Foundations of R Software

Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 53
Some Examples of R Programming

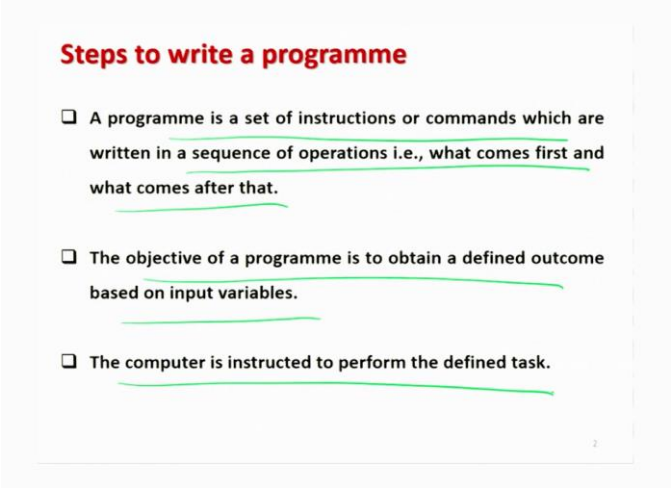
Hello friends. Welcome to the course Foundations of R Software and welcome to the last lecture of the course. So, now, you can see we have learnt many aspects of the R software, basic commands, basic function, how to create graphics, how to write functions etcetera. Now, in this lecture I am not going to entertain any new topic, but my objective is that I just want to show you some salient features of the R software through the programmes.

And my objective is this, I would like to explain you how you can write a programme in the R software and for that I will try to take couple of examples and through those example I will try to show you the different features of this R programming ah.

Well, I have here one question that some of the candidates may not be knowing anything about the programming, and some candidates may be knowing very well about the programming. So, what I am going to do, that in the first couple of minutes I am going to give you a very brief introduction to the programming that what do we expect and what and why do we do this programming. And after that I will try to take couple of example to explain you the different types of applications and features of the R programming.

So, let us begin about lecture and try to understand first, the very basic fundamentals of this programming and to know why do we need programming. So, let us begin our lecture.

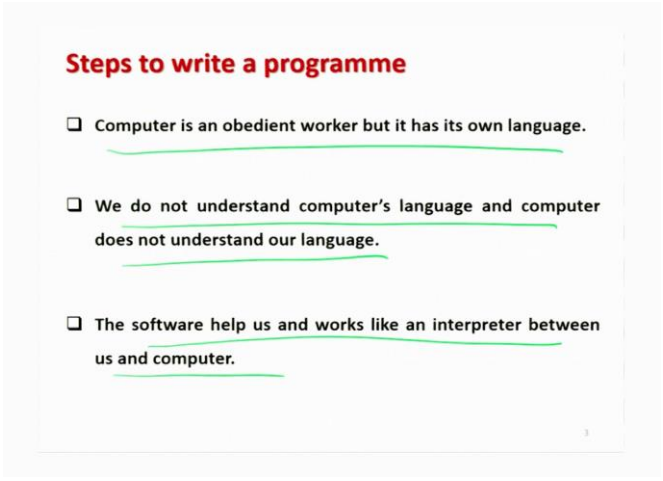
(Refer Slide Time: 01:46)



So, you see first question comes here, what is the programme and what are the different steps to write a programme? So, a programme is simply a set of instructions or commands which are written in a sequence of operation, that whatever you want to do first they are written first and then whatever you want to do in the second step they are written at the second step, right. So, whatsoever come first and whatsoever come after that all the instructions are given in a sequence, right.

And the objective of a programme is to obtained and outcome which is predefined and which is based on some input variables and some given mathematical function. And the only thing is this that we can do such calculations and manipulations manually also, but we try to take the help of computer and the computer is instructed to perform this defined task.

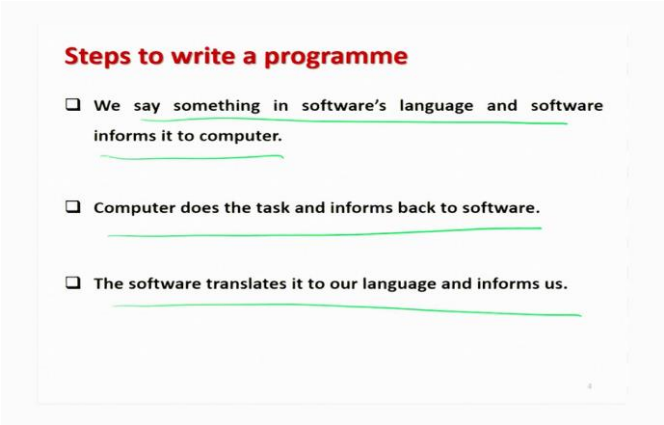
(Refer Slide Time: 02:33)



The only reason is that because computer is an obedient worker and it can get our things done fast. And the only issue is this the computer does not understand our language like as Hindi or English, but computer has its own language.

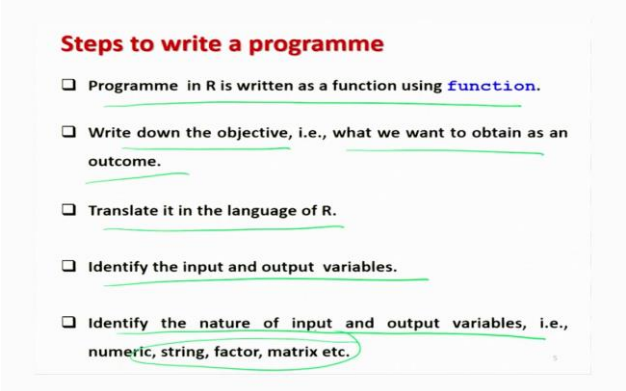
And our problem is this we do not understand the computers language and computer does not understand our language too. So, now, what is the solution? The solution come through the software. The software help us and works like an interpreter between us and the computer. Whatever we try to give or whatever we type in programme in English language, they are translated to the language of the software, so that the computer can understand it, computer does the work.

(Refer Slide Time: 03:08)



And then it is reverted back in the same language what we know. And what we try to do? We try to say something in the software language and software informs to the computer. Computer does the task and informs back to the software and then software re-translate the outcome into our language and inform us through the outcome, right.

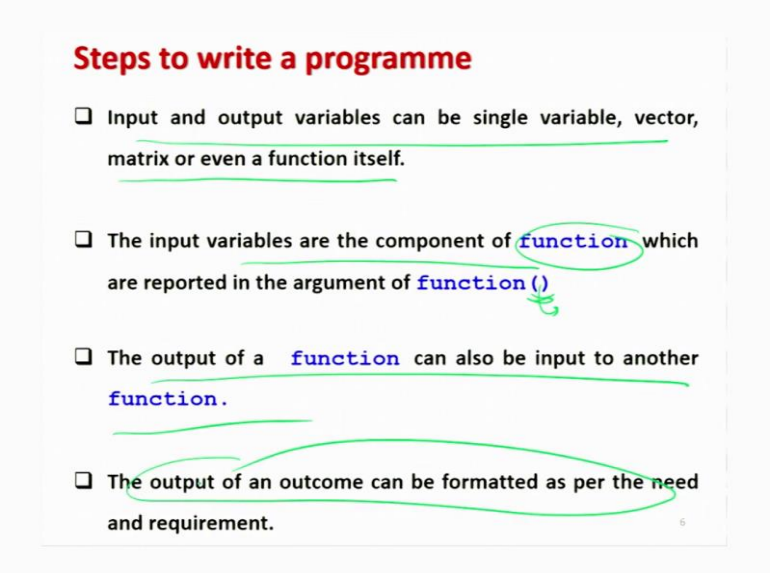
(Refer Slide Time: 03:25)



And we already have discussed that the programmes in the R software, they are written in the form of a function. And in order to write down function or the programme, we have to first define the objective. And we should know what we really want to obtain as an outcome of a programme.

And then we try to translate it to the language of R that, how to communicate with the R using the commands and functions. And in order to do it first we try to identify the input and output variables, we try to identify the nature of input and output variables, like as whether it is numeric, string, factors, matrices etcetera.

(Refer Slide Time: 04:01)



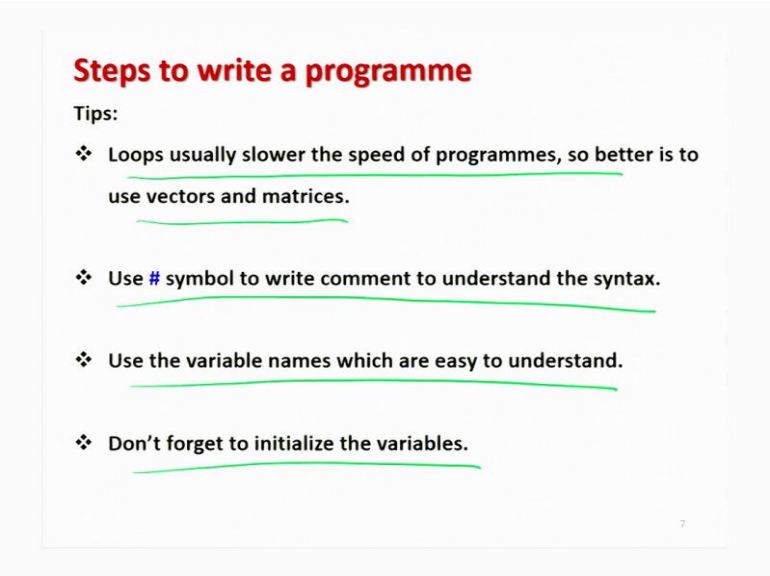
Now, you know all of them, right. And now this input as well as output variables, they can be a single variable, they can be a vector valued, matrix or even a function itself, right.

And whatever are the input variables of the programme, they are the component of this function and these components are reported inside the argument that is here with in the parenthesis of the command function.

One thing you have to keep in mind that whatever is the outcome of one function that can also be an input for another function, right. And that is the biggest advantage of this R software that you can call within a function, a function and that will try to work as an input also, right.

And the output of an outcome can be formatted as per the need and requirement that can be printed, soft copy, hard copy, they can all be arranged, right.

(Refer Slide Time: 04:43)



So, some very basic and fundamental tip for the programmer who are going to use the big data. Usually, it has been observed that the that loops usually slower the speed of programmes, they consume more space in the memory. So, people try to prefer to work with the vectors and matrices which makes the speed of the programme quite fast.

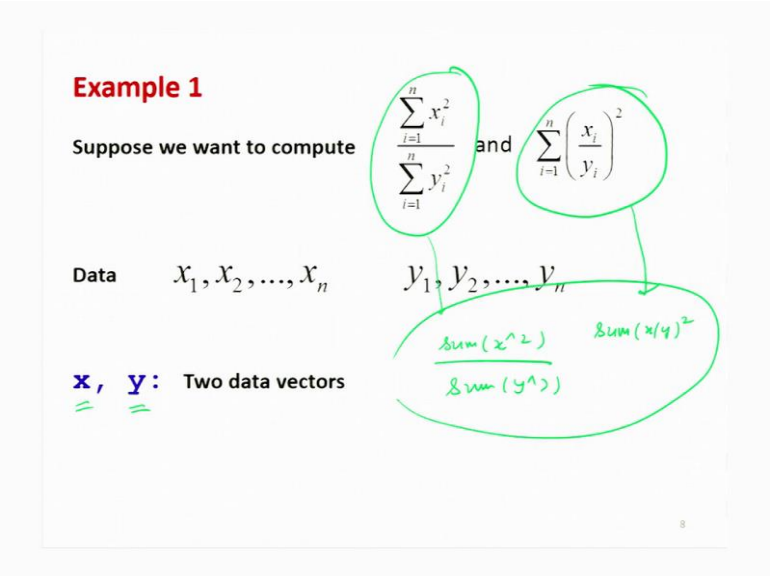
At this moment we are trying to handle very small programmes in which the speed does not make much different, but when you are trying to write down the big programmes dealing with big data sets, then you will see that sometime it takes long time to execute a programme. And to my experience when we are trying to conduct the simulation, this time can be from couple of seconds to the couple of days, right.

And then always try to use the hash symbol to write the comment inside the programme, so that you can understand the syntax at some future point.

And then try to use the names of the variable which are easy to understand. For example, if you have two variables height and weight, so it is better to use height for height and weight for weight. If you try to indicate the height by the variable weight or vice versa then it will be very confusing.

And do not forget to initialize the variables and try to remove it from your memory before you begin. Otherwise, some earlier variables or earlier values will take over and we will never come to know. And the outcome will be very different, ok.

(Refer Slide Time: 06:01)



So, now I try to take here couple of examples. And one thing I can accepted before beginning the example that my objective here is not really to teach you programming or to explain you that how are you going to write the programme.

My objective is very simple, I want to show you that whenever you are trying to write a programme what are the different aspects which you have to keep in mind, and how you have to proceed. For example, in this example, I just want to write a programme of the

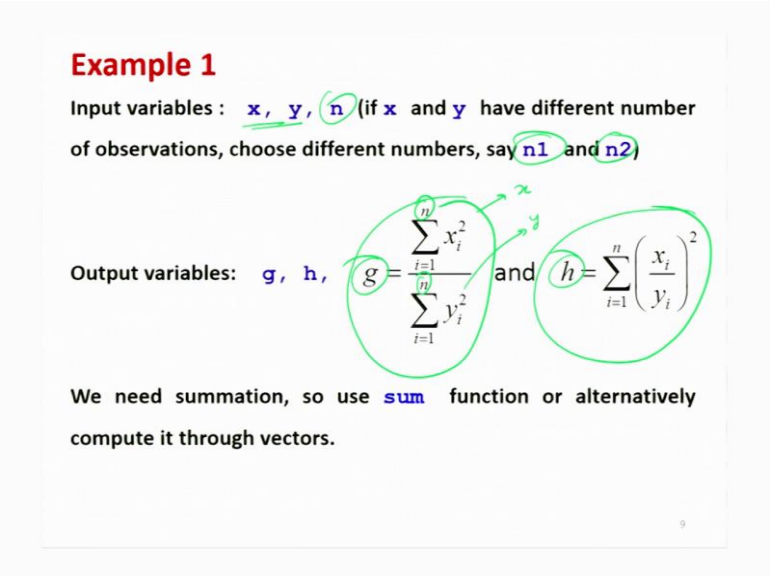
$\frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} y_i^2}$ and $\sum_{i=1}^{n}\left(\frac{x_i}{y_i}\right)^2$. Well, I will show you that this command can be written now

you know in a very simple words that, if you try to have the two data vectors consisting of the values of x and y as x and y then this can be written here as a simply here is some x hat 2 divided by some y hat. Whereas, this can be written here as a some x upon y whole square.

But my objective is not to compute this value, but I want to show you the steps. And well, those steps are going to be a couple of steps and, but finally, I will show you that

the length of the programme is very less. But then, I am trying to explain in a more detailed way, so that it will carry over to the couple of slides. But it does not mean that the programme is long or it is complicated, right.

(Refer Slide Time: 07:12)

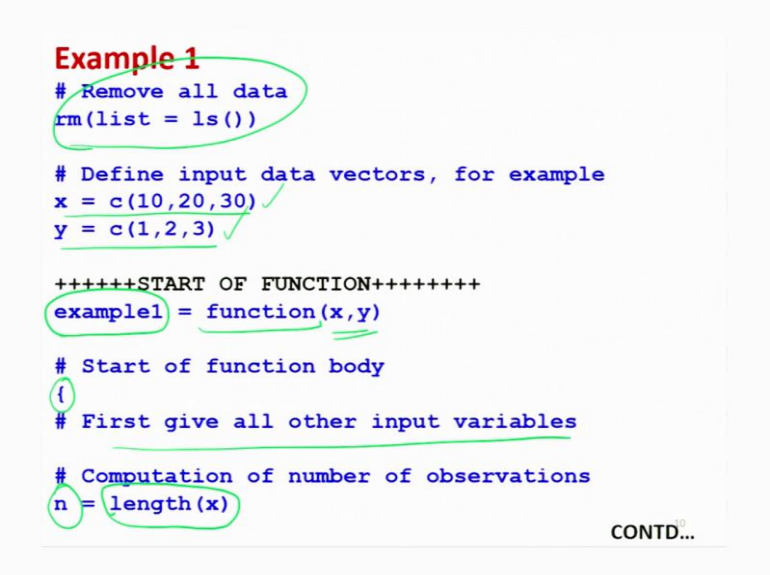


So, suppose I want to compute here these two quantities. So, I try to denote the $\frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} y_i^2}$ as g and $\sum_{i=1}^{n} \left(\frac{x_i}{y_i}\right)^2$ as say h. So obviously, what we need here that, we need to store the values of x i in a data vector x and the values of y i is in a data vector y. So, we try to denote them here x and y. And now you can see here this is here n.

So, in this case, I have taken the summation which is going to the equal number of observation. But if there are say different number of observation, then you can use here n 1 and n 2 also in place of without any problem. And I try to denote this function as say g and this function here as say h, right.

Now, if I try to explain you that how I am going to do here that I wish to show you the application of the loop.

(Refer Slide Time: 08:03)



So, that is why I am just trying to write in a more detailed way, so that you can understand. At least those candidates who do not have very strong programming background, they will also be able to understand, and those who have very good programming language for them it is very easy example.

So, as a first rule of the programming that try to remove all the data from your memory, so that you know what you are going to give that is going to be used later on ah. Definitely, when you want to execute the programme you need some data input values. So, I try to define here x and y here as say like this.

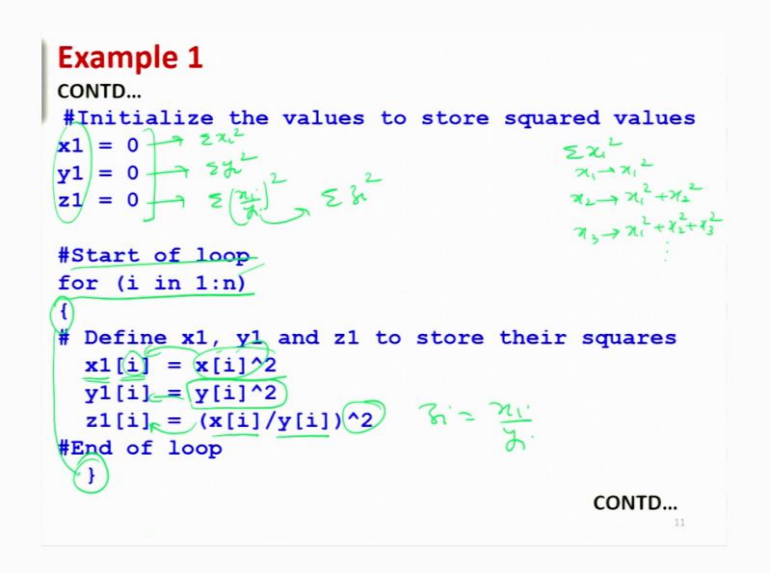
It takes, say 3 values x takes 10, 20, 30 and y takes 1, 2, 3, yeah. You can take anything and this number can be anything. Now, I want to write a programme to find out this g and h, right. So, I just try to give it here a name say here example 1. So, this is the name of the programme.

And after that I will try to write down here the command function f u n c t i o n, and within parenthesis you have to give the input vector. So, now, you have here x and y, and yeah the value of n, I am going to compute it from the number of observations in x and y, right.

So, now, I begin here the starting of the function. So, for that I have to write down here a curly bracket and then after that I have to give here all the input command for the computation.

Now, the first input value comes here which we have not defined is the value of here n. So, for that instead of giving it from outside, I try to give it from inside using the command l e n g t h length of x. Actually, this is always a better idea that when you are trying to compute something try to compute, try to write command in which the things are calculated automatically. Otherwise, they can always be a chance of mistake.

(Refer Slide Time: 09:41)



And now since I want to use here the loop, so I try to see here if I am trying to find the value of a summation x i square. So, what I have to do? First, I have to find out the value of x 1 and then x 1 square, and then I have to find out the value of x 2. And then I have to add it in the say x 1 square like as x 1 square plus x 2 square and then I have to get here the value of x 3, and then I have to add this value x 3 square in the earlier obtained sum.

So, for that in order to begin the loop, I have to take here say some initial value. So, I try to define here one initial value for summation x i square, one initial value for corresponding to summation y square, and one initial value for summation x i upon y i whole square which I am going to call it here as a summation of z i square. So, that is why I have defined here x 1, y 1 and z 1 which all take value 0.

Now, I begin here the loop and I choose here the for loop. So, for loop you have to write f o r and then within parenthesis i in 1 colon n. Now, I need to define here this x 1, y 1, z 1, so that the individual values of say x i square y i square and x i upon y i whole square they can be stored.

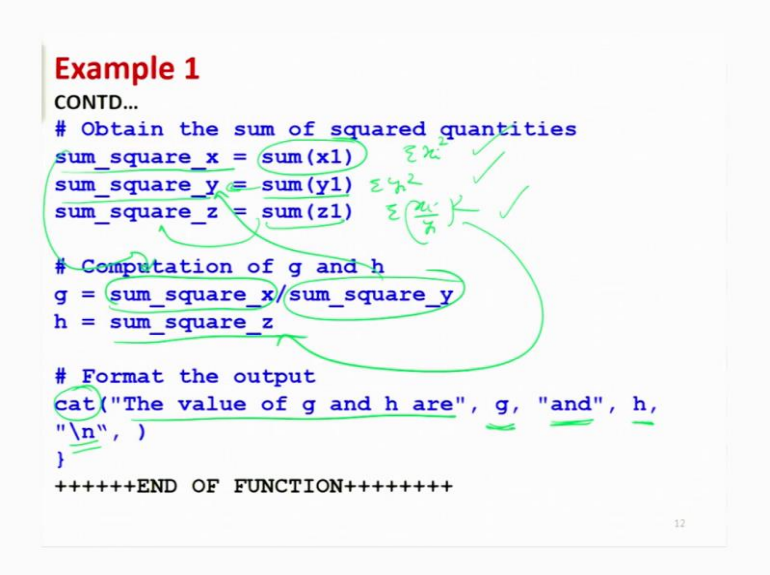
So, I try to define it element wise, so you know that when you are trying to write down here x 1 and inside the square bracket you are trying to write down here i; that means, the i-th value in the value will be stored here. So, I write here x 1 i is equal to x inside the square bracket i and then whole square.

So, whatever is the i-th value of the data vector x, that will be squared and that is going to be stored in the variable x 1 at i-th location. Similarly, the value of y i square will be computed by the command y square bracket i whole square and that is going to be stored in the variable y 1 at the i-th position.

And similarly I try to define here one more variable here z i. So, z i is something like here x i upon y i. So, this becomes here x square bracket i divided by y square bracket i and x square. So, now this value is going to be stored in the new variable here z 1 at the i-th position, right.

So, now, this completes my loop and I give here this bracket. So, this loop is you can see here within these two brackets, these two curly brackets, right.

(Refer Slide Time: 11:51)



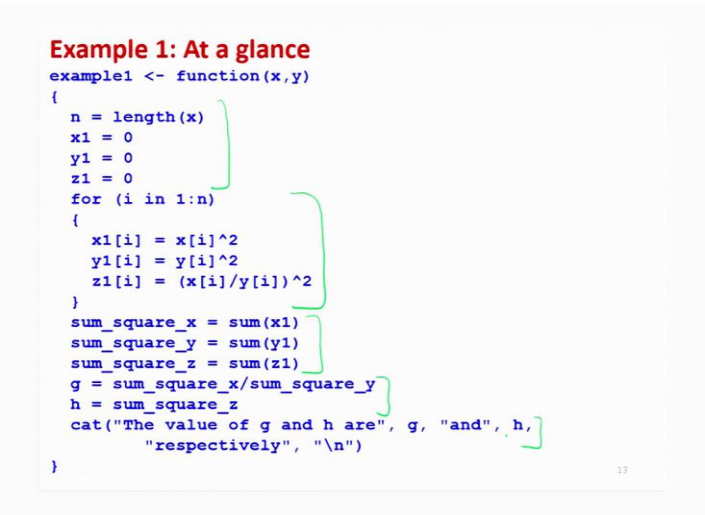
And now I have to obtain the sum of all the values in the x 1, y 1, z 1. So, I try to define here a variable here some underscore square underscore x and then using the function sum I can find out the summation x i square as say here sum of x 1.

And similarly, I can find out $\sum_{i=1}^{n} y_i^2$ and $\sum_{i=1}^{n}\left(\frac{x_i}{y_i}\right)^2$ by using here the command sum of y 1 and storing it to some underscore square underscore y and sum of z 1 and storing it to a new variable sum underscore square underscore z, right.

So, now, I have obtained here this summation x i square, summation y square and summation x i upon y i whole square. Now, I need to compute my here g and h. So, g is simply your here $\frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} y_i^2}$. So, I try to define here this sum of square of x divided by sum of squares due to y from these two values. And then h that is already the sum under underscore z, this has been obtained here also.

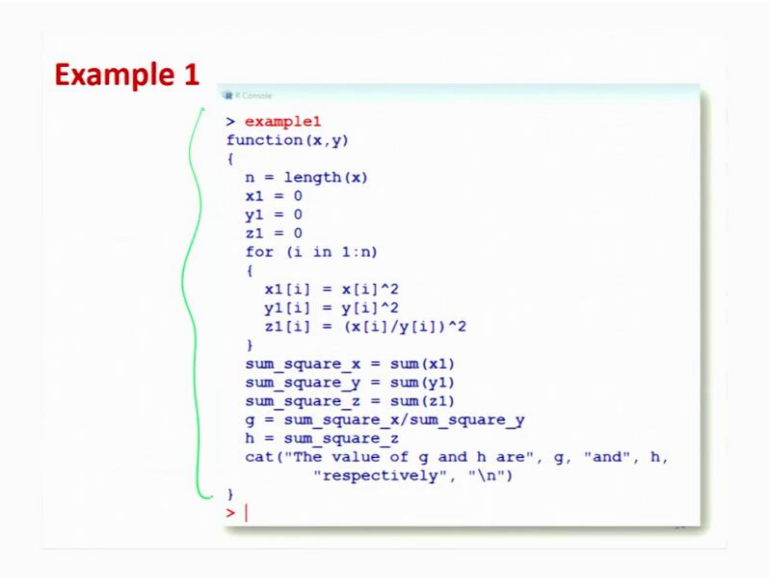
Now, after this I want to take the output in the format that the value of g and h which is a string is like this R, the value of here g, and then the string h, and then the value of here h and then here next line. So, now you know all these commands. So, I use here the command here cat function and I want here a formatted outcome. So, now, after this I will get here the outcome in the way I want.

(Refer Slide Time: 13:12)



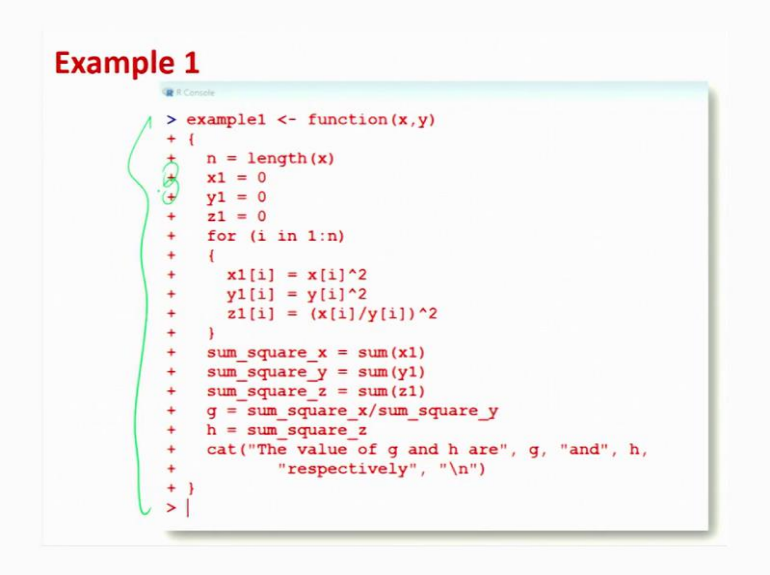
And if you try to see this whole this function in a single slide, you can see that it is not a very difficult programme. I have just given here the input, and initial values, and here it is here the loop function and then I have here defined the sum of the squares and then I am defining here g and h and here is the outcome, that is all, right.

(Refer Slide Time: 13:30)



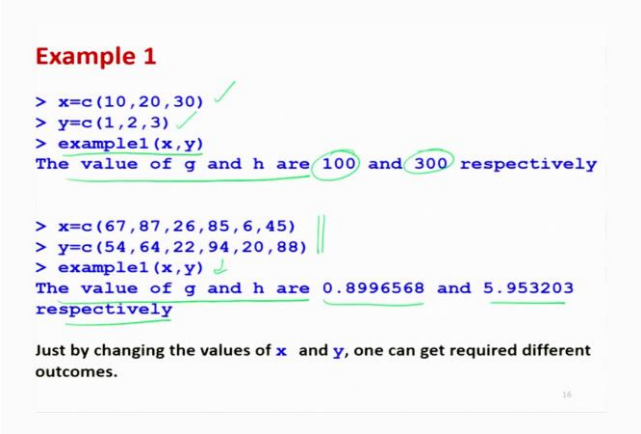
And if you try to see it in the R console also, it will look like this.

(Refer Slide Time: 13:34)



And means, when you try to type it inside the R console, there will be here plus sign. Now, you know what is the meaning of this plus sign, right.

(Refer Slide Time: 13:41)

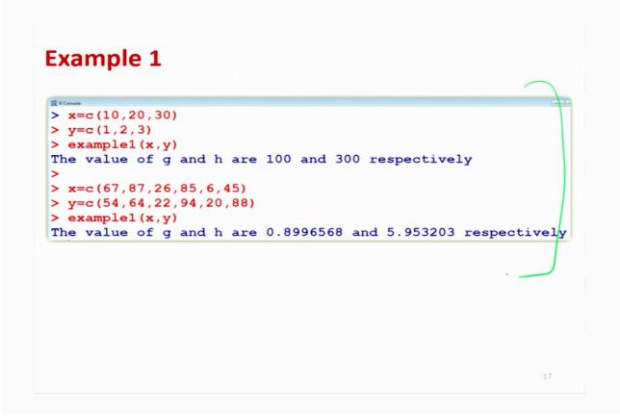


And now after this since you have taken the value of x to be here 10, 20, 30, the value of y to be here 1, 2, 3. Then you try to write down here example 1 and within parenthesis x, y. To execute it the outcome will look like this, the value of g and h are 100 and 300, respectively. And now you know how this outcome is coming, right.

Similarly, if you try to take here one more example in which you are trying to take different values of x and y, and then you try to execute it, now, once again the similar output will come in which the new values of g and h are going to be reported. The value of g and h are here like this and this, respectively.

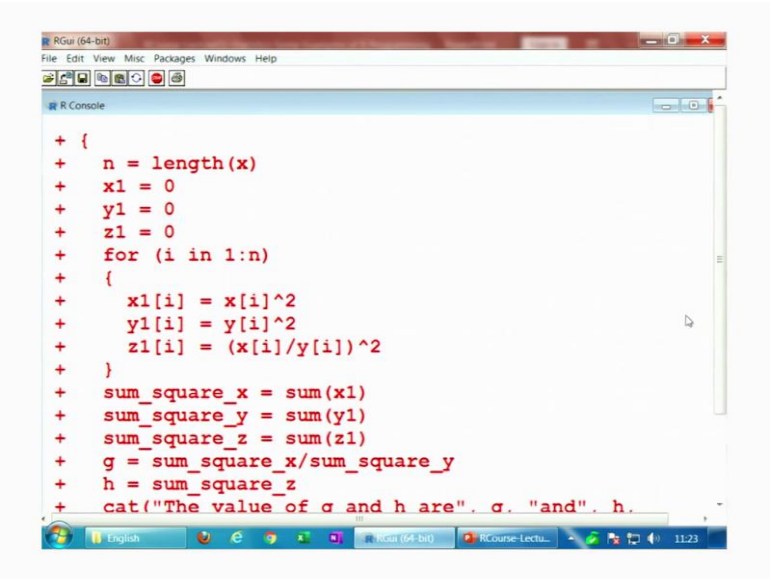
So, you can see here another advantage is that just by using the values of say x and y, you can very easily execute the programme as many as times you want, and depending on the value of here g and h. And this is here the screen shot.

(Refer Slide Time: 14:26)

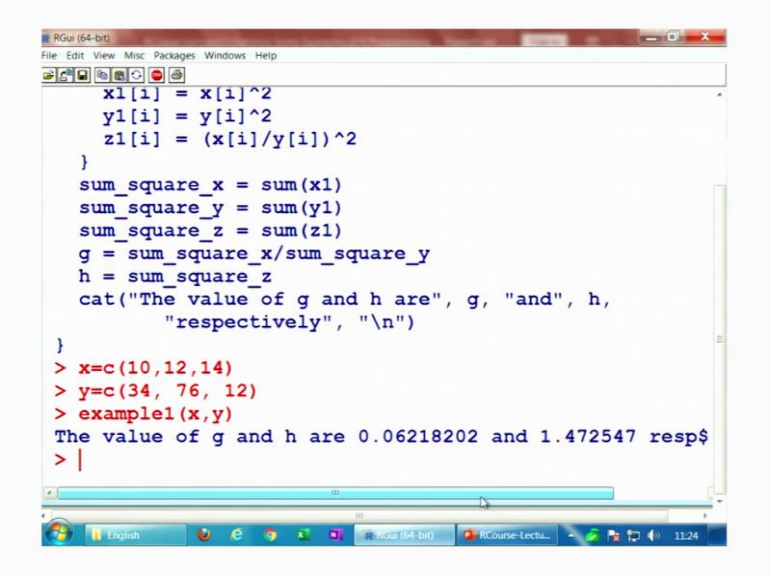


So, before I try to move forward, let me try to show you this example on the R console. So, I try to just copy this command on the R console, so that you can see here this is here like this, right. So, if you try to see here your programme here is like this example 1, you can see here like this, right.

(Refer Slide Time: 14:41)

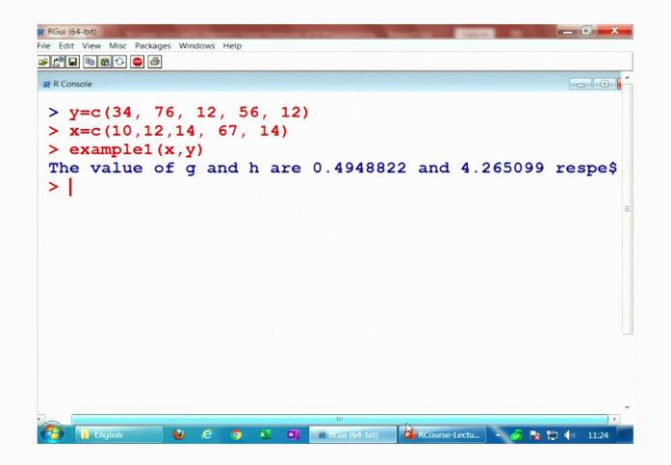


(Refer Slide Time: 14:43)



So, now, if you try to define here x is equal to suppose here is 10, 12, 14 and then y here as say here c, it can be anything 34, 76, 12 and so on.

(Refer Slide Time: 15:16)

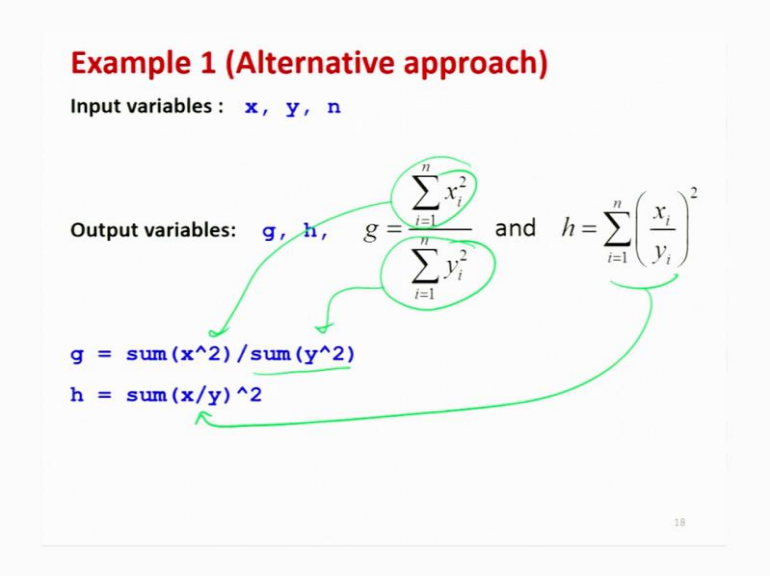


And then if you try to write down here example 1, and then x and here y, see here the outcome will look like here this, right. The values of h, g and h are like this.

And in case if you try to replace or you try to change the values of here x and y here like this, suppose if I want to add here some more values here like this. And then similarly, if I try to take here some values of here x like this, I try to now make it here 5 values in place of 3 and if I try to use an example 1, x, y you will get here this new values and you can see here this is the value of g and h are like this, right.

So, you can see here it is not a very difficult job to write such programme and to execute them, right.

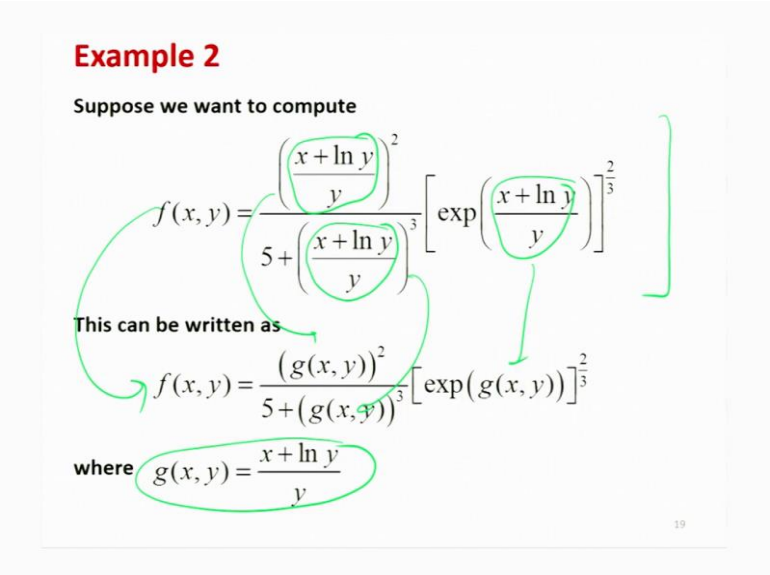
(Refer Slide Time: 15:43)



So, now, as I told you that if you really want to ask me how to write this programme efficiently, then I can write down the values of g and h in a single line like this. This is quantity is your here some of x square and summation y square summation like in the sum of y hat 2 and then this here is here like the sum x upon y and say hat 2.

So, we can see here just by using the commands of the R software which I have defined in a very mathematically friendly way, you can write such functions in a single line, right. So, that is that one test, that is what I was telling you, right from the beginning that this R programming and built-in functions help you a lot writing the programme compactly and efficiently, right.

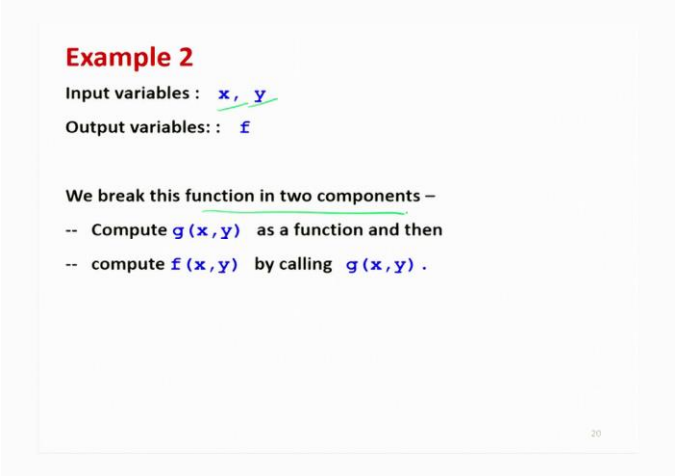
(Refer Slide Time: 16:24)



So, now, let me try to take here one more example and where I try to show you that I want to compute this function, right.

So, if you look at this function very carefully, you can see here this is here a function $\left(\frac{x+\ln y}{y}\right)$ which is used here at say here 3 places. So, I try to rewrite this function as here g(x, y), and then this $f(x,y)=\frac{(g(x,y))^2}{5+(g(x,y))^3}\left[\exp(g(x,y))\right]^{\frac{2}{3}}$ and $g(x,y)=\frac{x+\ln y}{y}$, right.

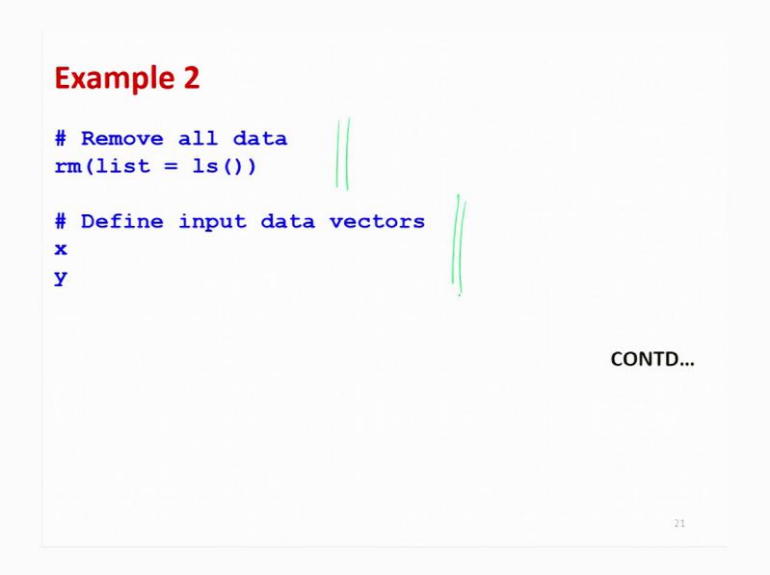
(Refer Slide Time: 16:58)



Now, I want to show you something here. So, you can see here in this case there are two input variable, say x and y, and the output is that, ok, you want to compute the say here function and you want to know the value of your f(x, y).

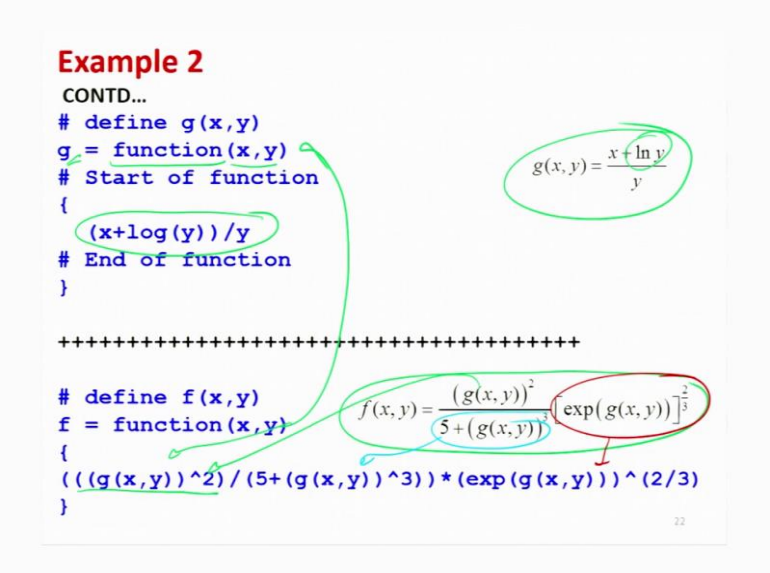
So, but if you try to see here that this f(x, y) this is depending on the value of here g(x, y). So, what we can do here, that we can compute here g(x, y) and use it as an input variable inside the f(x, y). So, how to get it done? And the advantage that the more complicated programming can be broken into different components, simple components and different people can write the programmes, and then this programme can be joined together without any difficulty.

(Refer Slide Time: 17:33)



So, as usual I will try to remove all the data and then I will try to define here the input variable that is the first step.

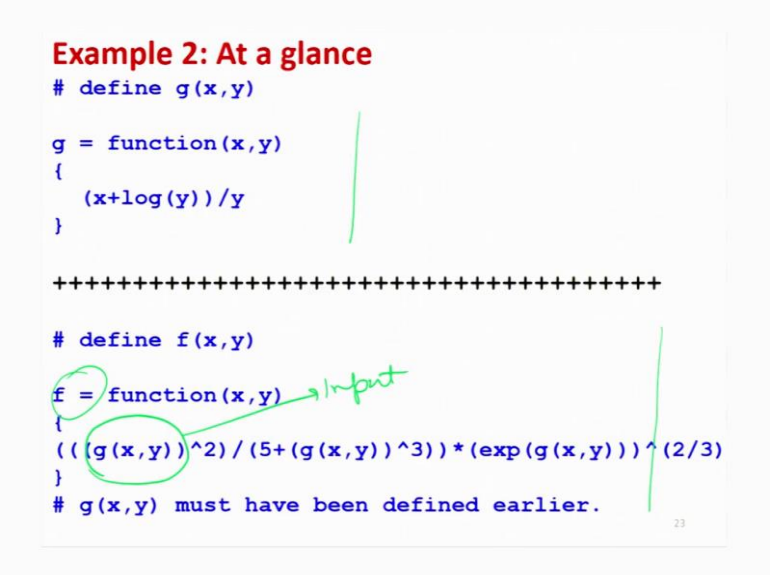
(Refer Slide Time: 17:39)



Now, in the second step, I simply try to define here g(x, y) and f(x, y). So, you see now this g(x, y) here is like this. So, writing down this function is very simple, defined here g and then function and then within parenthesis x, y and it is simply your here x plus log y divided by y. So, that you already have learned that how to define this natural log and then this will complete your function.

And similarly, if you try to now define here this function here, so you already have defined here this here g here, right. So, now you would try to bring this value of g here directly here. So, it is like here you can see g(x, y) whole square which is here, then this 5 plus g(x, y) cube which is here, and then after this it is exponential of g(x, y) raise power of 2 by 3 which is here, right. So, you can see here now I have written this programme.

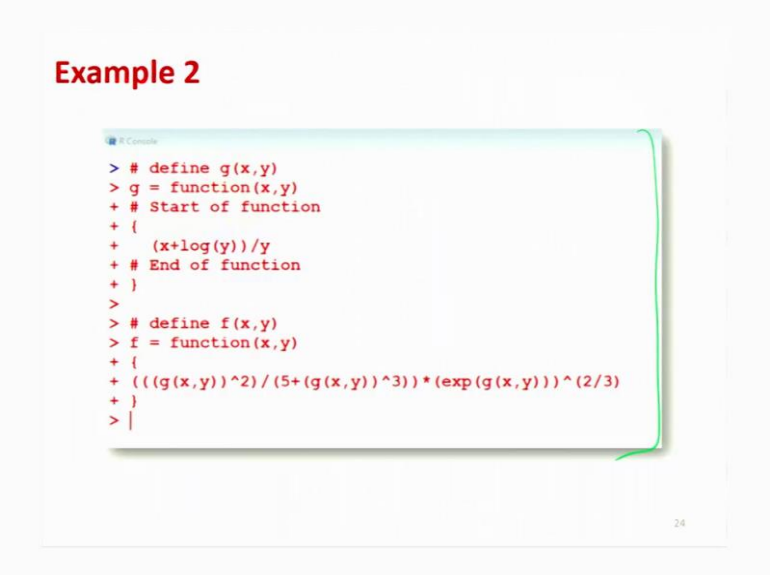
(Refer Slide Time: 18:29)



Now, in case if you try to see briefly in a single slide that how this programme can be written. This is just your here g(x, y) and this here f(x, y), yeah.

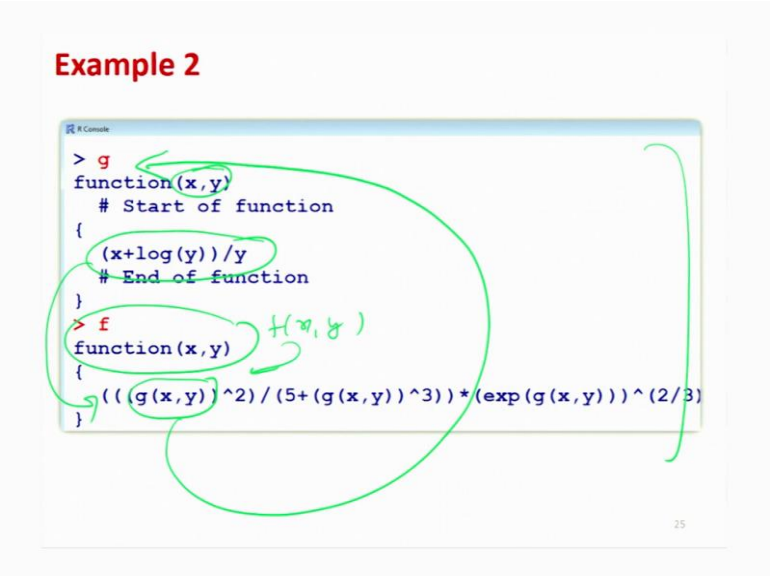
Now, if you try to see one very peculiar characteristic of this R programming. Here you are trying to use here g(x, y) as an input, so that means, when you try to execute this function here f it is going to look for the value of g(x, y) and the value of g(x, y) is being computed externally outside this function. So, the function g should also be available at the same place where you are trying to write down the f function, right.

(Refer Slide Time: 19:06)



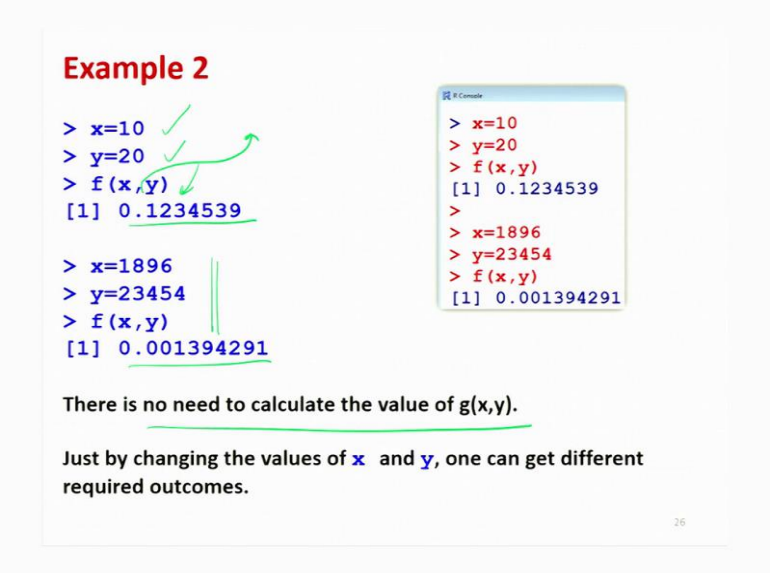
So, this is how I try to write down.

(Refer Slide Time: 19:08)



And this is my here the screenshot of this programme.

(Refer Slide Time: 19:10)



Now, if you try to see what I am going to do, that is very interesting and this is what you have to observe. I simply define here x and here y, and I execute here f(x, y) and we are getting here this value.

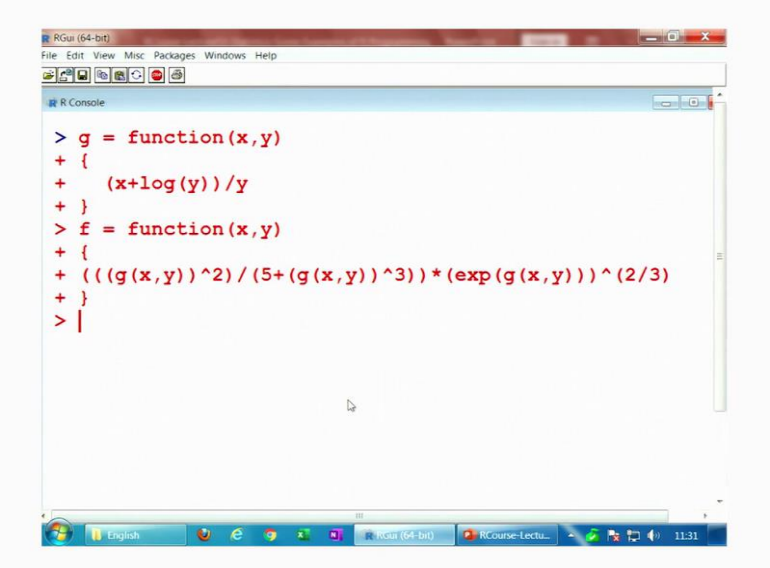
Now, think and tell me where is your g(x, y)? You have not computed the value of g(x, y), but you are simply computing here the value of f(x, y). But what is happening that when you are trying to compute the value of f x, y this programme goes outside and then it tries to compute the value of g(x, y). Just like here I can explained you on this screen shot.

When you are trying to compute here f(x, y) then control comes over here and it tries at, ok there is a function here g(x, y). So, this function comes out of this programme and in the same directory it tries to search where is this g(x, y). So, it is going to take the same input value x and y to this function g x here, and it will try to compute this function and then it will try to bring it here.

So, you can see here the function is automatically jumping out of the function. It is going to the other function. It is trying to compute the values and it is trying to bring back the numerical values as an input to the next function, right. So, this is what is happening here.

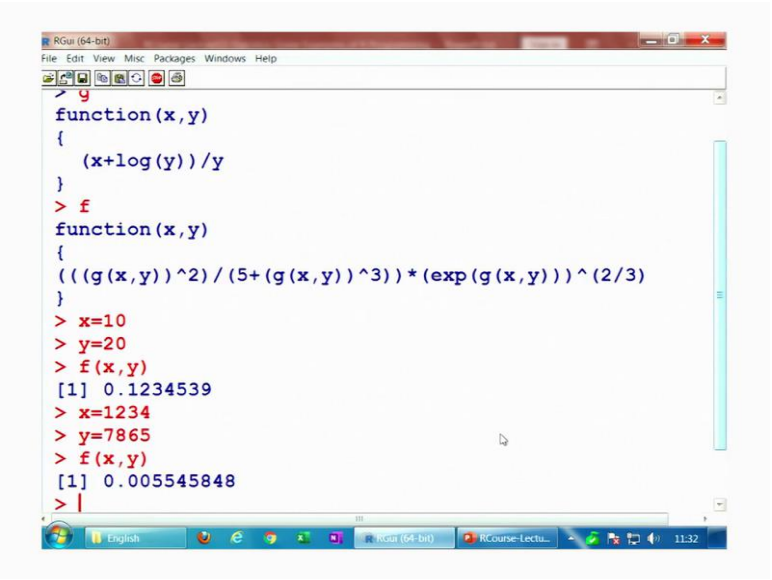
And similarly, if you try to change here the value of here x and y, you will get here new values. And you can see here there is no need to calculate the value of g(x, y). So, now, if you try to extend this concept to bigger example that there is a very complicated function, and as a programmer you try to divide the programming in say here different components.

(Refer Slide Time: 20:43)



And you try to give it to the different people in your organization, and all of them they are trying to just write a programme for a small section of the programme. And then you are trying to just use all of their programme and try to combine or try to call them in a single programme, right.

(Refer Slide Time: 21:07)



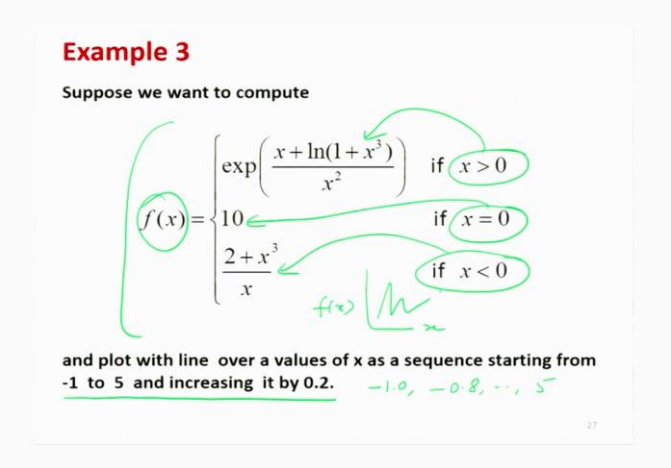
So, let me try to show you here this here these two functions. So, I have copied here the function for this g and f. So, you can see here what is your here g and this is your here f. Now, if you try to see, I try to choose here some value of here x say 10 and y is equal to 20.

Now, if you try to see what I am going to do, I am simply trying to going to execute here f(x, y). I am not computing here g(x, y) just before you, right. So, if you try to see here as soon as I enter it gives me here this value.

So, f(x, y) from here it has used the value of g(x, y) for that it has gone outside this function and then it has brought the numerical value of g x, y and it has used it. Similarly, if you try to take here any other value of x like this and if you try to repeat this command you can see here this is here f(x, y), right.

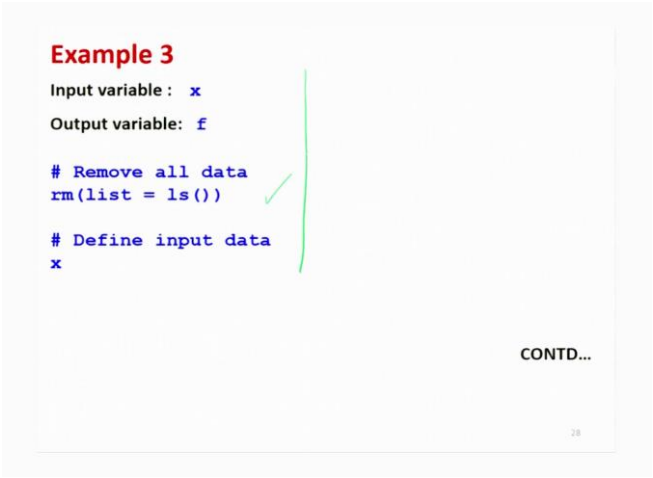
You have nowhere use the value of g(x, y). But that is automatically computed that is the very strong feature of this R programming which is quite popular actually. And that is why this R software gained the popularity, ok.

(Refer Slide Time: 22:00)



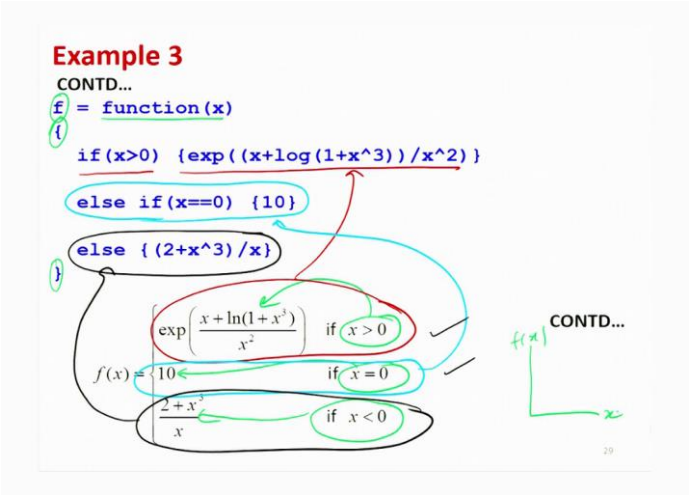
Now, we consider here one more example and you have seen this type of functions in mathematics and our objective is this, I want to make here a plot between here x on the x axis and f x on the y axis, right. So, what this function is trying to say? It has divided the range of x into 3 parts when x is greater than 0, this function is given by this function. When x equal to 0, then the value of this function is given by 10. If this x is negative, then the value of the function is obtained by $\frac{2+x^3}{x}$. So, I try to take here some values of here x on the x axis and corresponding to which we need to compute the value of x first and then we have to plot. So, we try to choose here the values of x between minus 1 to 5 and I try to increase it by say 0.2. So, this will be minus 1.0 to minus 0.8 and so on up to here 5. So, I will try to take these values on the x axis, the values of f x on the y axis and then I will try to make here a plot.

(Refer Slide Time: 23:01)



So, now you can see here by looking at this function you can see very easily that you can use here the if-else condition. So, for that first you try to do the basic operation, try to remove all the data and try to define the input variable.

(Refer Slide Time: 23:14)



Now, in order to use the if-else condition you know that if this condition is true, then this is going to be executed and if x is equal to 0, then x is going to be, then f x is going to be here 10. And if both of them are not true, then whatever is left that is going to be executed here like this.

So, now, I try to use the if-else-if condition and if-else-if condition execution that you had learnt earlier and you can see here you can write this programme very easily. So, first I try to take here the first condition that if x is greater than 0, then f(x) is going to be computed by this exponential of x plus log 1 plus x cube upon x square.

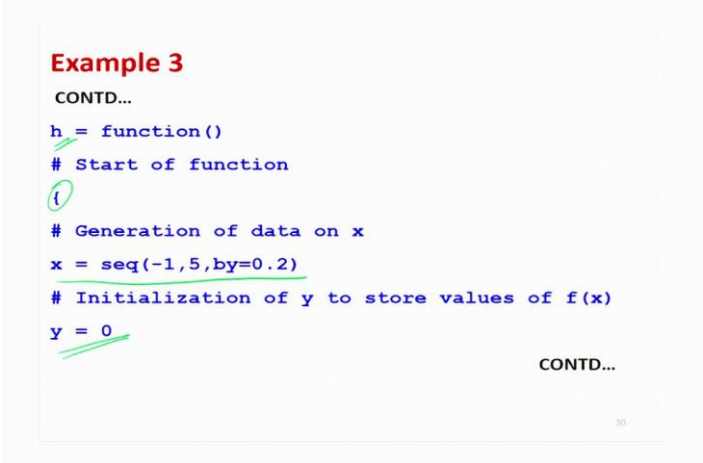
And I try to write down here if x is greater than 0, then within the curly brackets I write this expression. Now, you know how to write down this expression in the R language. And then I try to give here second option that if x is exactly equal to 0, then the value of f(x) is going to be 10. So, this I write down here that if x is exactly equal to 0, then the value of f(x) is going to be 10 and this is your under else-if.

And now finally, if you try to see your third option here, that if the first two condition are not correct, then obviously, the third condition is going to be correct and then it comes under here else and then $\frac{2+x^3}{x}$. And all these things I have enclosed in the programme

name or function name f. I write down here f is equal to function within parenthesis, then put variable here x, and I try to write down all these 3 commands inside a curly bracket.

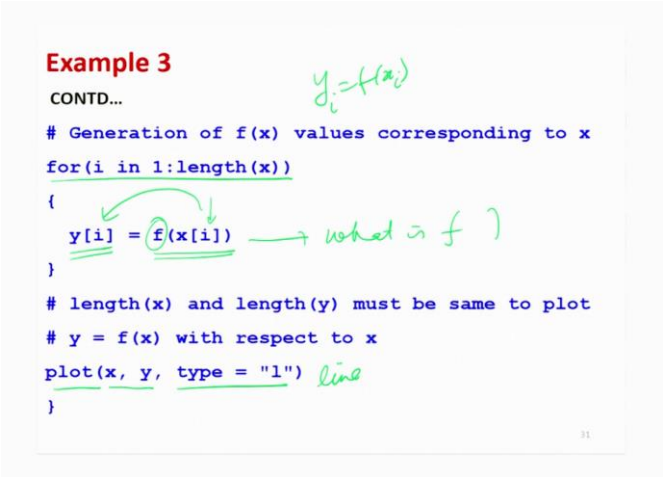
But now here you have given the programme to compute the f(x). Now, you need to write down the programme that how you can create the curve between x and f x. So, for that you try to define here x and then plot command.

(Refer Slide Time: 24:53)



So, I try to define here a new programme here h and then I start here the programme. So, first I have to generate the values of x. So, I try to use here a command sequence from minus 1 to 5 by 0.2, and then I try to define here the value of f(x) as here y and then I try to initialize it, y equal to 0.

(Refer Slide Time: 25:11)

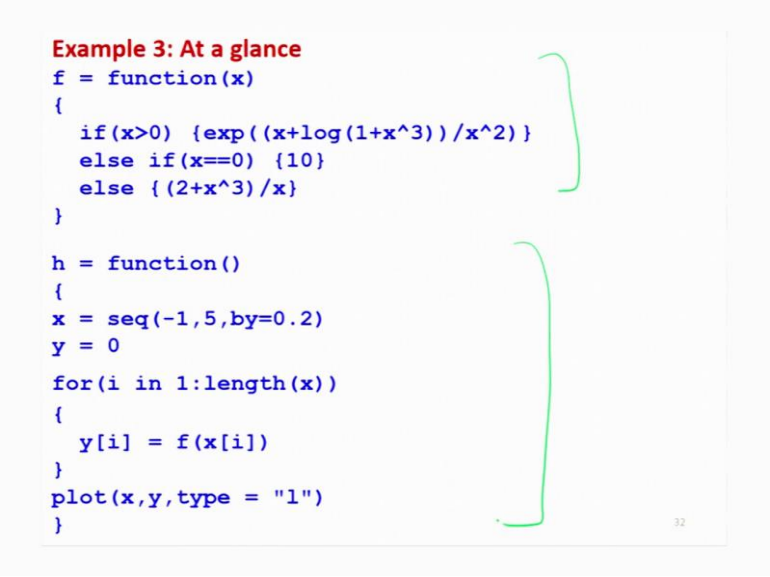


And then I have to compute the value of f(x) that is your here y for each x i. So, it will be like here that I want to compute the value of y i for each value of x i, right. So, for that I try to write down here a loop. Well, you can do it directly also, but I am trying to write down here a loop. And then I try to write down here y square bracket i is equal to f of x square bracket i. That means, for each of the ith value in the x, the value of y is going to be computed using the function f.

You are nowhere given here f. What is here f? That we do not know because this has been defined externally here. You can see here, this is not a part of this programme. So, once again as we get in the earlier example, it will go outside the programme and it will try to compute the value. It will bring it inside the programme, and it will try to execute it.

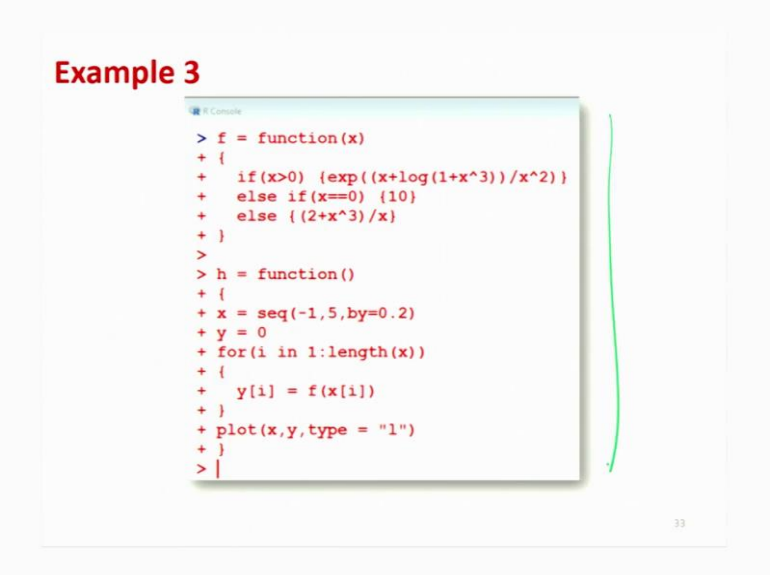
So, then I try to use here the command here simple plot command, plot x, y and then type I am going to take here l that is line, right. So, this is my programme.

(Refer Slide Time: 26:08)

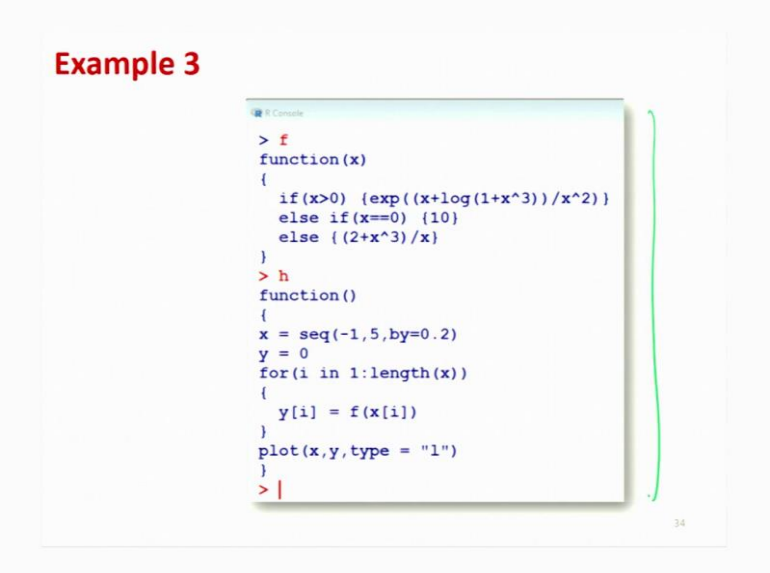


Now, you can see here this programme is very simple. This is your here f and this is your here h, that is all.

(Refer Slide Time: 26:13)

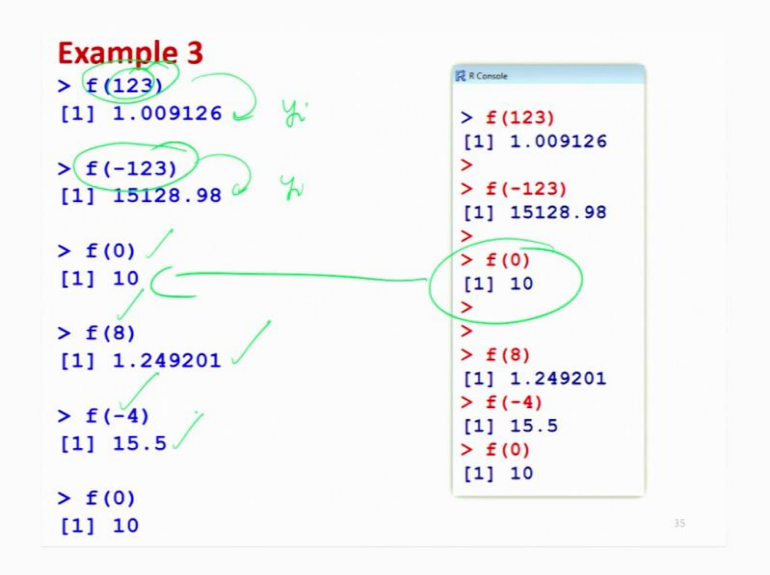


(Refer Slide Time: 26:15)



And if you try to now execute it on the R console, this will be your here screenshot of the programme.

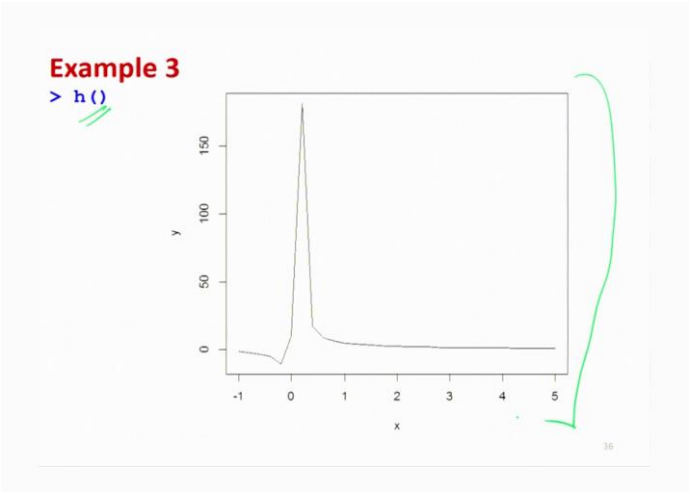
(Refer Slide Time: 26:18)



And now if you try to see here, we do not want it, but just for the sake of illustration if you try to take x equal to 1, 2, 3, then the value of f is going to be like this. If you try to take here x equal to minus 1, 2, 3 then the value of f x is going to be here like this. So, these are the values of here y is, right.

Similarly, f of 0 you can see here f of 0 should be actually here 10 because your function at x equal to 0 it should be equal to 10. So, and then if at x equal to 8, f value is here like this, at minus 4 the value here is like this and so on, right.

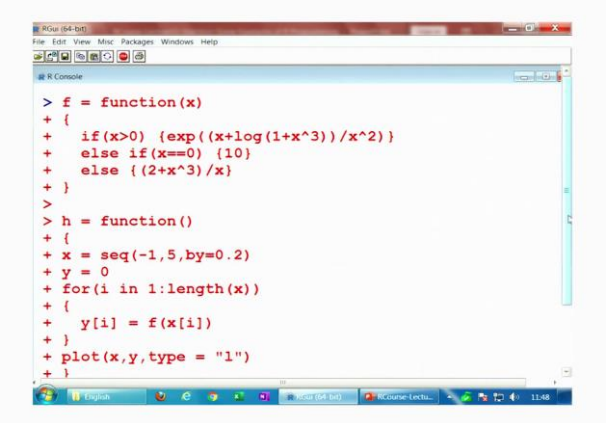
(Refer Slide Time: 26:49)



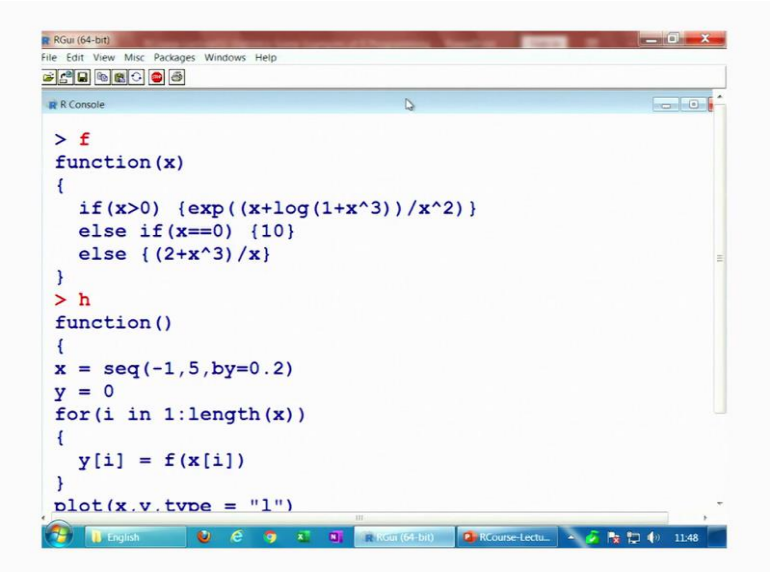
So, now if I try to execute my here function x, so what it will do? It will try to choose, actually it will try to generate the value of x and it will go outside the function h and it will try to compute the value of f x i. And then bring it inside the programme and it will use the plot command to create this plot and the plot will look like this. You can see here it is not a very difficult job. And you can very easily do it on the R console, right.

So, let me try to execute this programme on the R software, and then we try to see that. So, I try to simply copy here both these functions and you can see here, these are your here functions h.

(Refer Slide Time: 27:23)

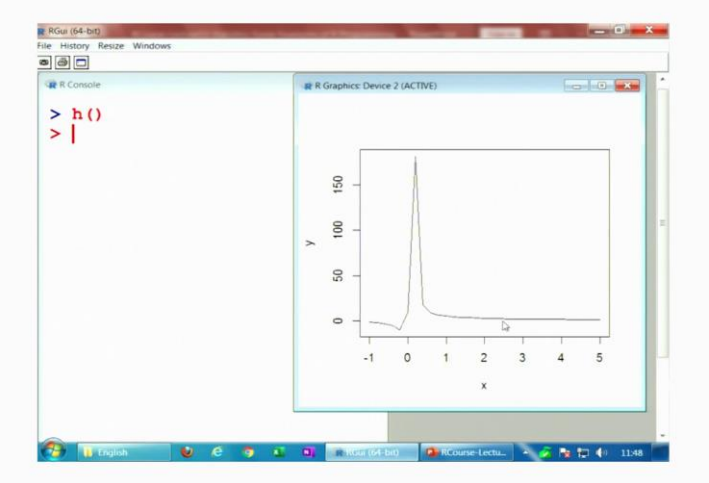


(Refer Slide Time: 27:26)



So, if I try to see here this is your here f and this is your here h, right. And now we are going to create here a plot. Let me try to adjust my screen here.

(Refer Slide Time: 27:35)



Now, if you try to see here I just want to; I am not using here any of this here function f. I am simply using here h and as soon as you enter you can you will see here you will get the this type of curve, right.

And then similarly if you try to change the value of here x according to your choice, if you try to change the values of say here f(x) according to your choice, then you can create such a graphics without any problem and you can plot such functions without any problem in the R software.

So, now, in this lecture, I have taken 3 examples and my objective was essentially to show you that when we are trying to use the programming in the R software, then the input of a programme can also be another function which is defined outside this programme. And this is a very strong feature in the R software and which made the R programming very popular.

And now you have seen that within the R programme also, you can use the R programming language and you can use the built-in packages also. And you have combined both the things together in order to find out the sum mean.

You do not need to write the programme separately, but you can simply write down the function like sum, mean, etc. And all these functions as a very obedient friend they will try to give you the correct value.

And now you can see here whatever you have done, means I have taken here very simple examples in the entire course to convince you that if R is a free software, it does not mean that it is going to give you the wrong value.

And when you are convinced with this smaller value which you can compute manually, with your own hand, then you should be confident about some bigger value, bigger data sets, and some complicated computation, that R is very dependable and R is always going to give you the correct value unless and until someone has made the mistake in the programming.

So, now with this lecture, I come to end to not only to this lecture, but to the course also. But I am going to finish this course from this lecture, but your journey towards the R software is going to begin from today after this lecture.

I have tried my best to select or to and to give you some selected commands, syntax, during this entire course. But as I said many times this is not the end of the list of the commands and functions which are possible in the R software, there are many more function, many possibilities are there and in the last two decades R has made a very good progress.

Now, R has diversified to many areas in which people are trying to use it, but surely you will agree with me that covering all the topics in the same course is not possible. And I and as I said in the beginning itself my job, my aim was not to teach you the R software, but my aim was to take out the fear of learning R from your heart, that is all. And I believe that now there is no more fear in your heart, and now you can start flying and the sky is the limit for you to reach in your life, in your career and in the use of your R software.

So, I wish you all the best. And I wish and pray to god that you achieve good life, you get more successes. And I will see you sometimes, somewhere, in some other course or sometime physically. Till then, may god bless you. And I will see you, till then goodbye.