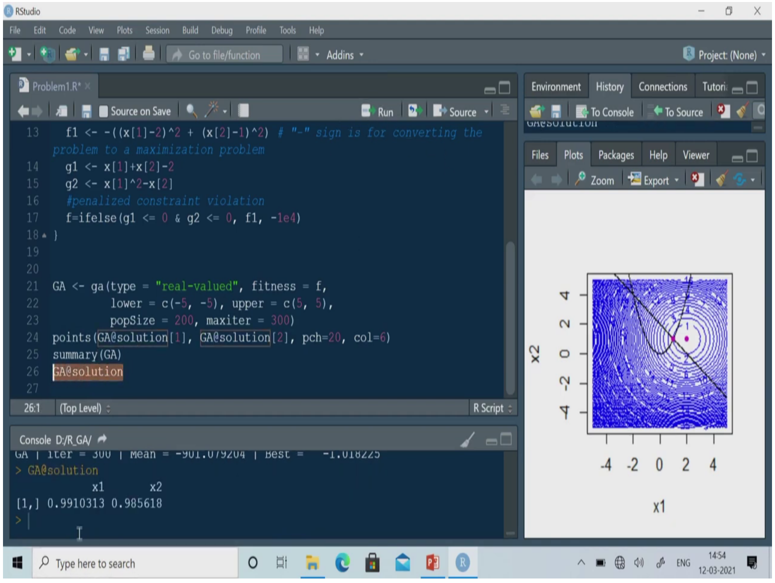
Suppose if I increase my population size to 200 ok and generation you make it 300; just see whether I am getting the exact solution that 1, 1, so I will execute this one.

(Refer Slide Time: 23:50)



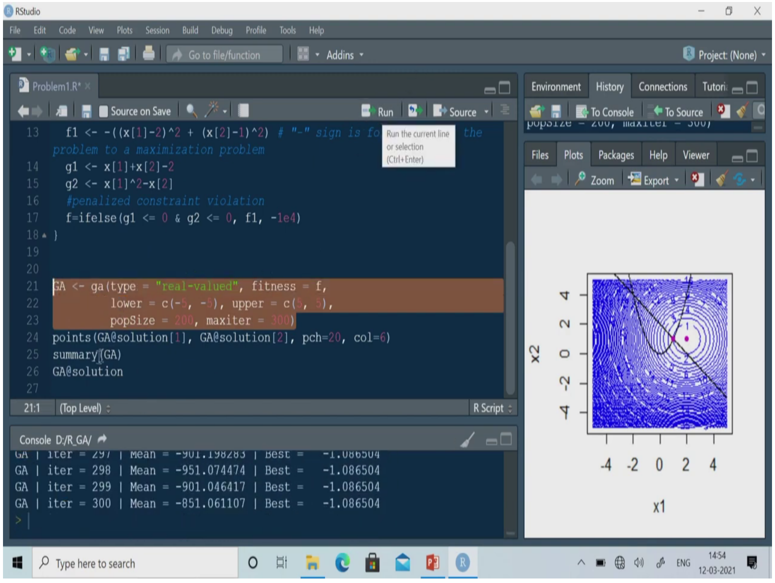
Now, generation is 300. So, up to 300 and population size is 200 and just see whether I am getting the solution or not.

(Refer Slide Time: 24:01)

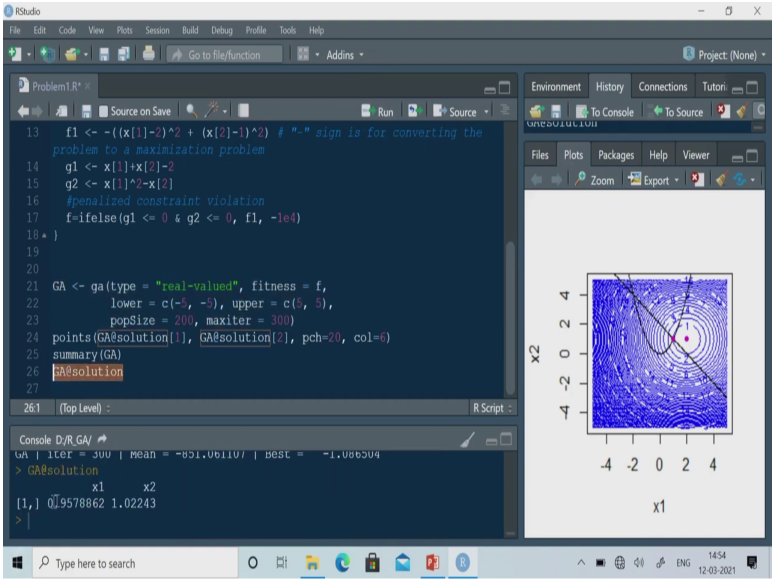


So, I am getting 0.99 and 0.98 that is close to the x l optimal solution 1, 1 anyway, so I can try again.

(Refer Slide Time: 24:15)



(Refer Slide Time: 24:21)



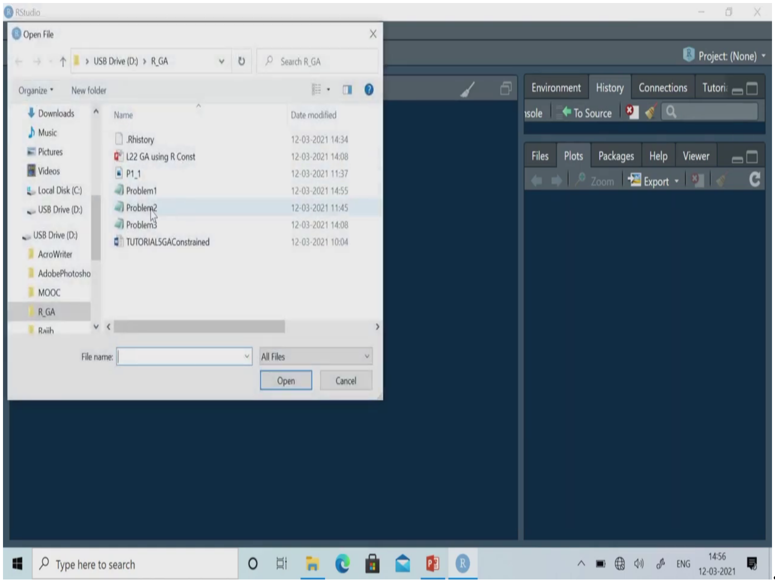
So, this time let us see. So, I am getting 0.95 and 1.02. So, therefore, the GA, in GA you may not get the exact optimal solution, but you will get a near optimal solution or near optimal means, I am telling near the global optimal solution of the problem.

So, as I said, so if you want to get the exact optimal solution, so after that; so using this solution that whatever base solution you are getting from GA, you can take that solution as a initial point and you can apply any classical method to get the exact optimal solution.

So, that we can do and this is known as hybrid optimization technique. So, therefore, I am using here two algorithms; one is the genetic algorithm and then whatever solution you are getting, the base solution you are getting from genetic algorithm, you take that solution as a starting point for your classical source and you will get the exact optimal solution of the problem.

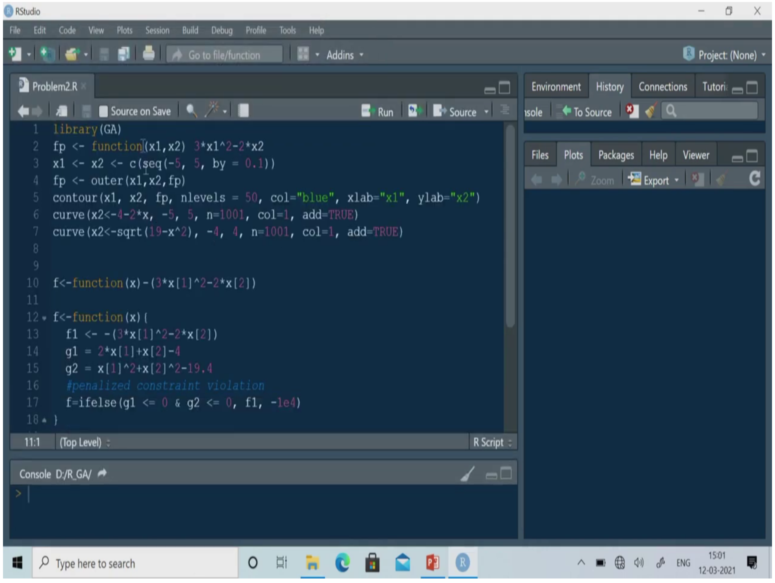
So, I will also show some example of hybrid method. So, where I will apply the classical as well as non-classical optimization your, where I will apply classical, both classical and non-classical optimization algorithm. Now, let us see the second example problem. So, I would like to clean everything. So, here I would like to clean the figure, then clean the console and whatever data stored here. So, I would like to clean it and then I will also like to close this particular problem.

(Refer Slide Time: 26:02)



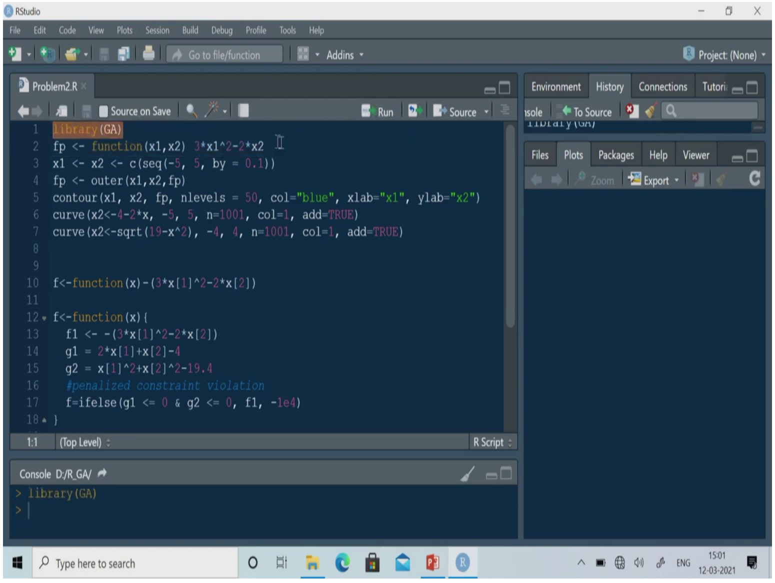
So, let us open the second one that example 2, ok.

(Refer Slide Time: 26:07)



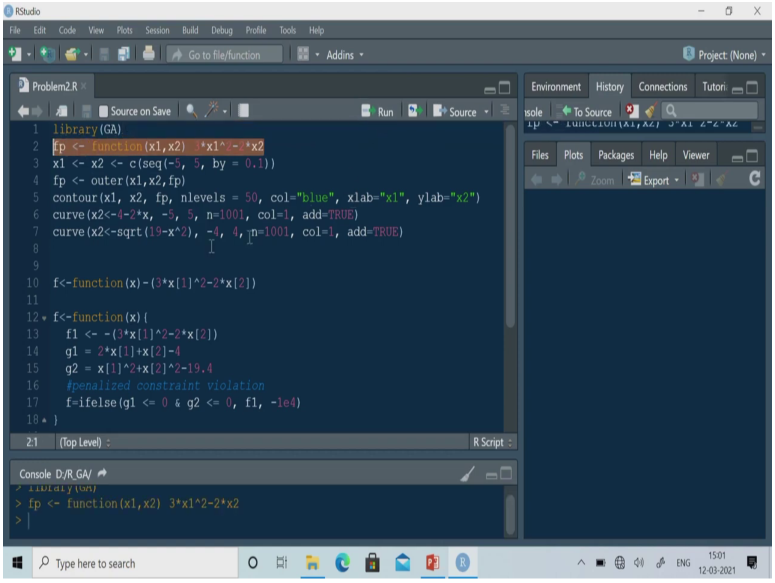
So, here the function is thrice x 1 square minus twice x 2. So, this is the function. So, I am writing it, just to plot that one; so I am writing it f p and this is a function of x 1 and x 2 and the function is thrice x 1 square minus twice x 2. So, I am executing this one.

(Refer Slide Time: 26:35)



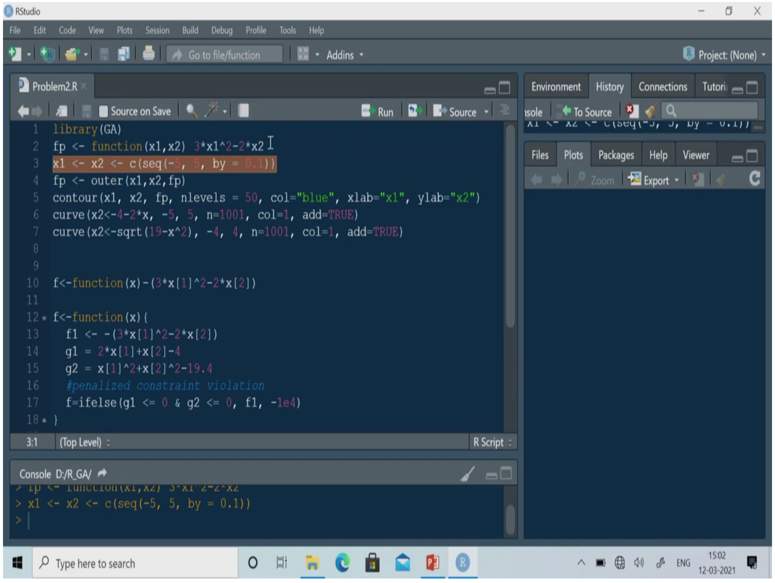
So, before that let me run the library, GA library.

(Refer Slide Time: 26:40)



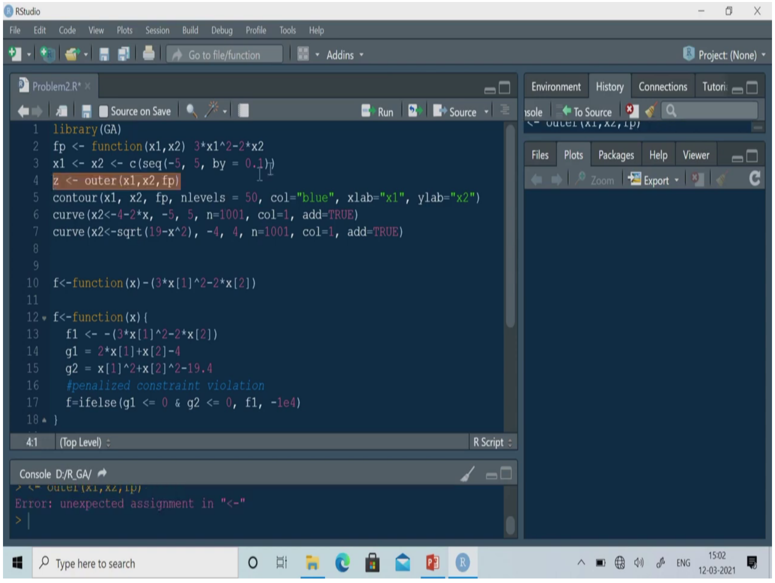
And then I am using executing this particular line and then I have to define what is x 1 and x 2. So, this is also between minus 5 and plus 5, the difference between two values is 0.1, so I am just, I have just defined this one.

(Refer Slide Time: 26:56)



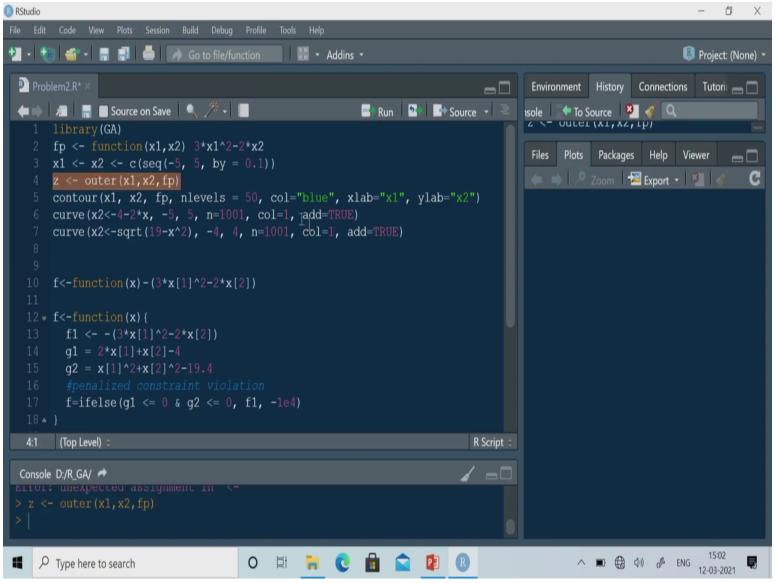
So, I am getting x 1 and x 2 and then just make it z, ok.

(Refer Slide Time: 27:01)



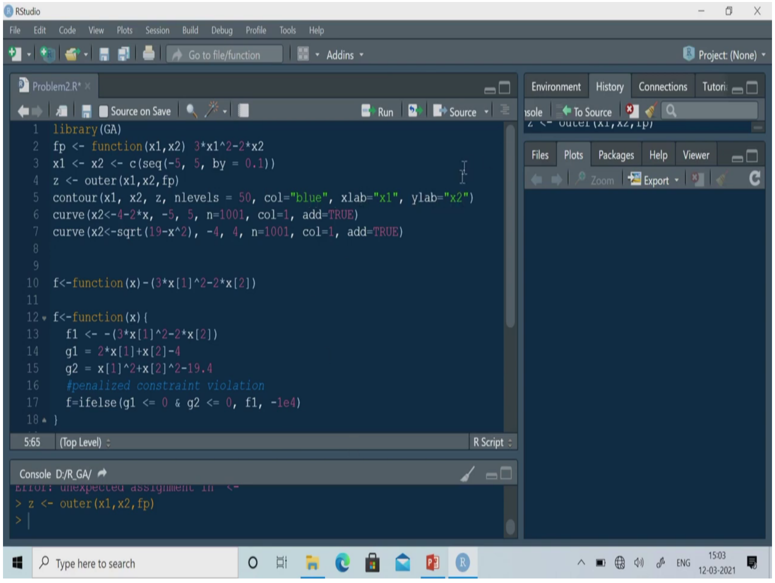
So, this is your z, z equal to outer, x 1 x 2 and fp.

(Refer Slide Time: 27:11)



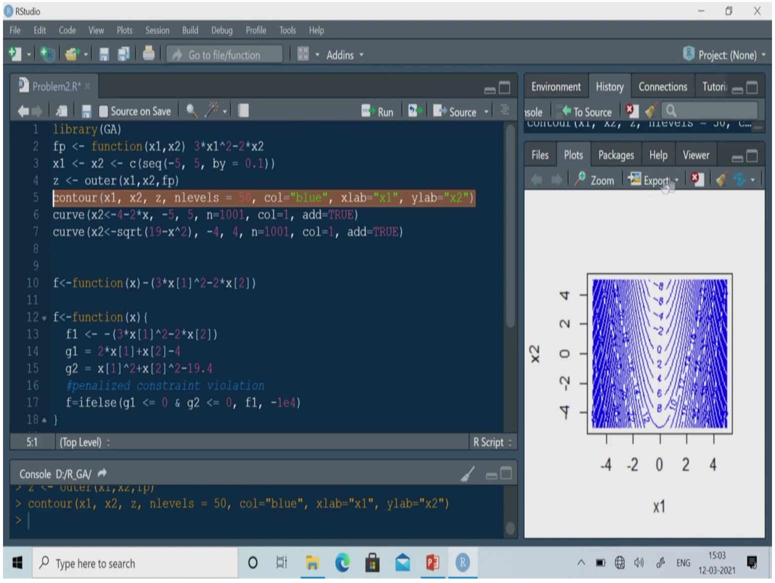
Let us run this particular line. So, I am getting z and then I am plotting it.

(Refer Slide Time: 27:17)



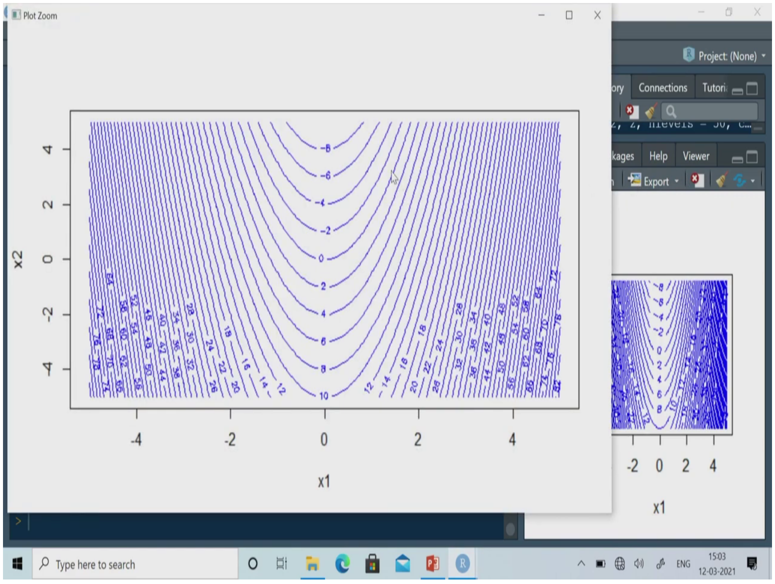
So, now, it will be your z, x, y, z, that is x 1, x 2 and z; number of interval I am putting 50, then this is blue colored and I am putting level x 1 and x 2.

(Refer Slide Time: 27:32)



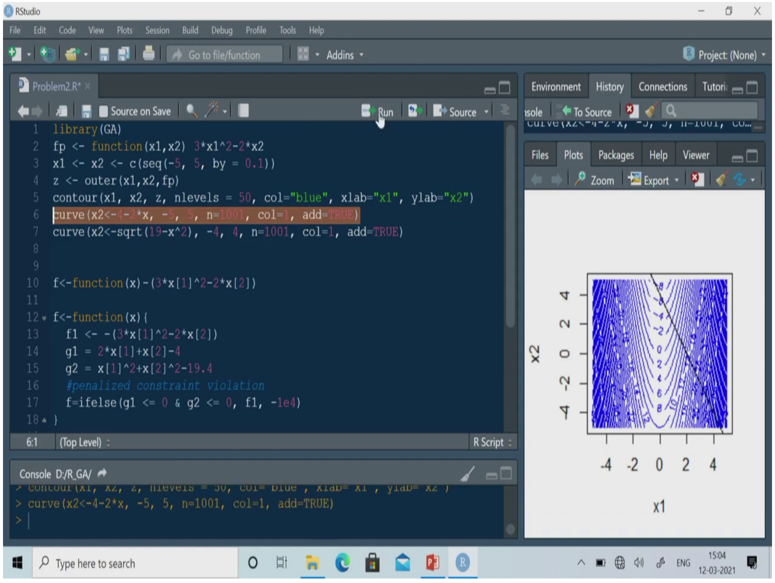
So, if I run this particular line, so I should get the contour.

(Refer Slide Time: 27:35)



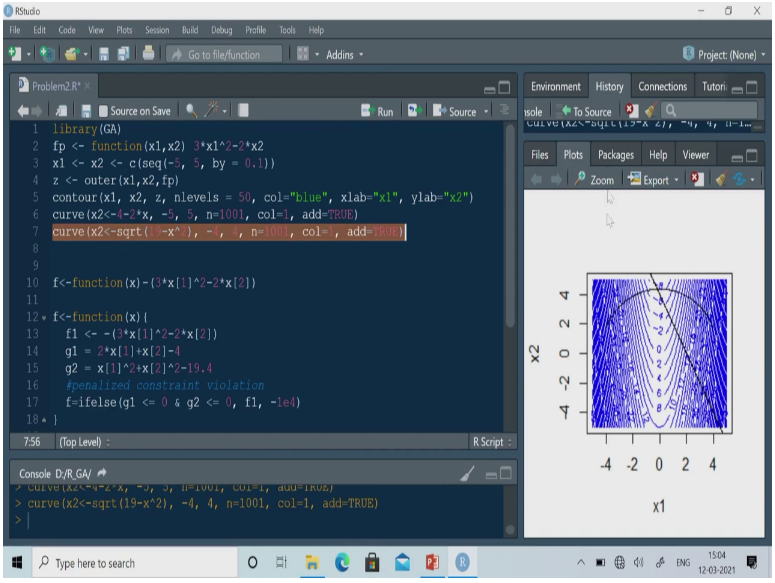
So, this is the contour I am getting and as you have seen, so unconstrained solution is somewhere here, unconstrained minimum point is somewhere here. Then let us plot the constraint. So, if you look at the constraints are twice x 1 plus x 2 less than 4. So, what I have done that, x 2 equal to 4 minus twice x 1 and then x 2 equal to 19 minus x 1 square and square root of that, square root of 19 minus 4 minus x 1 square. So, I am just writing in that way in order to plot that. So, I have written that x 2, that is 4 minus twice x 1 and it is between minus 5 and plus 5 and then I am putting color one and this is TRUE.

(Refer Slide Time: 28:26)



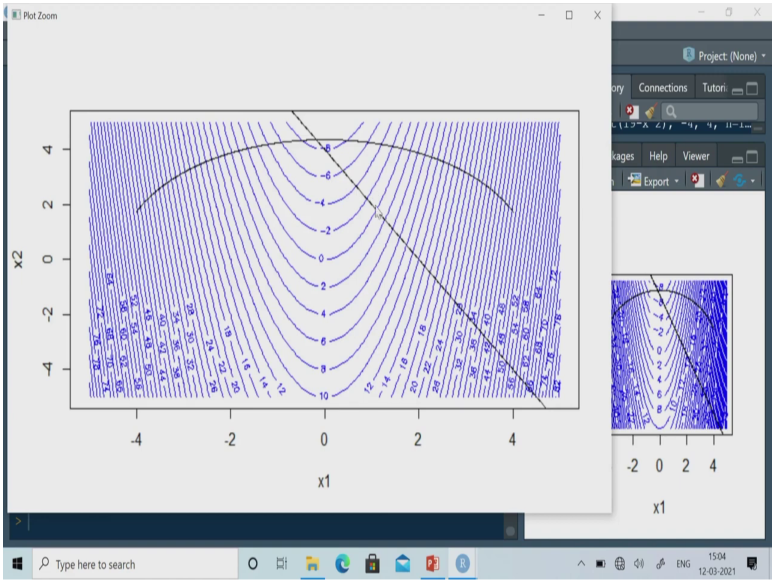
So, let us plot this particular line, I am getting that one and similarly I am putting the constraint, second constraint and the second constraint is square root of 19 minus x 2 square and this is between, I am putting between minus 4 and plus 4.

(Refer Slide Time: 28:45)



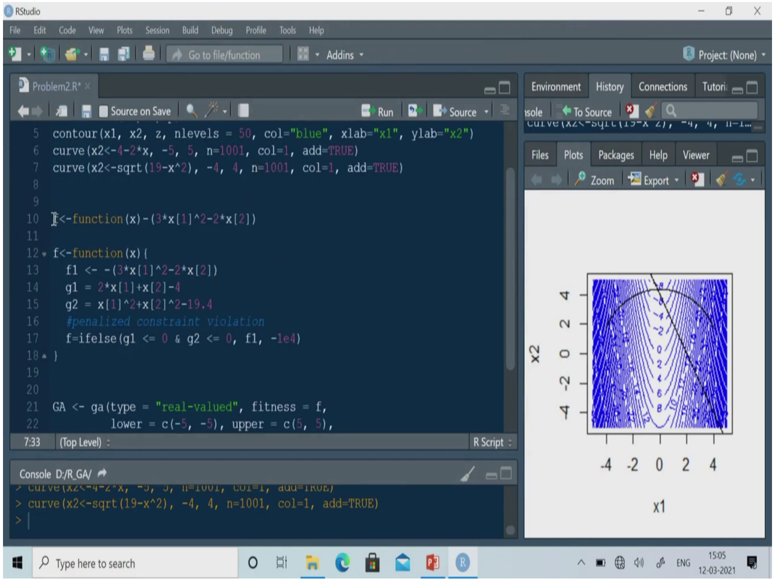
And if I execute this one, then I should get the constraint.

(Refer Slide Time: 28:49)



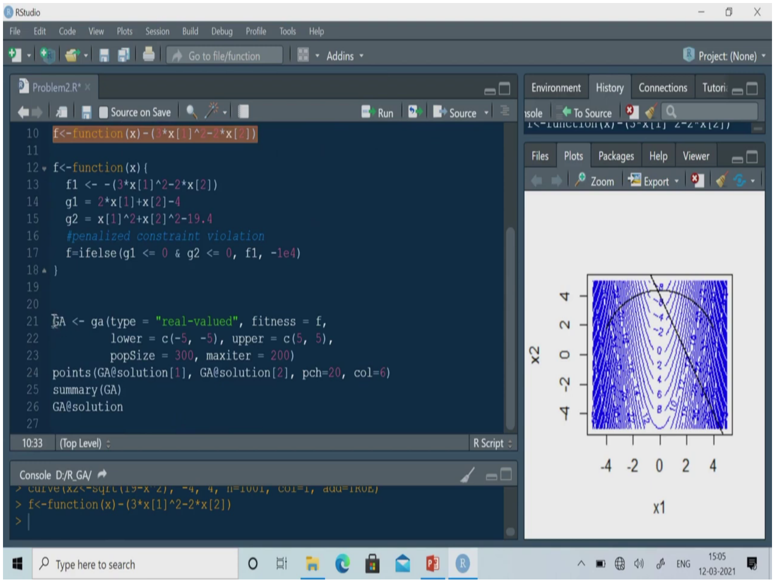
So, here. So, now, these are the constraint. So, this is your, this constraint is your first constraint that is g 1 and other one is g 2 and optimal solution is basically the point, where this constraint are cutting each other, ok. So, this is the optimal constraint optimization, constraint optimal solution. So, this is the constraint optimal solution. So, I have plotted this thing.

(Refer Slide Time: 29:16)



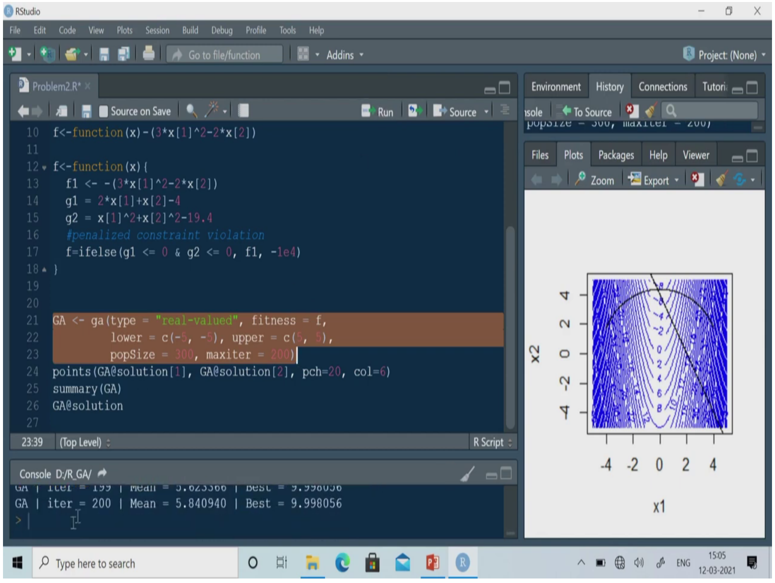
Now, let us find out the unconstrained solution. So, I am just writing the objective function, there is no constraint.

(Refer Slide Time: 29:24)



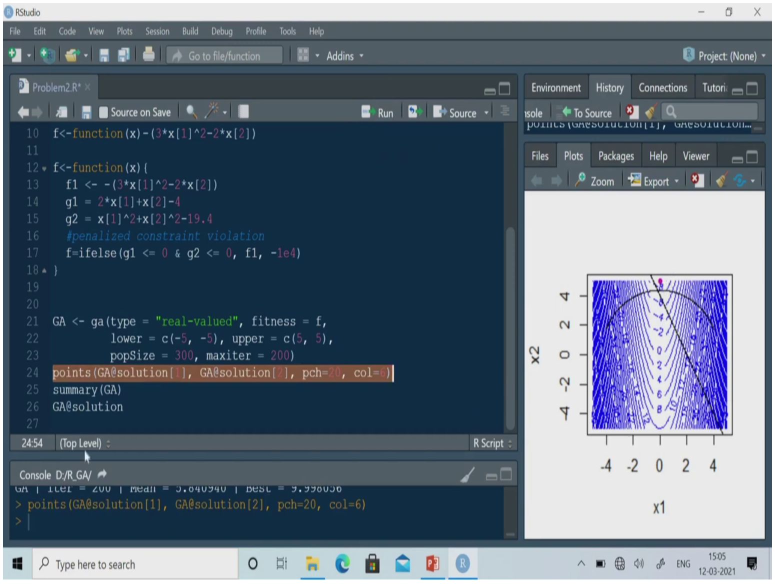
So, if I execute that one and then after that I am using GA. So, in this case also I have used population size 300 and maximum iteration 200 and then I have defined lower bound and upper bound and I use real valued GA and fitness function is f.

(Refer Slide Time: 29:42)



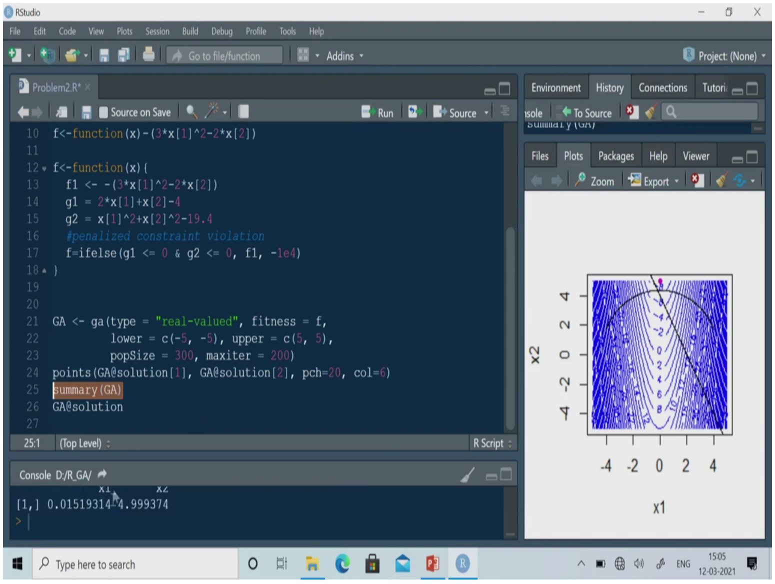
So, I have executed up to 200 iteration and let us see what solution I am getting. So, using this I can plot the solution over this contour map.

(Refer Slide Time: 29:56)



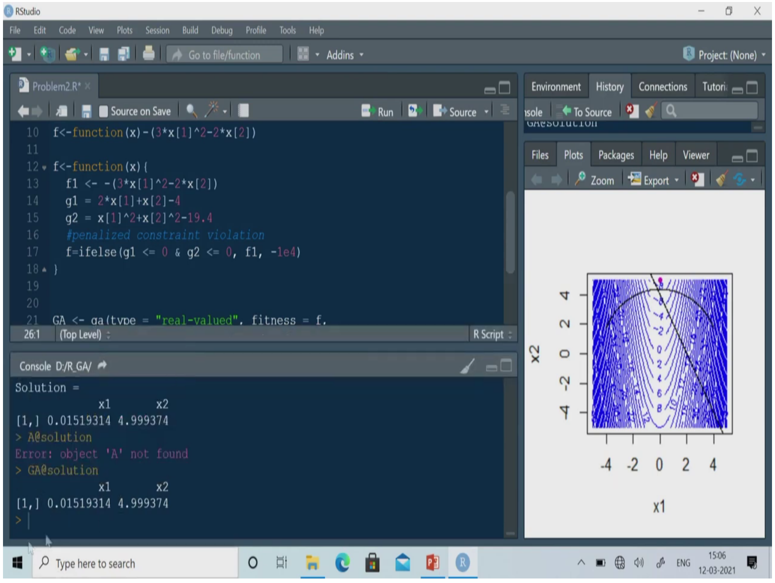
So, you can see that I am getting these particular solution.

(Refer Slide Time: 30:02)



So, I can also see what is the solution that summary I can see. So, this is the solution and yeah, so I can also see this.

(Refer Slide Time: 30:07)

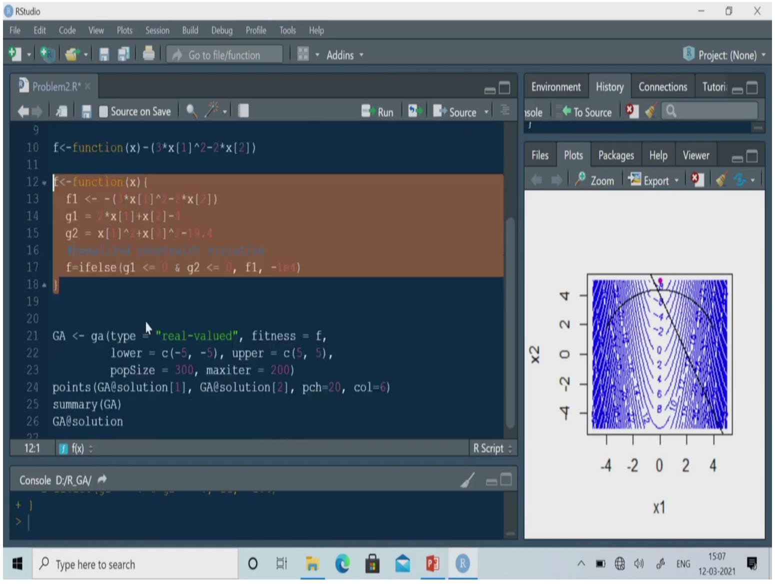


So, I can see the solution that is 0.1519314 and 4.99, actually it should be 5, ok. So, anyway, so this is the unconstrained solution. Now, let us see, let us see the constrained solution. So, what I have to do? I have to put g 1 and g 2 and like the other one; because the original problem is a minimization problem, so therefore, I have multiplied this objective function by minus 1.

So, I put minus 1 just to convert the problem to a maximization problem and then I have written g 1 and g 2. After that, so I have used if else function; that means if there is a violation that is g 1 is less than 0 or g 2 is less than 0, if then it will basically return f, there is no violation, otherwise it will return a negative value, large negative value. And therefore, that solution will not be considered by when it will go through the selection operator, ok.

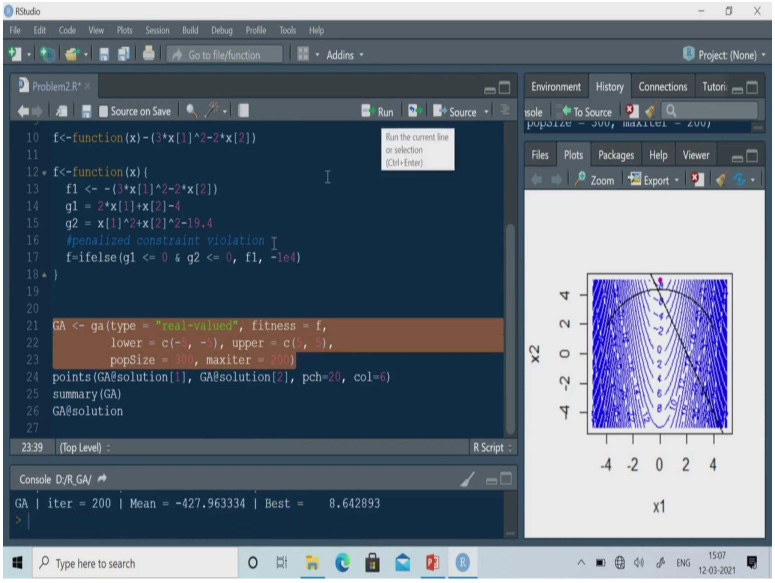
Therefore, I have penalized the solution using that. So, therefore, whatever f I am getting. So, if there is a violation, so it will written 1 e to the power 4, ok. So, it will written that or otherwise it will written the objective function value.

(Refer Slide Time: 31:45)



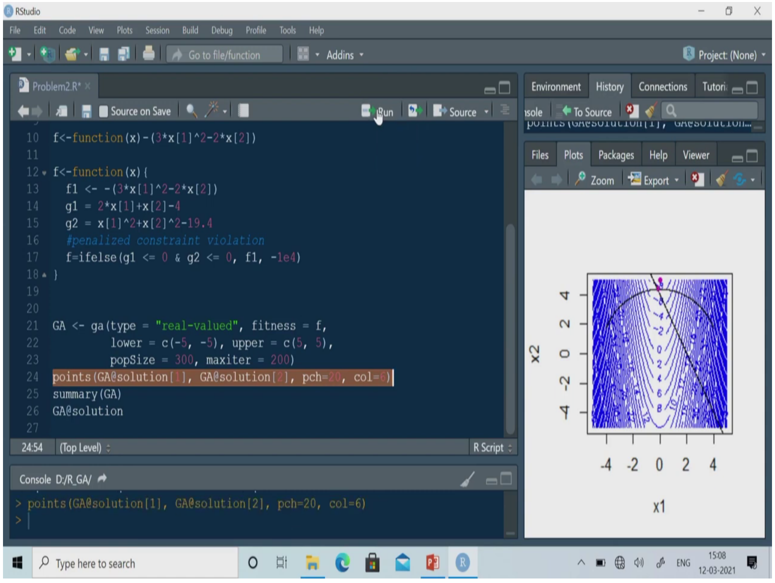
Now, let us run this particular function. So, it is running, then I am using GA now.

(Refer Slide Time: 31:51)



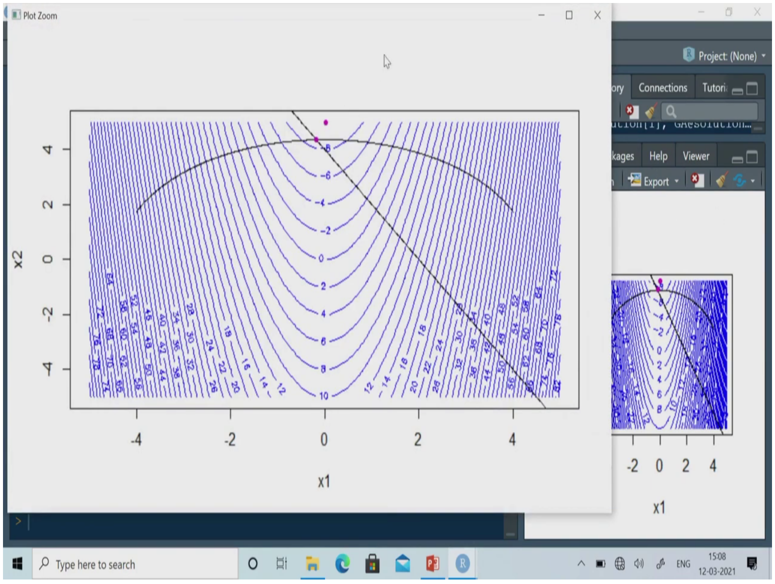
So, I am using GA now, now you see the solution, ok.

(Refer Slide Time: 32:00)



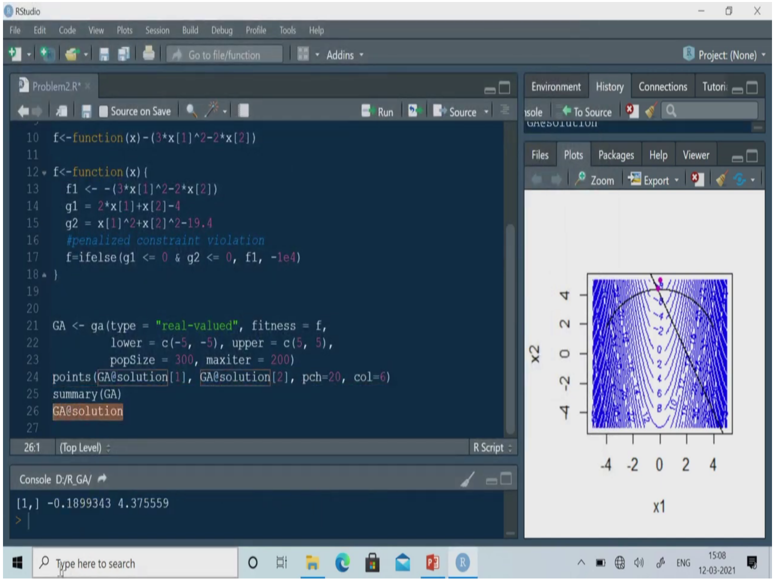
By using this particular line, so I can see the solution. So, the solution is somewhere here, you can check that one.

(Refer Slide Time: 32:04)



So, this is the solution, constrained solution and we got the constrained solution. So, you can see actually what is this solution.

(Refer Slide Time: 32:15)



So, from here I can check that one. So, this is the solution minus 0.1899343 and this is 4.375559, ok. So, you can see that one, the actual solution is 0.2 and minus 0.2 and 4.4; but we are not getting the exact solution, but we are getting a narrow optimal solution, that is minus 0.18 and this is 4.37, so it should be 4.4 and it should be 0.2.

However, as I said that I can apply the classical optimization method and I can get the exact optimal solution of this particular problem. So, I hope this is fine. So, we can handle a constraint. So, whether it is a linear or whether it is a non-linear constraint. So, we can just use, because if there is a violation, so I am putting a very low value; so that the when it is going through the selection operator, this solution will not be selected for the next generation, ok.

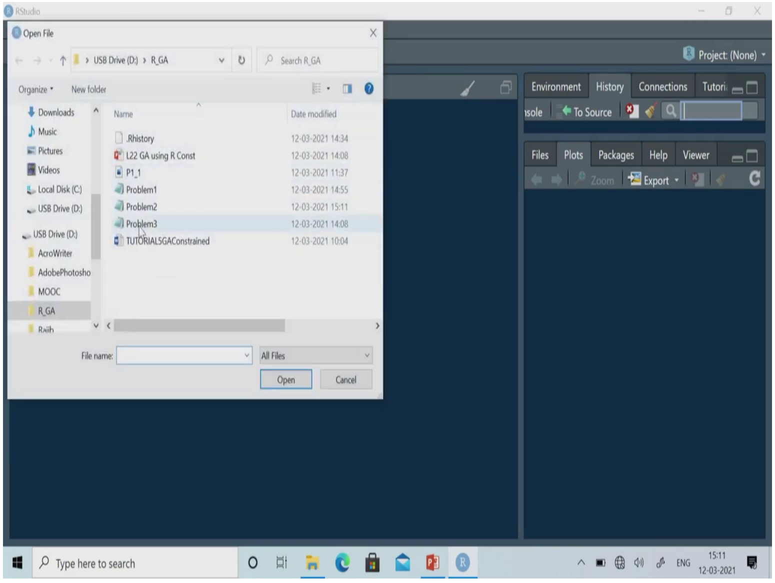
So, that way unlike the classical method, so in classical method, so what you have done. So, we have used a penalty parameter in some cases suppose exterior penalty method or interior penalty method. So, in case of extra penalty method, so we are using a very low value of penalty parameter, very small value of penalty parameter and then we are increasing it.

And but, in case of interior penalty method; so we are using high value initially and then increasing. But here you can define a large number, large value, large negative value if you are doing the maximization problem. If you are solving a minimization problem, then you put a large positive value. So, in that case selection operator will not select them. So, I hope this is clear to you. So, then let us go to the next problem.

So, I think I can show you again. So, finally, I got the solution. So, this is the constrained optimal solution of this particular problem. Now, let us go to the next problem. So, next problem is again it is a very simple problem, only we have one constraint; the unconstrained solution is 3 and 3, so that is the unconstrained solution.

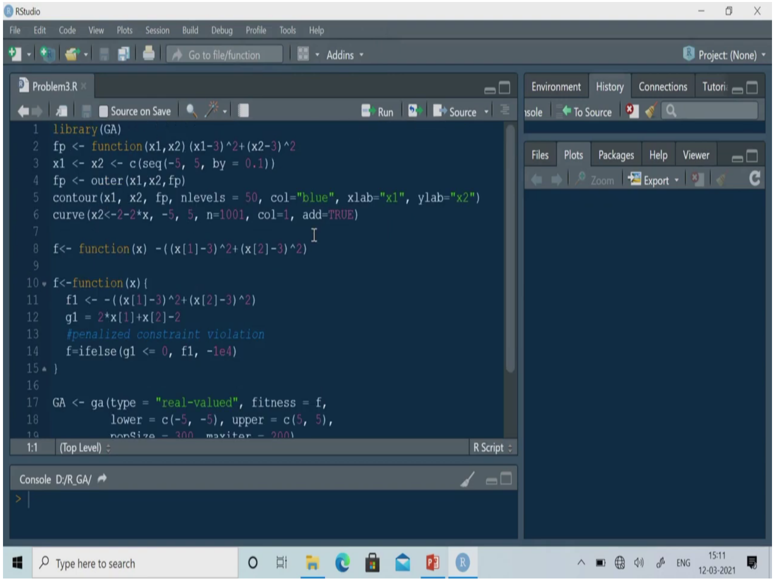
And once we are putting this linear constraint and then I should get a constrained solution of this particular problem; solution is 0.199 and 1.602, so that is the solution. And let us see I will show you the solution in R, ok. So, now I yeah, I will close this one, I will clean everything, ok.

(Refer Slide Time: 35:03)



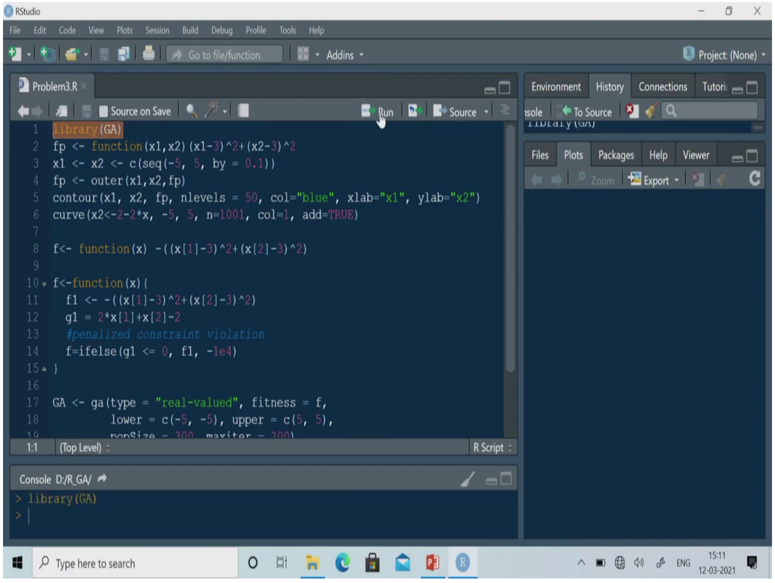
So, now I am opening, I will obtain the third problem.

(Refer Slide Time: 35:08)



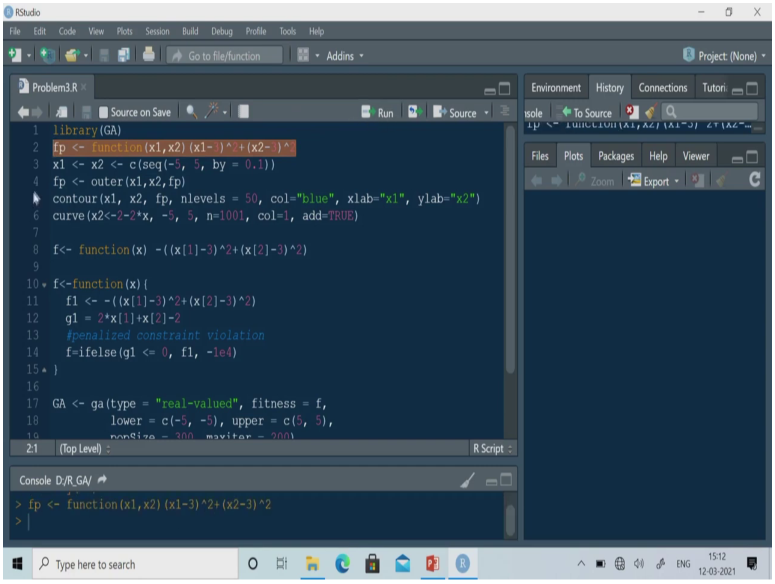
So, this is the third problem, ok. So, the third problem is again, I am just; so again these lines are just to plot the function. So, here I am defining the objective function.

(Refer Slide Time: 35:25)



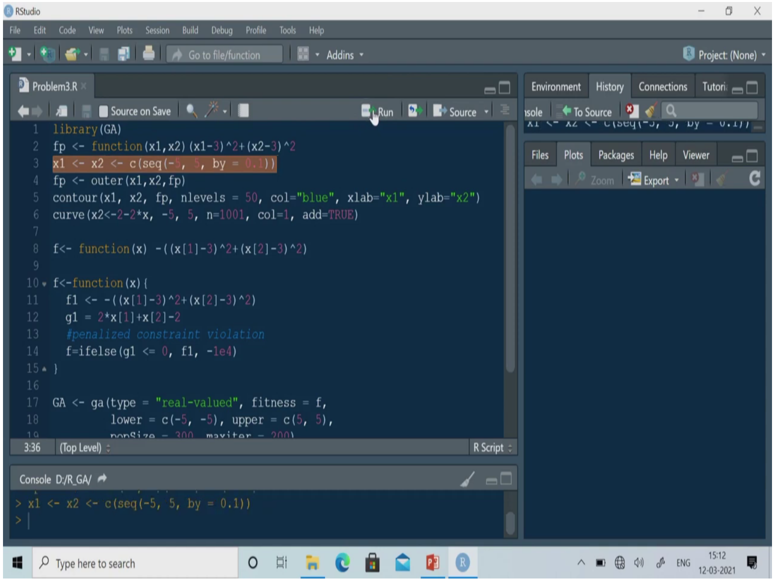
So, let us run the library, then execute the objective function, ok. So, that is x 1 minus 3 whole square plus x 2 minus 3 whole square.

(Refer Slide Time: 35:35)



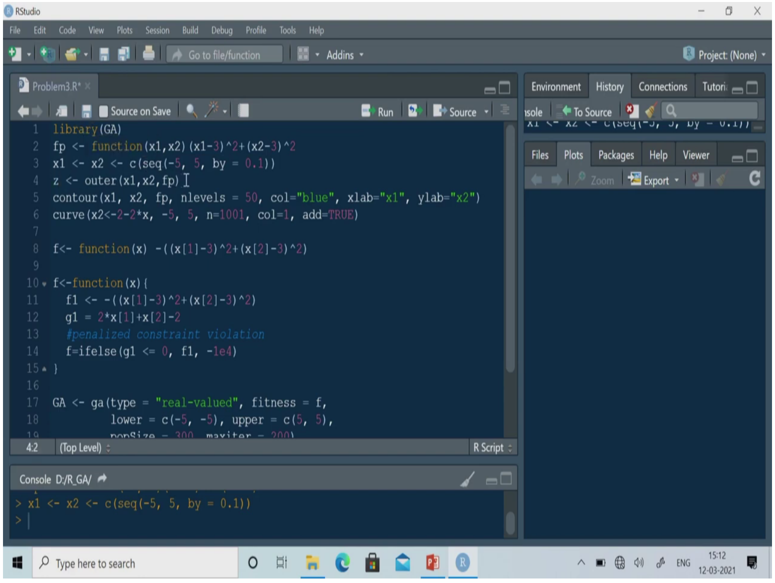
So, this is the objective function.

(Refer Slide Time: 35:41)



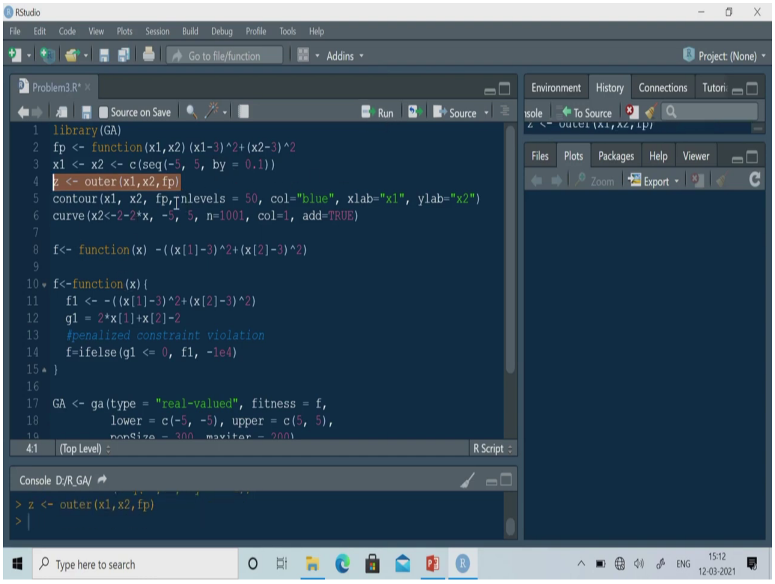
And then I am just generating values for x 1 and x 2.

(Refer Slide Time: 35:45) .



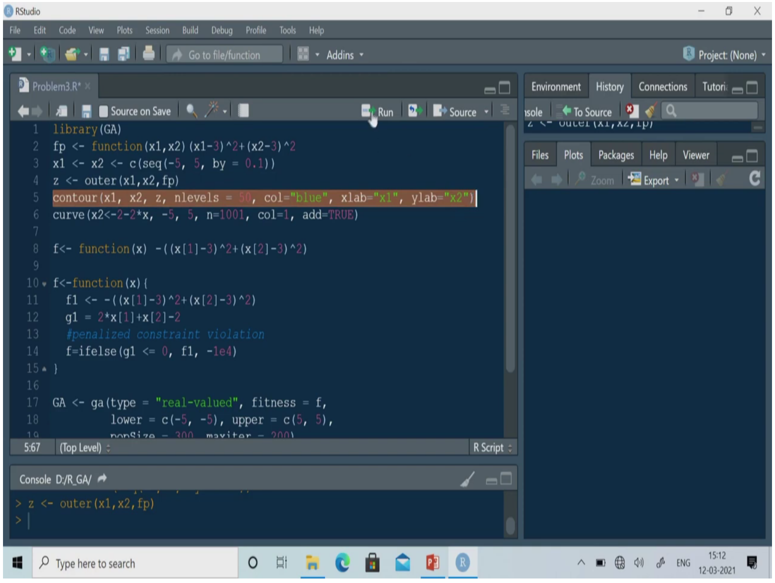
And then let us write it as a z again, ok.

(Refer Slide Time: 35:50)



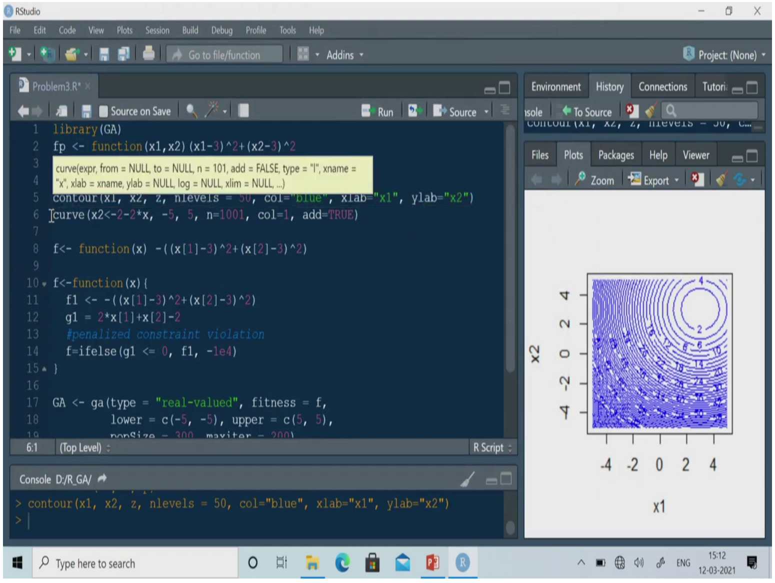
So, I am using the outer function to calculate the values at grid point, at different grid point and this is the z I am putting, ok.

(Refer Slide Time: 35:56)

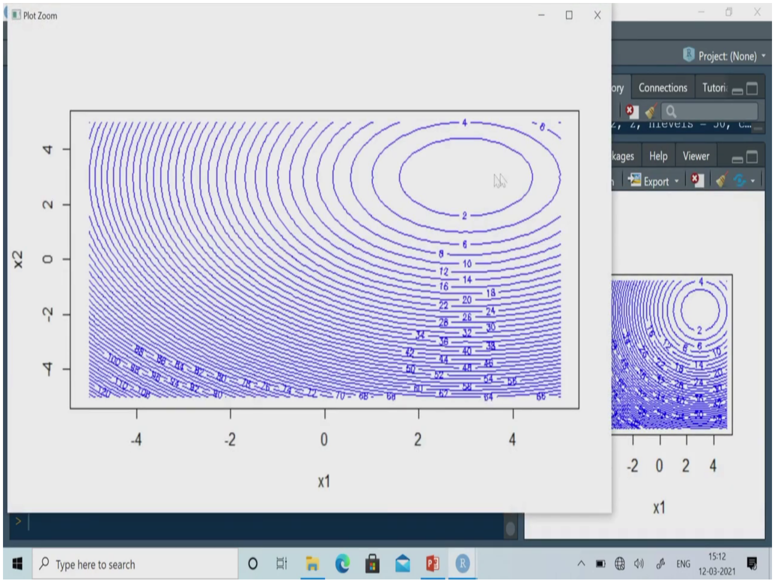


And this will give you the contour map, ok.

(Refer Slide Time: 36:01)

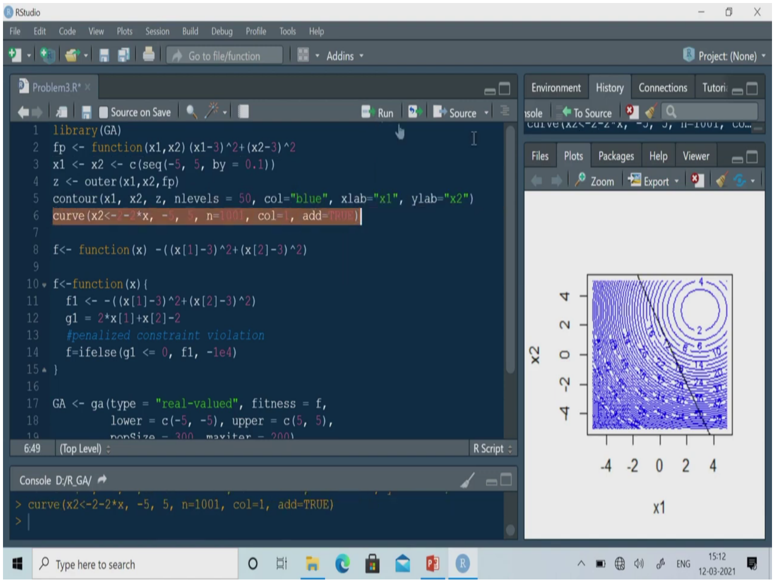


(Refer Slide Time: 36:04)



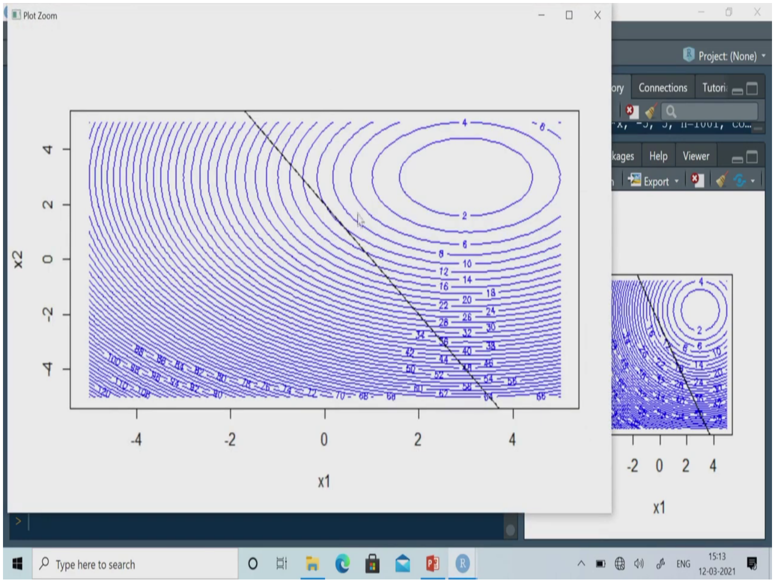
So, I am getting this contour map again. So, the unconstrained solution is somewhere here, that is 3 and 3. Now, I can put the curve, now I can put the constrained line, so I have use the curve function again. So, in this case x 2 equal to 2 minus twice x and it is between minus 5 and plus 5.

(Refer Slide Time: 36:29)



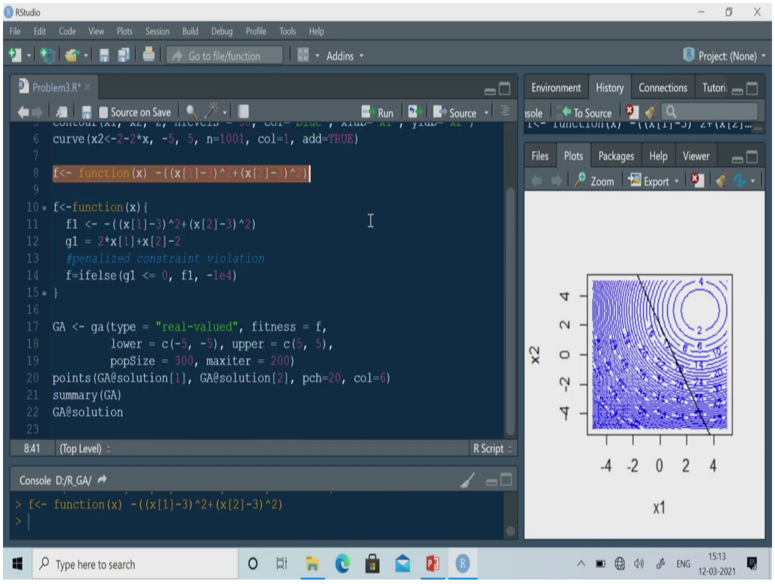
So, let us run this particular line.

(Refer Slide Time: 36:32)



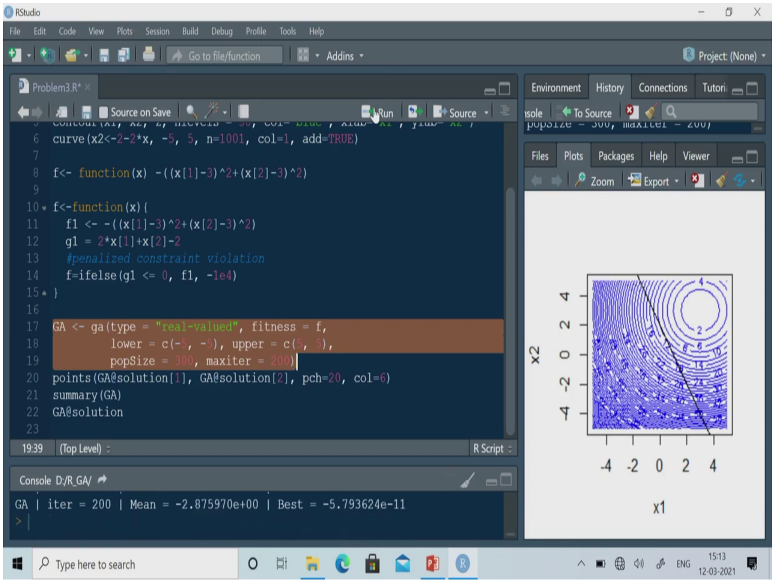
And I will get the constraint and your constraint optimal solution is somewhere here. So, this is the constraint optimal solution and let us see whether I am getting there one or not. So, before that let us find out the unconstrained solution. So, I am writing the objective function and I am putting negative, because the original problem is a minimization problem and I have converted this problem to a maximization problem by multiplying it minus 1.

(Refer Slide Time: 37:01)



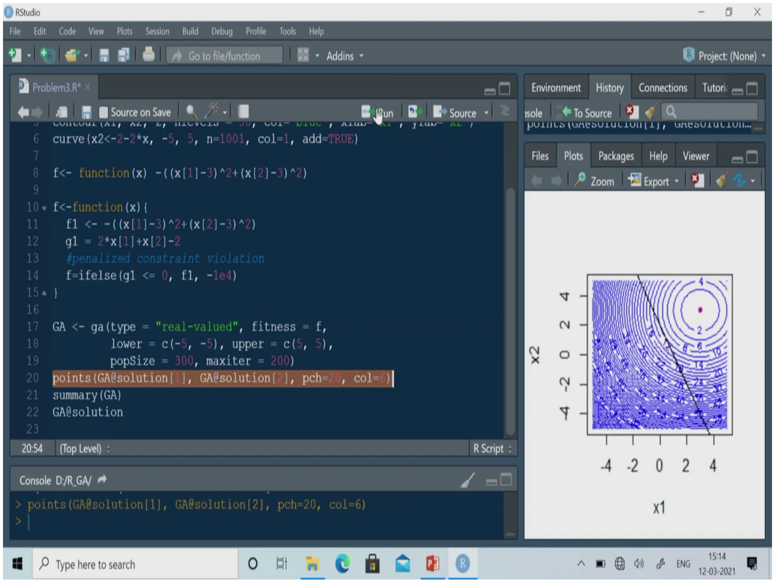
So, let us run this thing and then you execute the GA line, ok. So, now, I am not defining; again here population size is 300 and iteration is 200. And I have defined upper bound and lower bound; I did not define crossover probability and mutation probability, the default value I have taken here.

(Refer Slide Time: 37:21)

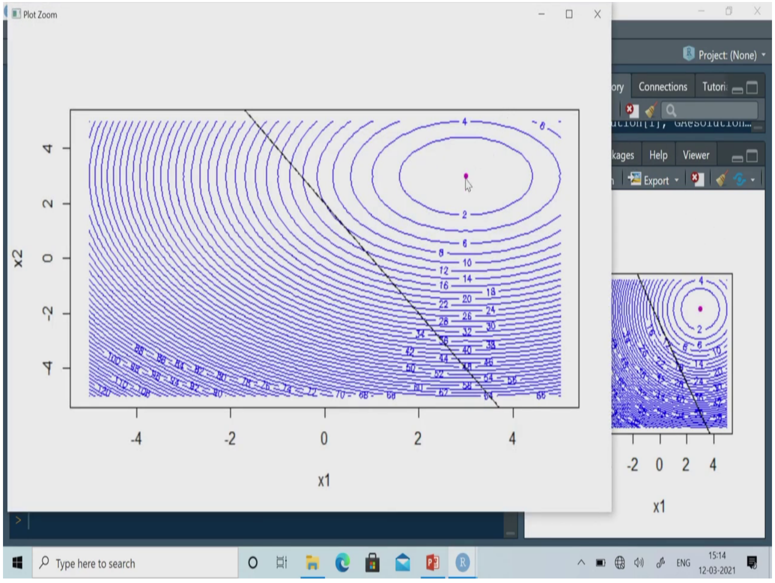


And if you are executing this one so, let us see what solution I am getting; I would like to plot this particular point over the contour map.

(Refer Slide Time: 37:33)

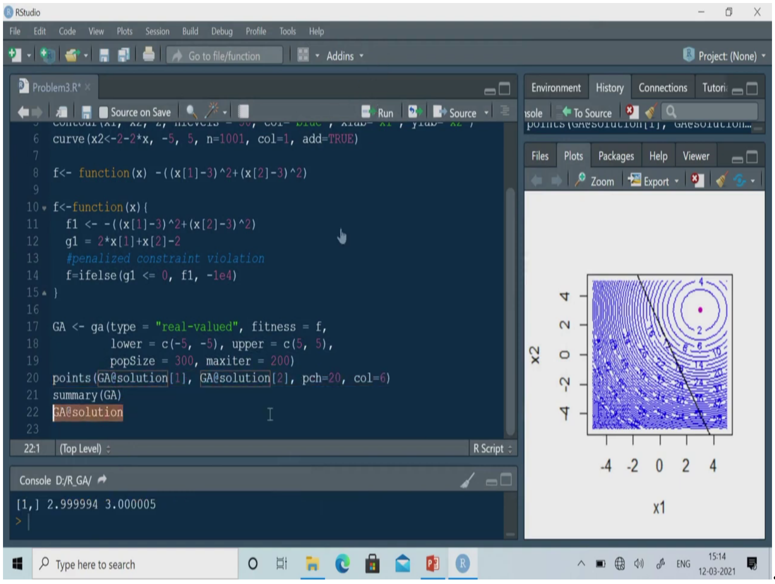


(Refer Slide Time: 37:36)



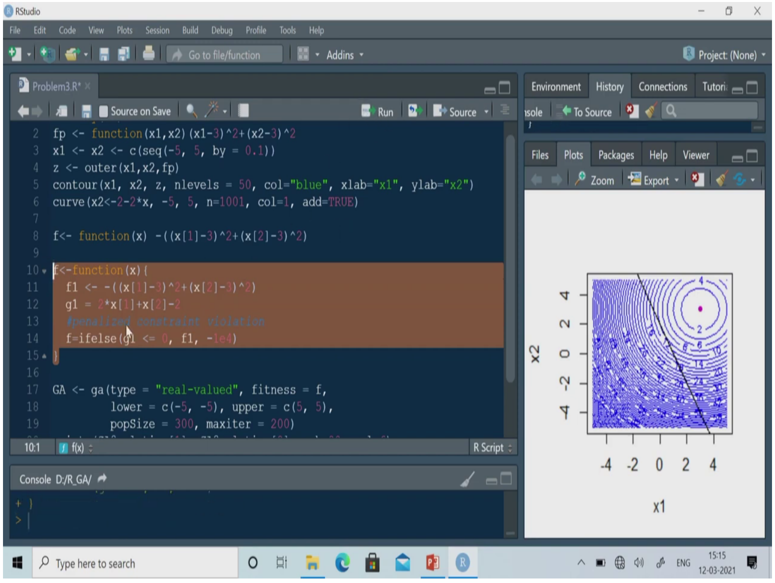
So, this is the, this is the point I am getting. So, this is the unconstrained solution and you can see what is the solution.

(Refer Slide Time: 37:45)



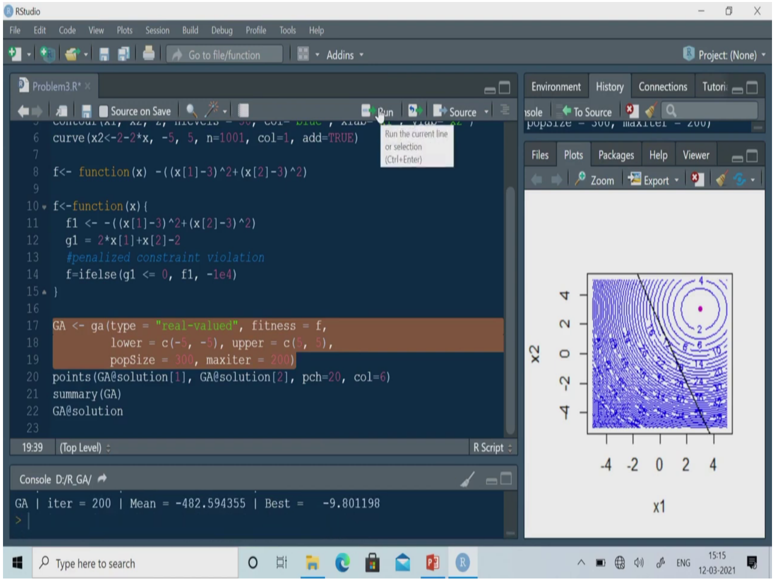
So, by running the GA solution, so I am getting 2.99 and 3. So, exact solution is 3 and 3; but I am getting 2.99, it is very close to the unconstrained optimal solution. Now, let us calculate the constrained solution and here I am defining the objective function and we have only 1 constraint, the constraint is twice x 1 plus x 2 minus 2 is less than equal to 0. And I am again I am using if else function, if g 1 less than 0, then it will return f 1 or otherwise it will return a small value, ok.

(Refer Slide Time: 38:27)



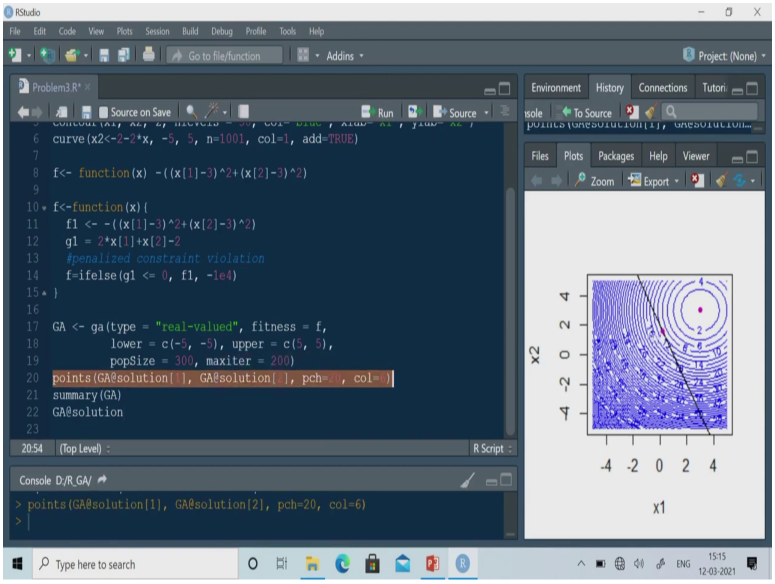
So, let us run this particular line then, ok.

(Refer Slide Time: 38:33)



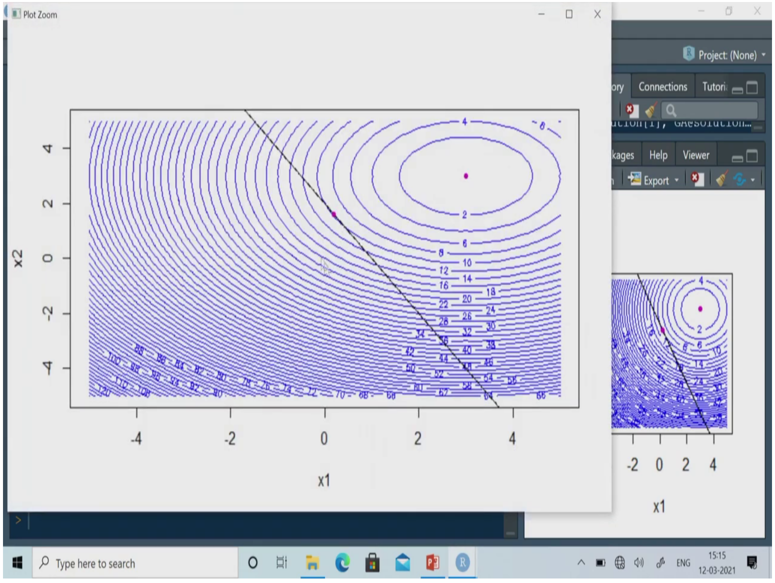
So, now, I would like to execute the GA.

(Refer Slide Time: 38:43)



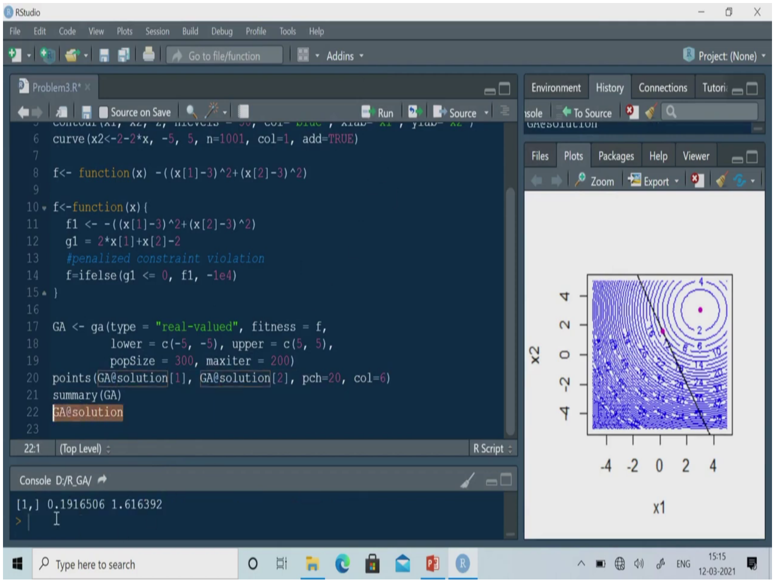
So, let us see what solution I am getting. So, in this case I am getting this solution.

(Refer Slide Time: 38:46)



So, you can see that one that, this is the constrained solution of this particular problem, so I am getting this solution.

(Refer Slide Time: 38:56)

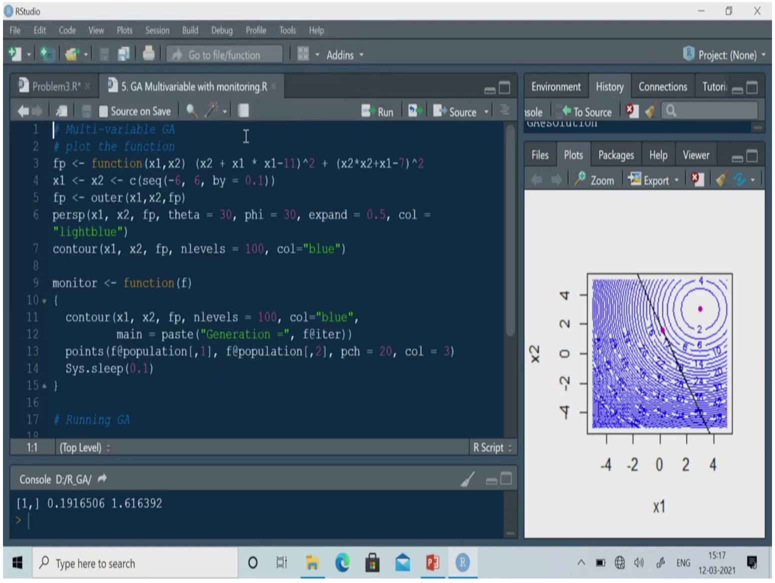


And the solution is 0.191 and 1.61 you can check; actually the solution is 0.199 and 1.6, so I am also getting close to that one. And as I said that you may not get the exact solution; but you can apply classical method to get the exact solution, but I am getting a near optimal solution here.

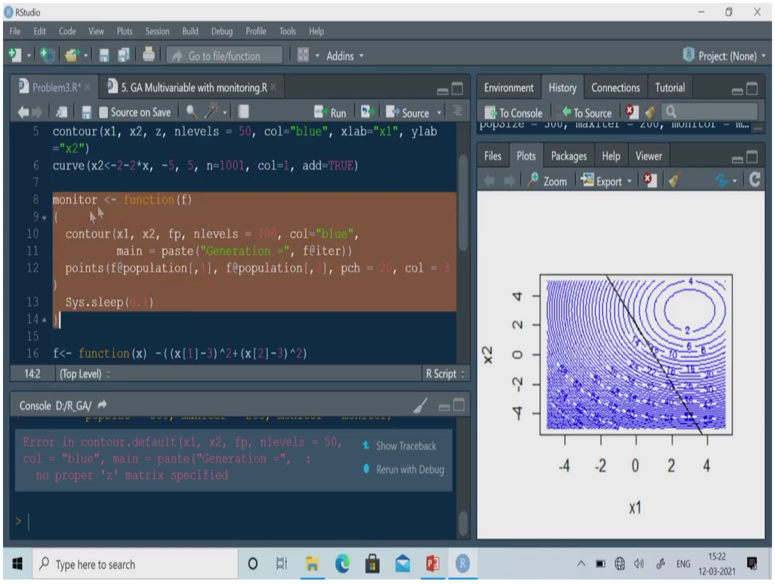
So, I am also getting the constrained optimal solution of this particular problem. So, I got the constrained optimal solution of this problem. So, similarly I can also solve a problem with more than two constraint or two variable; that I am not showing here, because I will not be able to plot that one, but I can also solve a problem with higher dimension.

If it is 10 variable, 100 variable I can try and with suppose 100 constraint that also it is possible to solve. So, I did not use the monitor function, but I can also use the monitor function to see the progress. So, let me try that one. So, if I can use the monitor function, let me try to use the monitor function.

(Refer Slide Time: 40:20)

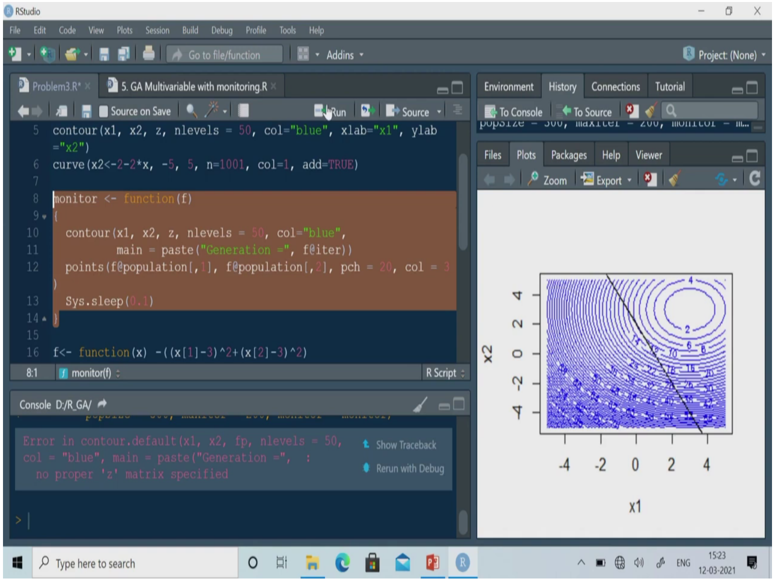


(Refer Slide Time: 40:22)



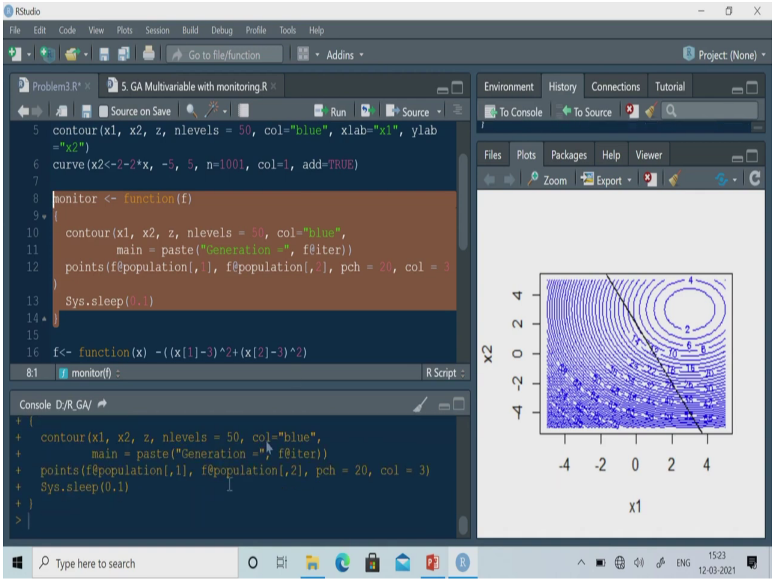
So, now, I have used this monitor function here. So, I have just copied from whatever I have done in the last class. So, here, so I have used contour and this is x 1, x 2 and this will be your z now.

(Refer Slide Time: 40:38)

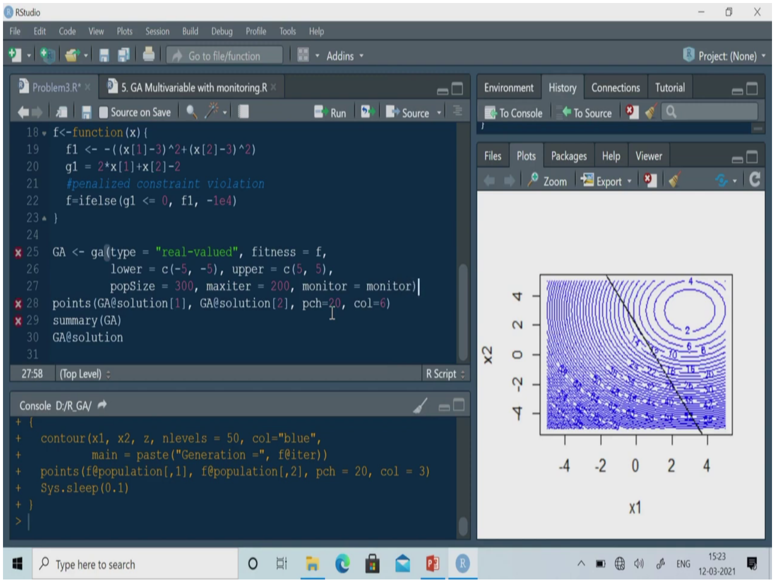


This will be z and number of levels you can make it 50 yeah, I think rest are fine. So, I will not change the anything and then let us execute this particular monitor function, ok.

(Refer Slide Time: 40:53)

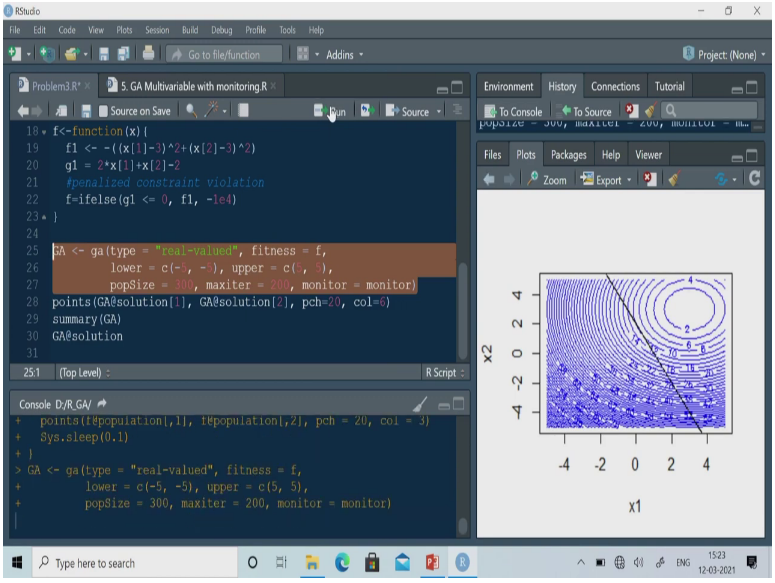


(Refer Slide Time: 40:56)

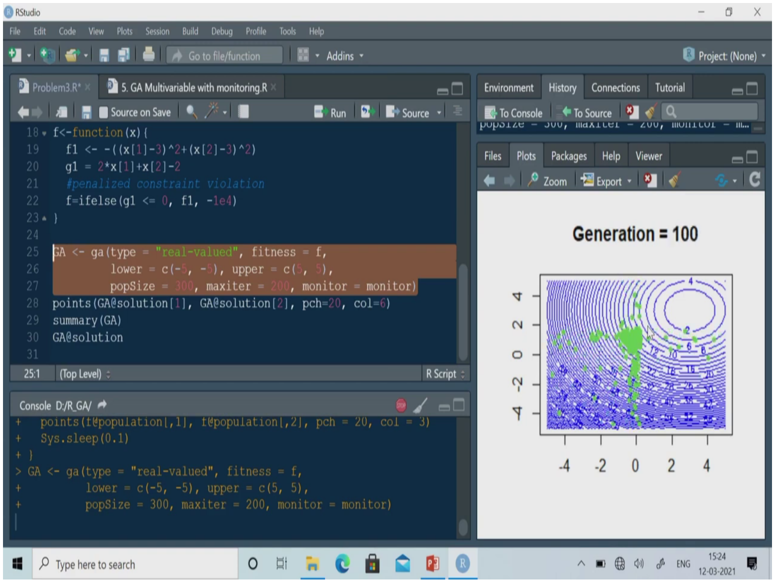


And now, so only change I have to do here; if I run GA, so I have to use monitor equal to monitor, ok.

(Refer Slide Time: 41:23)



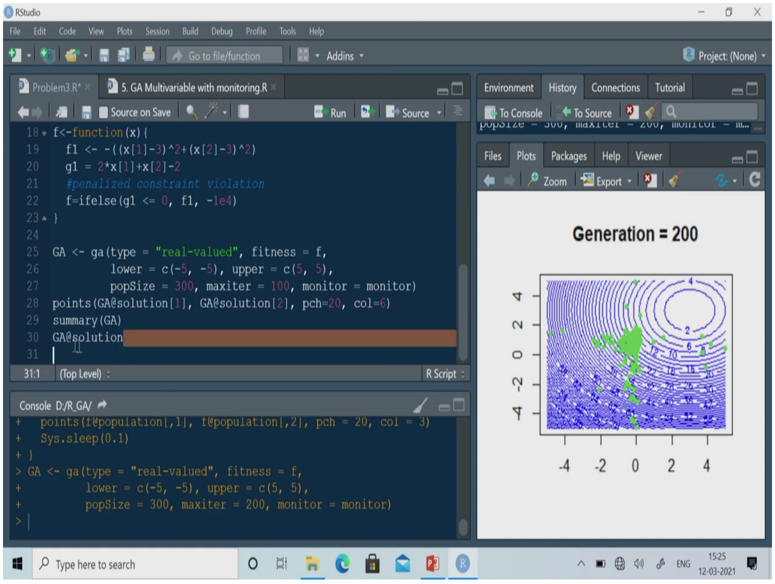
(Refer Slide Time: 41:24)



So, let us execute this particular line. So, I am getting that one. So, now, you just see, this is converging somewhere at constrained optimal solution; not the original unconstrained problem, the constrained solution is somewhere here and all the populations are actually migrating towards the constrained solution of this particular problem.

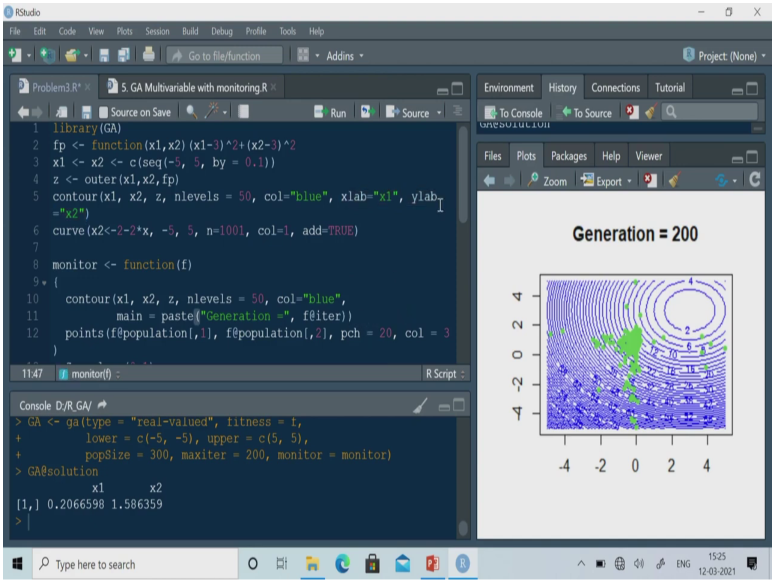
So, here only thing is that, I did not put the constrained line. So, I will also put that one, but I am getting the constrained solution.

(Refer Slide Time: 42:12)



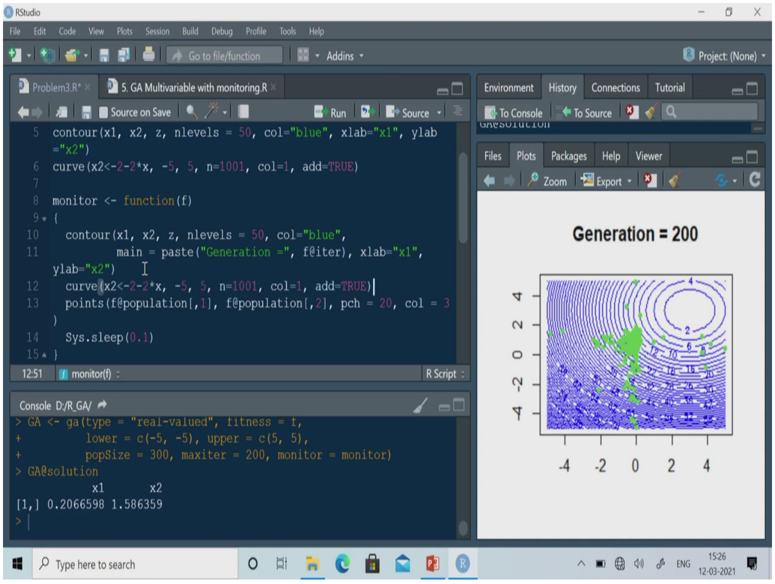
So, now, it will go up to 200 iteration. So, earlier iteration was 100, maybe I will change it to 100; otherwise it will take some time to get the solution. So, this is the 200 run, ok. So, I will make some any anyway.

(Refer Slide Time: 42:40)



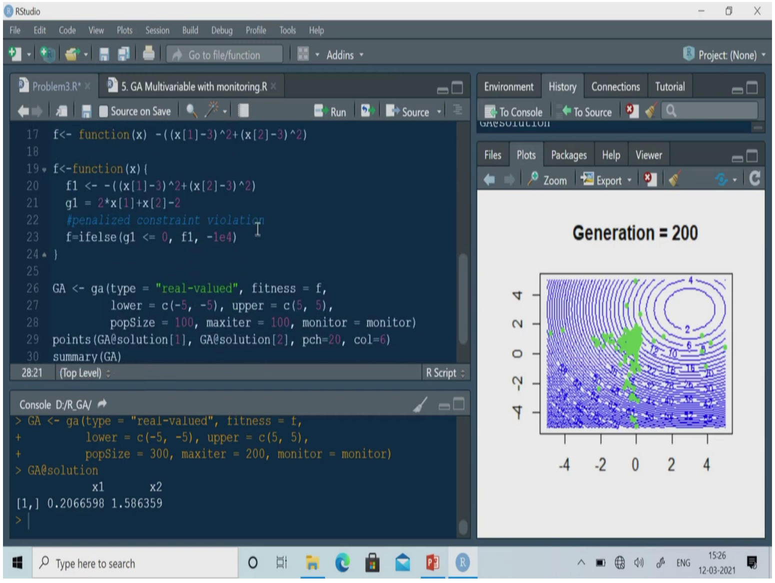
So, you can see the solution; now I think you are getting the solution that is 0.2 and 1.58, but let me change something here in the monitor function. So, I would like to put this x level and y level that I will put, x level and y level I will put.

(Refer Slide Time: 43:03)

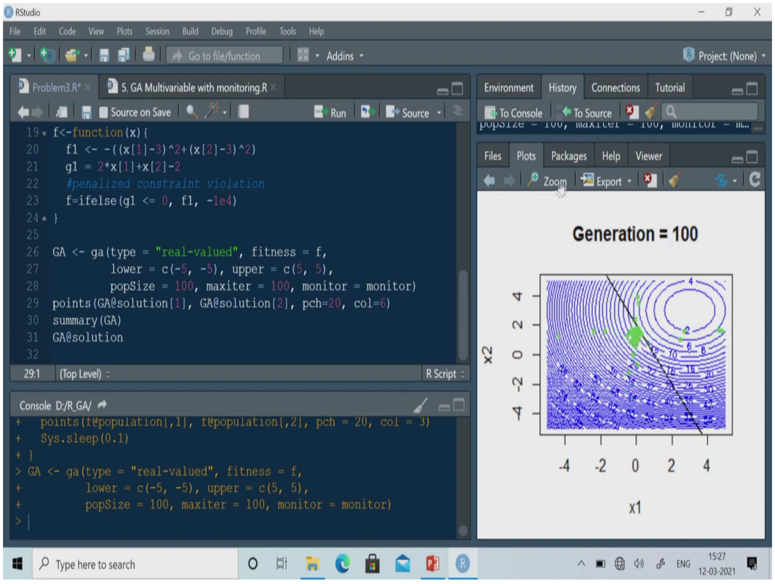


And I would like to put this curve here; this is the curve I would like to put and then let us see. So, now I am running it for 100 generation, maybe population size of 100, ok.

(Refer Slide Time: 43:30)

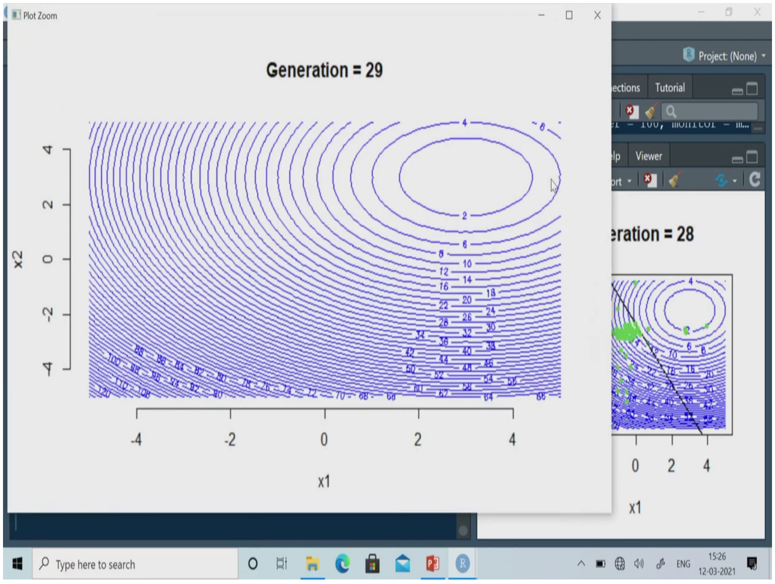


(Refer Slide Time: 43:36)



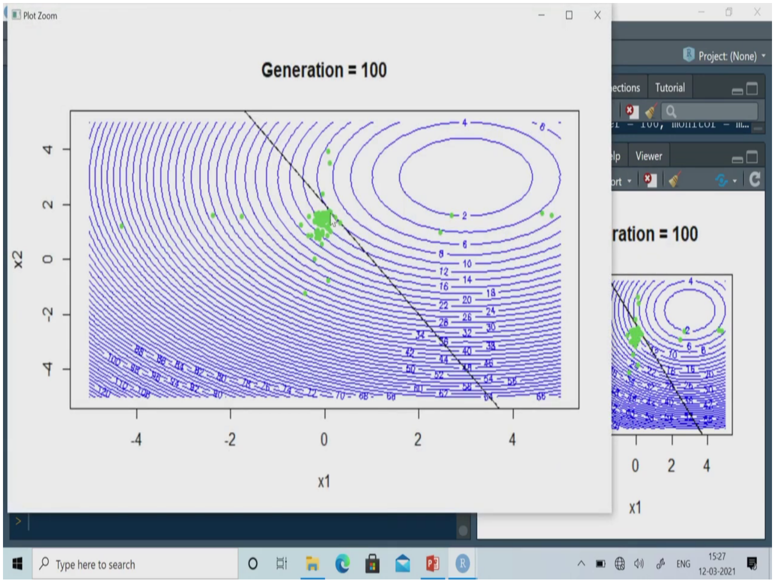
So, first execute this monitor function and then I am executing this GA. Now, you just see. So, I have this constrained line and the constrained solution is somewhere here.

(Refer Slide Time: 43:54)



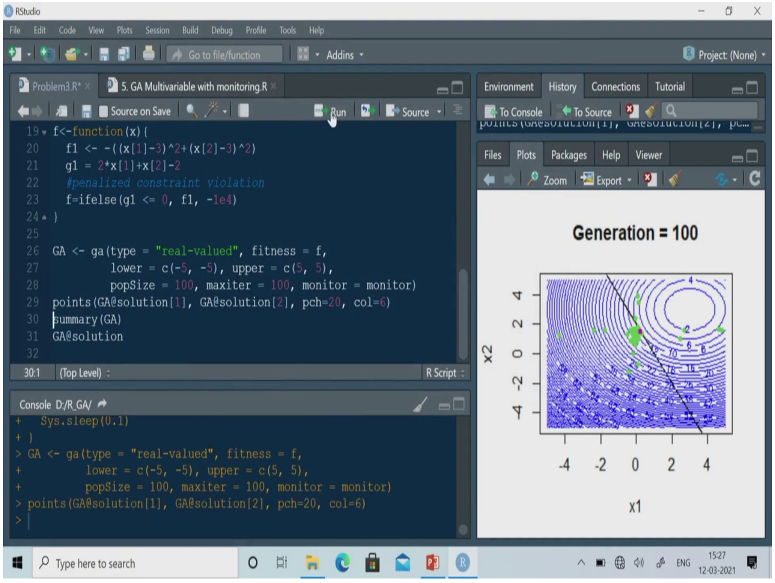
And the constrained solution is somewhere here and you can see with the generation, the population is actually converging at the constrained optimal solution, ok.

(Refer Slide Time: 44:15)



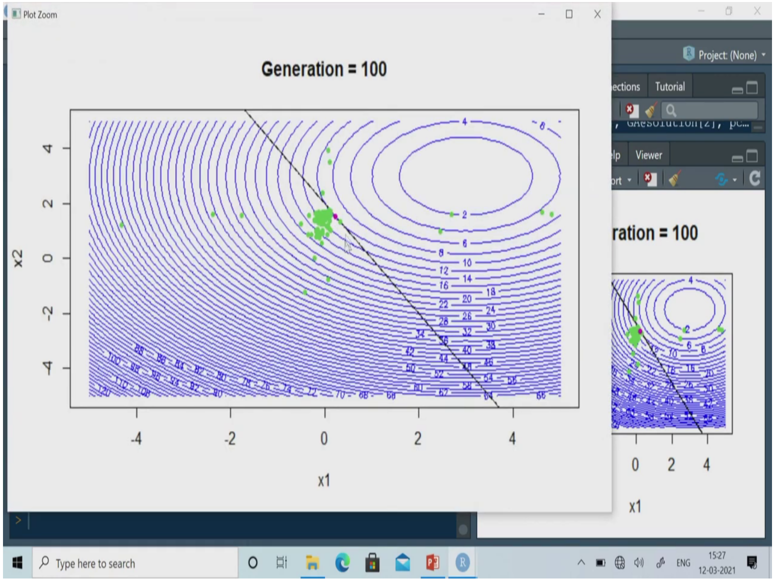
So, we got the solution. So, final solution is somewhere something like that. So, this is the constrained optimal solution and I can plot the solution using this point function.

(Refer Slide Time: 44:23)



And somewhere this is the solution you can see basically.

(Refer Slide Time: 44:26)



So, this is the solution. So, that way I can also put the monitor function and I can see that how these populations are migrating from one place to another or towards the constrained optimal solution that I can actually observe using monitor function.

So, I hope you will be able to solve a constrained problem using genetic algorithm or and in our platform. So, let us try yourself and if you have any concern, we can discuss in the next class.

Thank you.