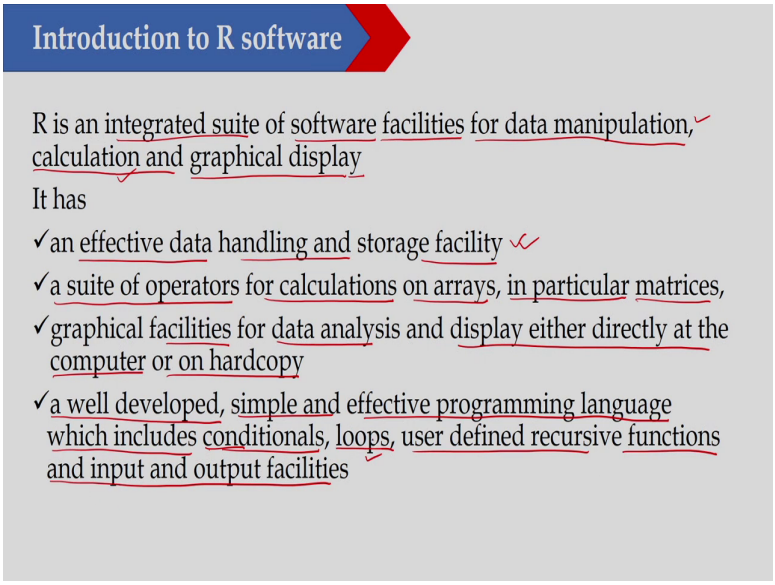
Optimization Methods for Civil Engineering
Prof. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 22
Introduction to R Software

Hello student, welcome back to the course on Optimization Methods for Civil Engineering. So, today I will introduce you R software ok. So, in this class, so, we will try to learn the basics of R, how what is R, how you can download R and then how to write simple lines how to do simple mathematical calculation so that I will discuss in this class.

(Refer Slide Time: 00:57)



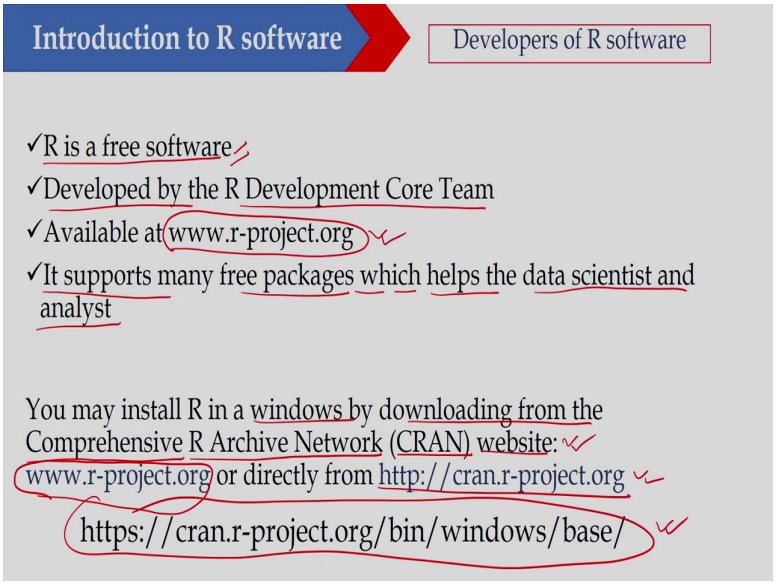
Now, what is R? R is an integrated suite of software facilitates for data manipulation, calculation and graphical display. So, it is an integrated suite and where you can do the data manipulation, you can do the calculation and you can also do the graphical display of your

results. So, it has an effective data handling and storage facility. So, in R you have effective data handling and storage facility, a suite of operators for calculation on arrays in particular matrices.

So, it has the graphical facilities as I have said graphical facilities for data analysis and display either directly at your computer. So, you can display on your computer screen or on a hard copy. So, you can also take a print out of that one. It has a well developed simple and effective programming language which includes conditional, loops user defined recursive functions and input and output facilities.

So, it has the facilities so where you can do the programming you can write the logical statement you can write the conditional statement you can use loops and you have also user defined recursive function and has the input and output facilities.

(Refer Slide Time: 02:44)

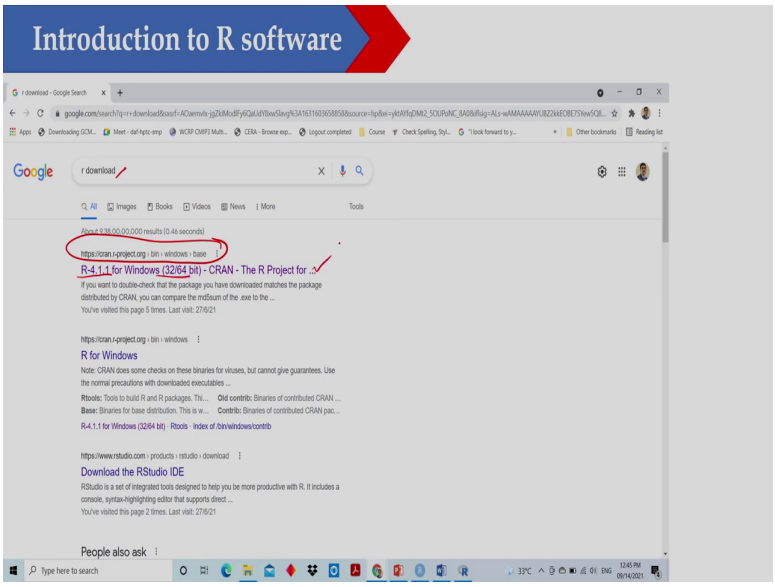


Now, this is a free software. So, this is the main advantage of using R. So, it is a free software. So, you need not purchase it, you can simply download from the net and you can use it. So, today I will show you how you can download this one. Then developed by R development core team and available in this particular website, I will show you this website how you can download that one, it supports many free packages. So, which help us the data scientist and data analyst?

So, I can also install the free packages. So, I will also show you there are some packages. So, that you can install and you can use for your work ok. So, you may install R in a Windows by downloading from the Comprehensive R Archive Network CRAN website. So, you can download from this particular website. So, this is the URL of this particular website. So, you can go up go to this and you can download it. So, I have shown I have given you the URL. So, you can visit this URL and you can download the software.

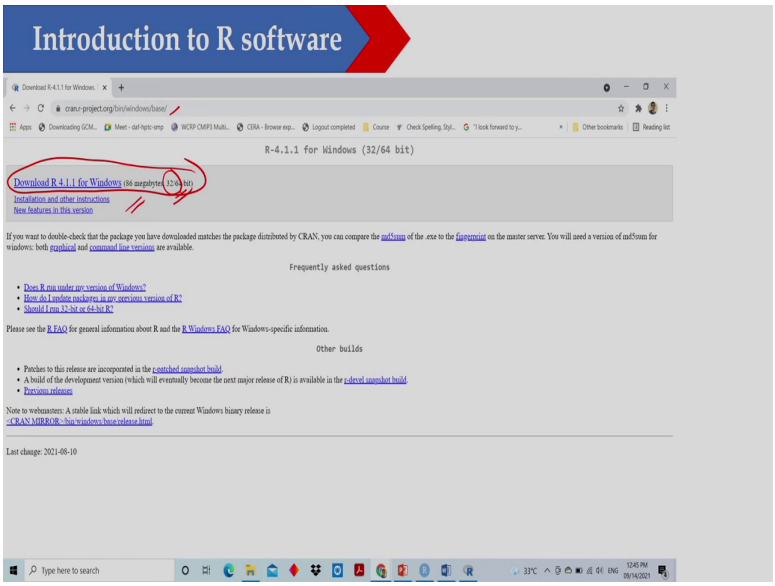
So, today I will show you how you can download this on a Windows platform, but if you are using Mac or if you are using UNIX then also you can install it. So, the free version of R can be installed on a UNIX system as well as on your Macbook.

(Refer Slide Time: 04:22)



So, what we can do basically directly you just write R download. So, in that case you go to this particular, Google will show you this one. So, we have that R 4.1.1. So, this is the Windows version. So, this is the website.

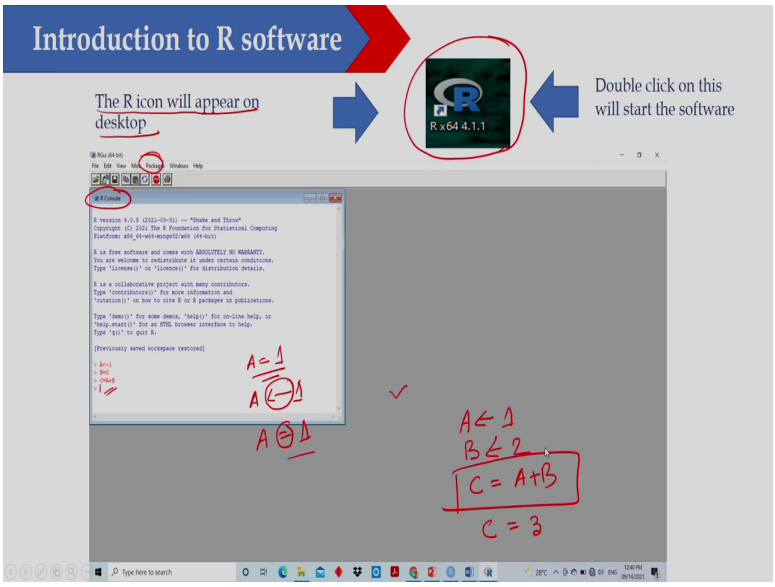
(Refer Slide Time: 04:41)



So, if you are clicking here, so, you will be getting this particular website. So, here from where you can download the software. So, I am using this one. So, I am downloading that 4.1.1 for Windows and it is for 32 and 64 bit. So, I will be using 64 bit version yeah. So, I can download.

So, you click here and you can download the executable file and installation is not that difficult. So, you just follow the instruction and you can install the software. So, I am not showing here, but this is not that difficult. So, you can do that.

(Refer Slide Time: 05:24)

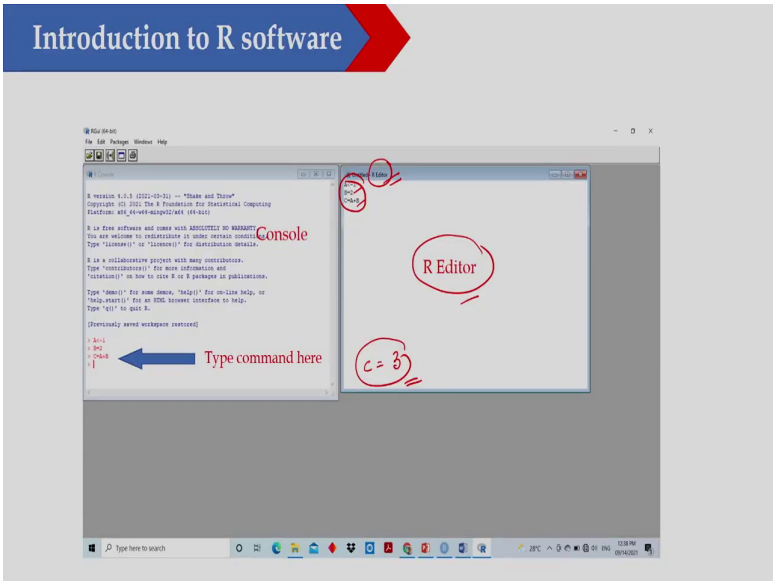


Now, once you are installing the software, so, this R icon, so, this is the R icon. So, R icon will appear on your desktop ok. So, it will be there and you just double click on this particular icon then R will be open. So, this is the R Window and you can see that this is the R console. So, R console here you are getting and then you have the other option, you can open the file, you can edit, you can view and you can see also packages. So, there are different options. So, I am not discussing here, but this is the R console you are getting and here you can actually do your programming.

So, you can execute some of the; suppose here in the first line I am assigning that A is equal to 1. So, I am assigning this value. So, how I am writing that A you can write it; you can write it the A equal to 1 or you can write A equal to 1 also. So, either you can use this symbol or you can also use equality symbol for assigning a value to a variable. So, you can do that.

Similarly, here what I have done that A equal to 1 then B equal to 2 and then I am calculating C equal to A plus B. So, this is very simple. So, it will give the value of C and C will be equal to 3 ok because A equal to 1, B equal to 2 and C equal to 3. So, I can do the calculation here itself. So, even I can also write a complicated statement here and get the result.

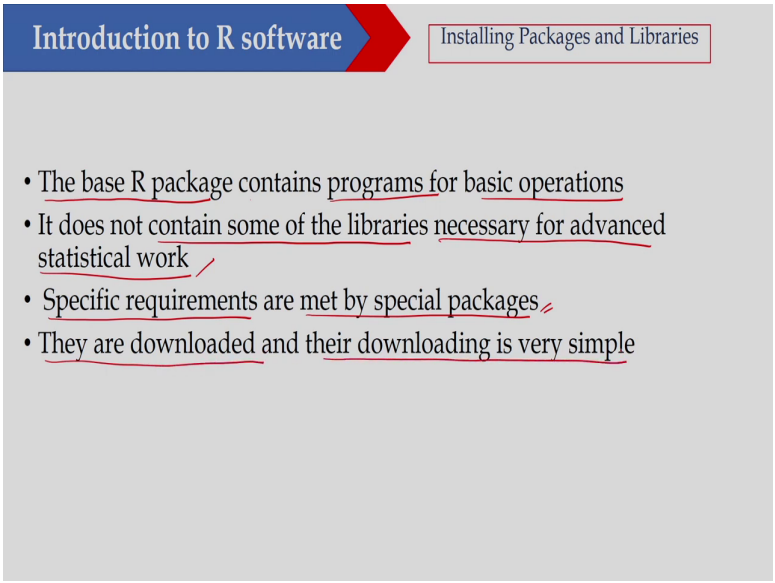
(Refer Slide Time: 07:18)



But what I can do basically? I can also use the R editor. So, how to open that editor? You go to file and open the R editor. Here I have open the R editor and the same calculation same calculation that A equal to 1, B equal to 2 and C equal to A plus B I have written here and once I am executing these 3 lines, so, I will get the value of C and which is equal to 3. So, I am getting the value of C which is equal to 3 in this case.

So, I can write a complicated program here also in using R editor, but R editor is not that efficient ok. So, R editor is not that efficient. So, therefore, I can use some other software. So, that also I will discuss like R Studio I can use. So, that I will discuss here.

(Refer Slide Time: 08:14)



Now, the R base package. So, whatever we have installed, so, this is the base package contains programs of basic operation. So, it contains the program of basic operation. So, it does not contain some of the libraries necessary for advanced statistical work. So, when you are installing R base package, so, it will not contain the library some of the libraries, so, which libraries are required for advanced computing statistical work.

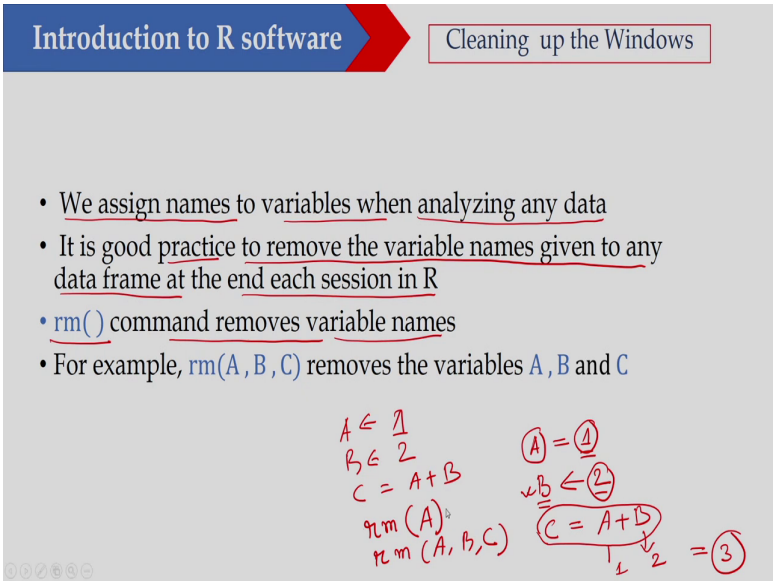
So, we can download that in the end of this particular your lecture. I will also show you how you can download the libraries ok develop for different purposes. So, you can download that

libraries and you can install that one and then basically you can use that one. So that I will show you, but whatever basic R package so, that only contains program for basic operation.

Just like you can write a simple code using different conditional loop, you can assign, you can write the equation, you can write the function, but you will not be able to do some other advanced work. So, that for that you can download the libraries. Specific requirements are met by specific packages. So, as I said that they are there are different packages. So, I will also show you suppose GA is one package. So, where genetic algorithm is there. So, in this class I will show you how you can download GA package ok. So, that is for specific purpose.

So, specific requirements are made by specific packages. They are downloaded and they are downloading is very simple. So, easily you can download that software ok. So, or that package ok. So, you can download that one.

(Refer Slide Time: 10:12)

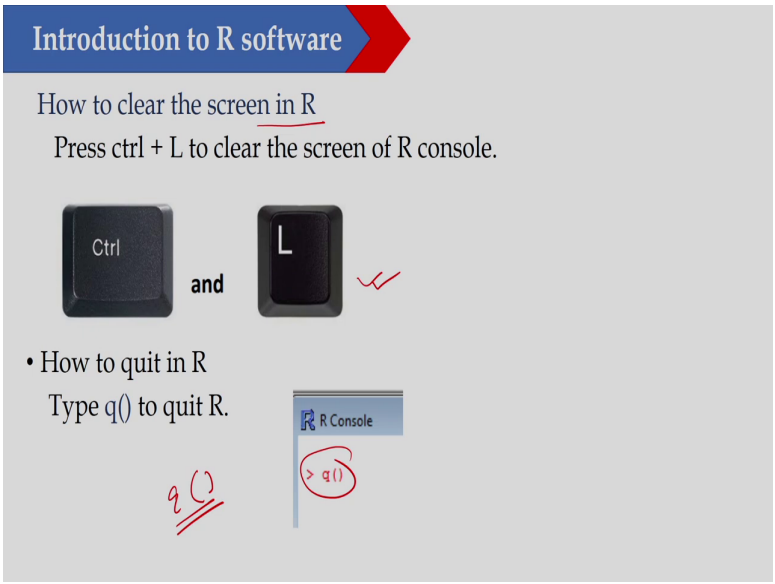


We assign names to variable when analyzing any data, ok. So, I will show you. So, suppose in the I have shown you that I would like to put that A equal to 1, so; that means, A is a variable and I am assigning a value equal to 1, B is basically 2. So, I am assigning a value 2 to the variable B and similarly I am calculating C equal to A plus B. So, here what I am doing?.

I am assigning a value 1 to A, similarly I am assigning a value 2 to the variable B then I am executing this particular line that is C equal to A plus B. So, therefore, the value of C will be equal to 3 in this case because A equal to 1 and B equal to 2. So, you should get 3 ok, it is good practice. So, when you are assigning value to a variable, so, it is a good practice to remove the variable names given to any data frame at the end of each session ok. So, end of each session you should remove that one. So, you can remove that one by using rm command this command will remove the variable. So, what I can do?.

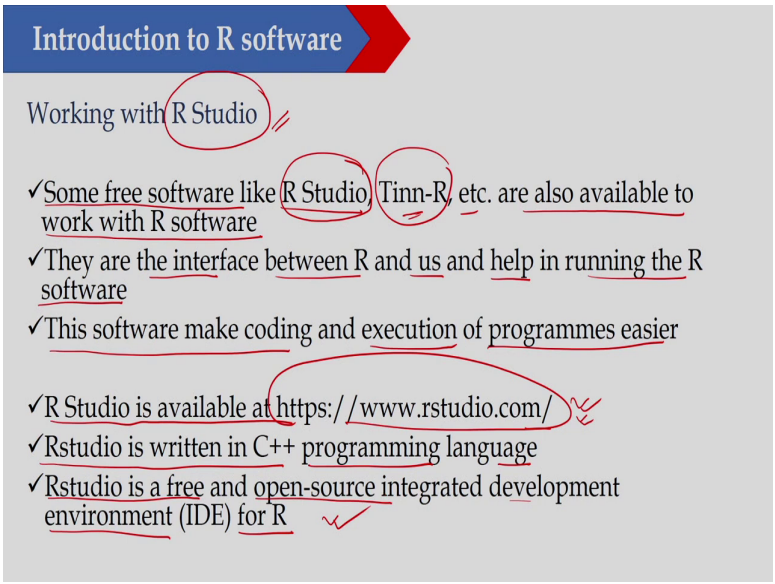
Suppose I have assign A equal to 1 ok, B equal to ok 2, then C equal to A plus B ok. So, I will get the value C equal to 3. Now, I would like to remove the variable A. So, what I will do? I will write rm A. So, it will remove the variable A. Similarly, if I write rm A comma B comma C, so, it will remove all these 3 variables.

(Refer Slide Time: 12:16)



Now, how to clear the screen in R? So, you can put control L to clean the screen. So, that I can do control L and similarly I can quit R by writing q ok. So, by writing q. So, by writing q I can quit R. So, you can close the R program by writing q.

(Refer Slide Time: 12:43)



Now, as I said the basic R whatever we have installed, so, it has an editor. So, where you can write your program, so, you can write the line and all the lines you can execute, but it is not that efficient. So, what we can do? We can also download R Studio. So, this is also a free software, you can directly download that one from the net.

So, there are some free software like R Studio as I said. So, I will download R Studio, but there is another one Tinn-R. So, that also you can download etcetera are available to work with R software ok. So, this is a platform actually. So, this is actually interacting between R and basically us.

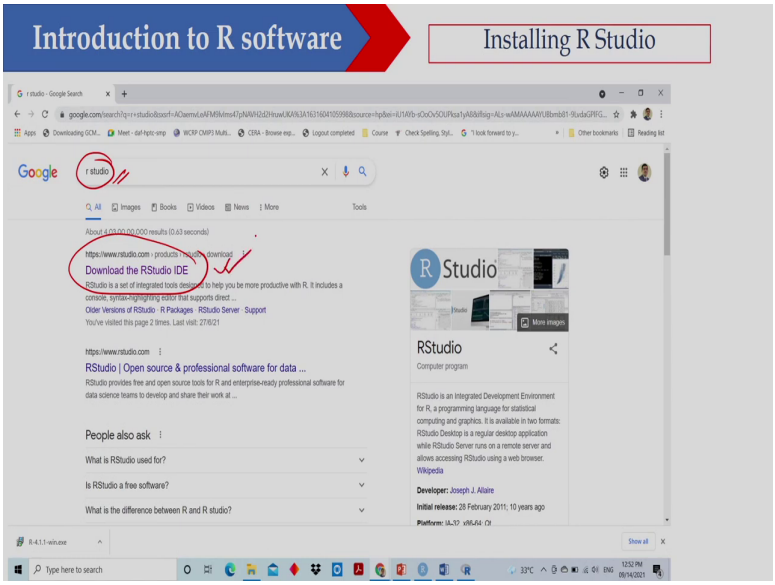
So, this is an interface and this interface can also be used and that is little bit more efficient than the R editor. So, they are interface between R and us and help in running the R software.

So, it is basically an interface and it will help you in running the R software. So, this software makes coding and execution of the programming easier.

So, you can as I said you can use the R editor, but you can also use R Studio and that is actually is really it is efficient in making the coding in executing and displaying the results. So, even graphical display also is better than R software ok. So, R Studio is available at this particular link.

So, you can go to that one that is www rstudio dot com. So, from here you can download that one. So, RStudio is written in C plus plus programming language. RStudio is a free software as I said and open source integrated development environment for R ok. So, this is a free software and you can download it and you can use it.

(Refer Slide Time: 14:49)

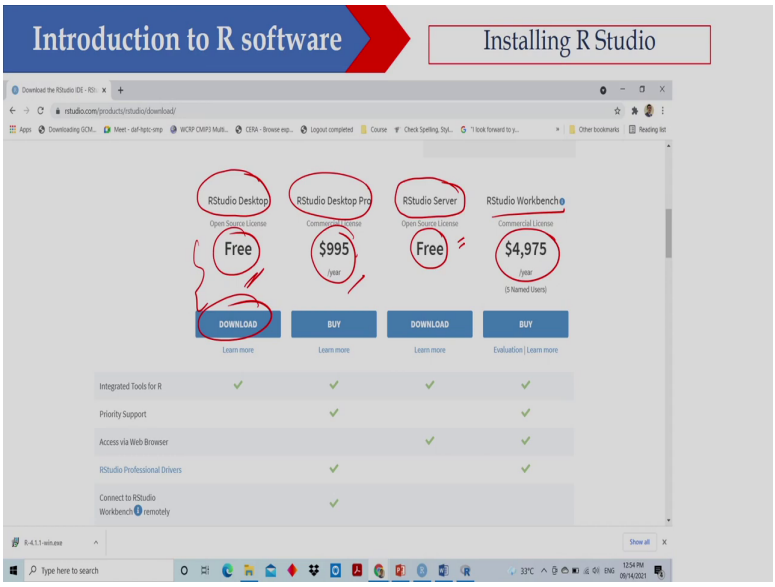


So, what you have to do? You again search in Google that is R Studio. Now, I am getting this link, that is download the RStudio ok. So, you please click in this particular link.

(Refer Slide Time: 15:06)

So, once you are clicking then you will get this particular page where download the RStudio, ok. So, download the RStudio.

(Refer Slide Time: 15:14)



So, if you scroll down then you will get this one. So, there are different version. So, this is RStudio Desktop and this is Desktop Pro then you can also download the Server one if you want to install in server and there is another RStudio Workbench ok. So, now this is free, this is free and this is also free, but if you want to purchase the Pro, so, it has some other facilities.

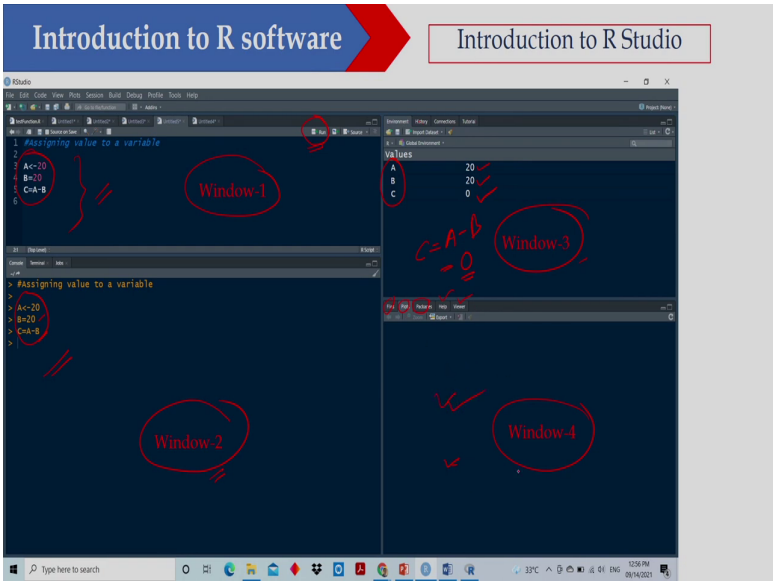
So, I do not want to use this one, but if you are suppose if you need this one then you can also you have to purchase this one, but for our work the free version is sufficient enough. So, I am not downloading the RStudio Server that is also free, but I will download the RStudio Desktop version ok. So, this is you can download.

So, you click on the download button and you can download that one and then you can install this, installation is also not difficult you just click on the exe file and then follow the

instruction given ok. So, you can install it very easily. So, here you should note one thing that so, we are working on R Studio ok. So, what you will do?.

You have to what you have to install R engine first ok the R engine and then you have to install R Studio, but when you are working. So, you will work on R Studio, but R engine should be there on your computer. So, therefore, first you have to install the R engine. So, as I have shown you and then you install the R Studio and then we will work on R Studio.

(Refer Slide Time: 17:03)



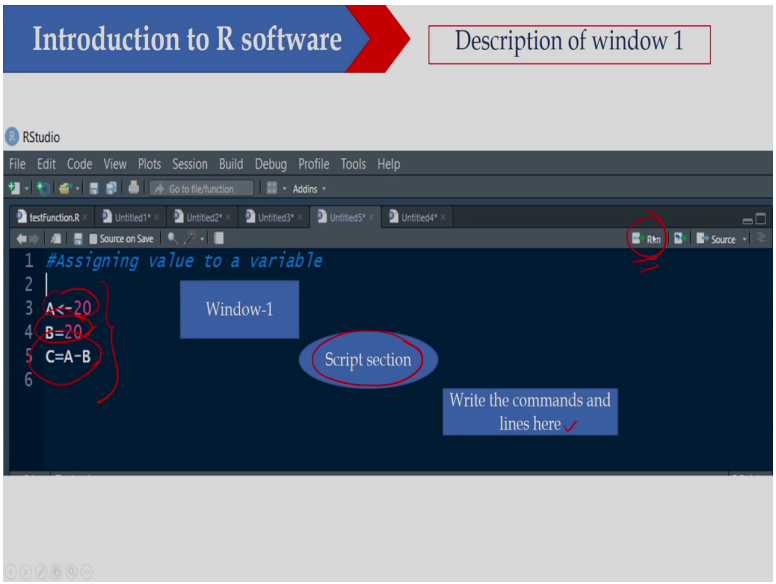
Now, once you are opening R Studio, so, you are getting this particular interface ok. So, we have here the Window 1, then Window 2, Window 3 and Window 4. So, we have total here 4 windows ok. Now, in Window 1 this is basically the editor. So, what I can do? I can write my coding or I can do the coding in this particular window, ok.

So, this is the editor and here I can do the or I can write this, but I can run this particular line using the Run button ok. So, using the Run button, so, you select that one and just by clicking at Run, so, these lines will be executed ok. So, once you are executed, so, this is the Console. So, here you will see that this lines has been executed ok. So, that A equal to 20, B equal to 20, A minus B. So, that has been executed and you are getting at Window 2.

And Window 3, so, you can under environment you can see all the variables here. So, in this case I have 3 variables here that is A, B, C. So, you can see that A equal to 20 I have assigned 20, B equal to 20 I have assigned 20 and C equal to; C equal to A minus B. So, therefore, C value is equal to 0. So, C is 0. Similarly, I can also see the history. So, if you click on History button, so, you can see the history the connection then tutorial. So, that also you can do it.

Now, if you look at the last one, so, last one you have actually, so, you can see the files, you can see the plots, you can see the packages. So, you can download the packages using this particular window and Help is there and Viewer is there ok, so, that you can see. Suppose if I plot something if I plot a curve, so, you can you will get your plot in this particular window.

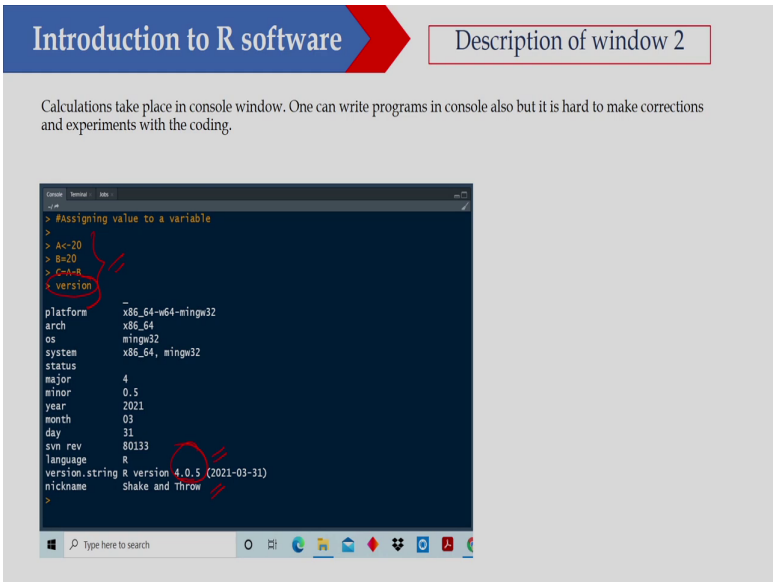
(Refer Slide Time: 19:11)



So, here this is the Window 1. So, as I said in Window 1, this is the script section. So, I can write my script here. So, write the commands and lines here. So, suppose I have written A equal to 20. So, it will not execute, so, I then B equal to 20 and C equal to A minus B. Now, if I want to execute this particular line, so, what I will do?.

I have to select this first this line and then I have to use the Run ok. Similarly, then if I want to execute this particular line then again I will select that one and I will click Run and similarly if I execute the third line then I will select that one and I can execute that particular line using the Run button ok. Or if I want if I select all of them then I can and then I use Run button then it will all these three lines will be executed ok. So, this is Windows 1.

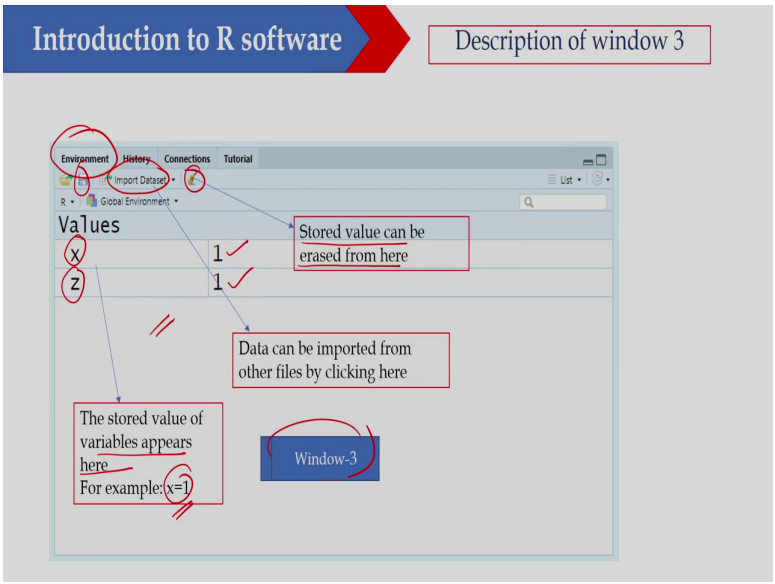
(Refer Slide Time: 20:15)



Now, Windows 2 as I said, so, whatever you are executing, so, this will be there in console ok. Then whatever you are executing the result you can see in this particular window. Suppose I would like to see what version we have actually. So, I am writing the version and it is showing the version ok.

So, this is the version I have basically. So, you can see. So, whatever R version is there that is 4.0.5 and this is the version I have your downloaded ok or this is the version I have installed here. So, whatever you are executing. So, you can you will get your result in this particular window.

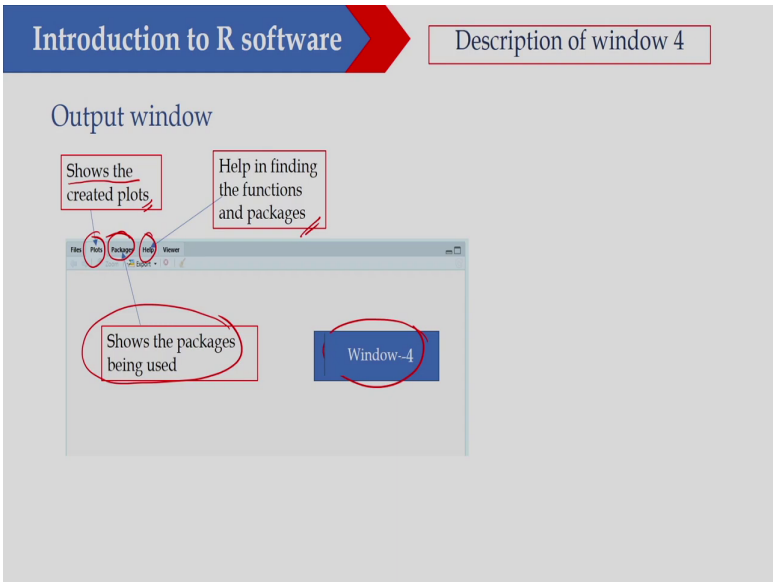
(Refer Slide Time: 20:59)



Then in the Window 3, so, this is under environment. So, under environment you can see the variables. Suppose you have x and z variable, x equal to 1 and z equal to 1, so, that you can see. The store value of variables appeared here for example, x equal to 1. So, whatever variable is there you can see in the on the environment window. And similarly, you can import database using that one. Suppose, I would like to import some data to this particular environment, so, I can import that one.

And similarly if I want to erase all these data if I want to delete all these data, so, whatever store value can be erased from here, so, by clicking at this particular button, so, I can actually delete all these variables. So, I can clean the environment ok using this one. Similarly, I can also save the environment ok that also I can do and there are some other option. So, I am not discussing here, but you can also explore those option.

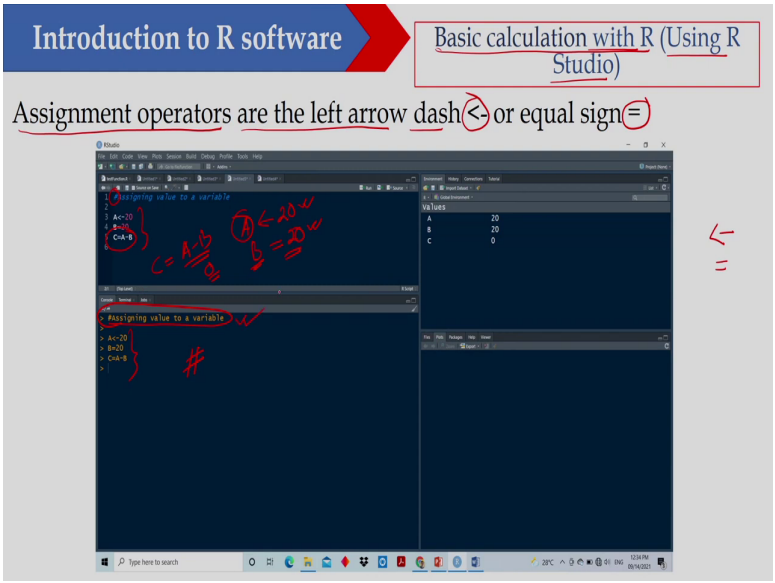
(Refer Slide Time: 22:09)



Now, the output window that is the Window 4, ok. So, as I said in the output window, so, it shows the created plot. Suppose if you have created some plots. So, you can see here similarly if you want to see what are the packages you are using, so, you can go to Package and then Help button will give you if you want if you need any help.

So, you can go to Help button and you can search for help. So, that also you can do. And you can install package using this particular your window also. So, I would like to install suppose a particular software that also you can install using this particular window.

(Refer Slide Time: 22:55)



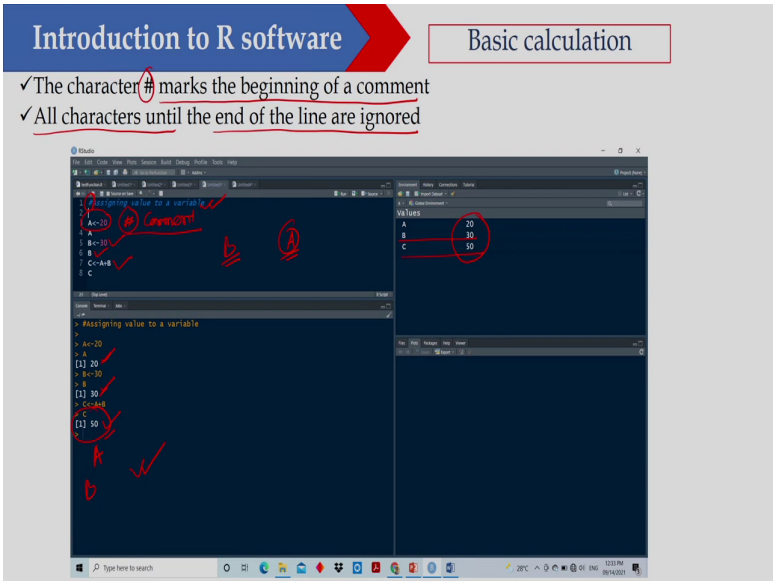
The basic calculation with R so, using R Studio. As I said I will work mainly on R Studio, so, but I have downloaded the R engine and then R Studio, but I am working on R Studio. I am not opening the R engine, but I am opening the R Studio. Now, assignment of operator are left arrow and dash.

So, I can assign a value to a variable using this particular left arrow and dash. So, this is left arrow and dash or I can also use the equality sign both are acceptable here. So, you can write suppose here I have written that A is equal to 20 or I can write B equal to 20, ok. So, both will assign; that means, A is a variable and the value of that variable is 20, B is a variable and value of B is 20 and C is a line where I am doing some operation that C equal to A minus B.

So, you will be getting the value equal to 0, ok. So, once I am executing this, so, I am getting this one. Now, if you are writing a hash in front of any line suppose here I am writing hash,

so, this is giving you a comment that particular line will not be executed. So, therefore, this has not been executed ok this line, but it is showing here and this is a comment you are getting.

(Refer Slide Time: 24:34)



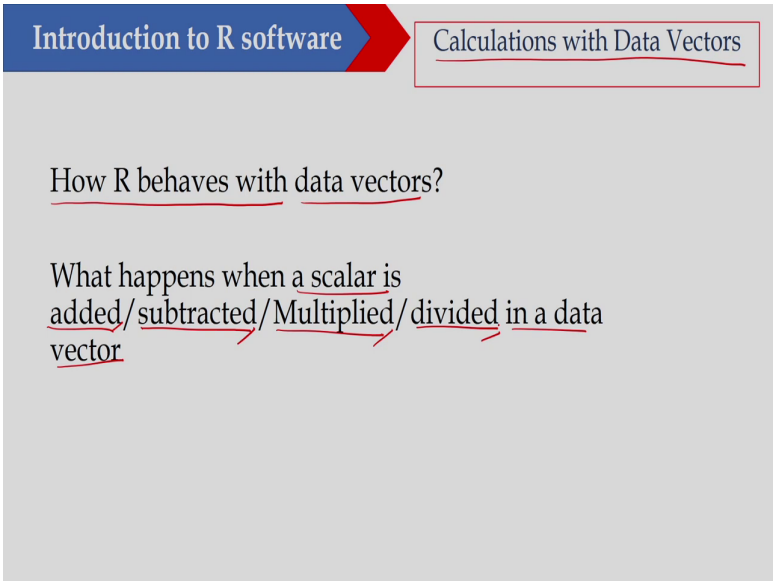
So, as I said the character has marks the beginning of a comment; that means, if I am if I put any hash, so, this line will not be executed and it is basically you are writing some comment. All characters until the end of the line will be ignored. So, all will be ignored. Suppose I can also write a hash here and I can write a comment here ok.

So, in that case this portion will be executed, but this portion will not be executed because I am putting this hash here ok. So, now, in this case, so, what I am doing? I am assigning A equal to 20. So, you can see that A equal to 20 B equal to 30. So, B equal to 30 C equal to a plus B that means, I should get 50. So, I am getting C value is equal to 50. So, here what is

happening? If I write A and I execute this one. So, it will display the value of A in this particular window.

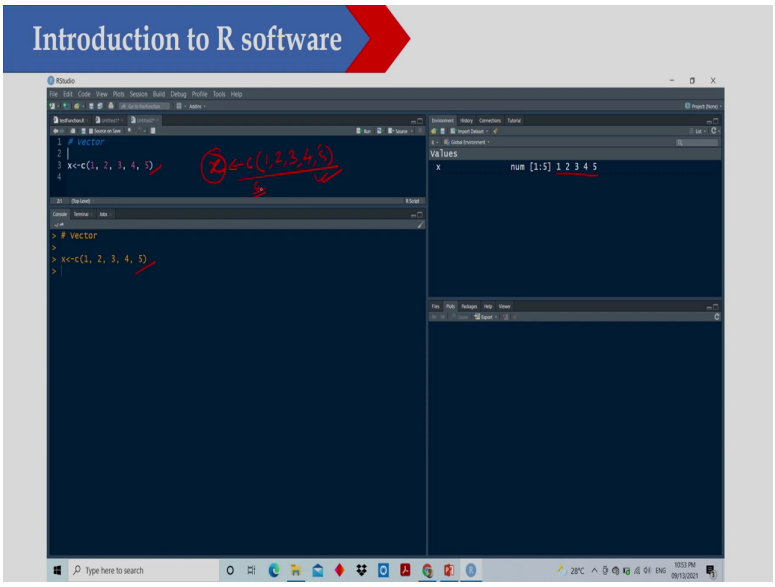
Similarly, if I write B here and then if I execute then I will get the value of B. So, I am getting the value of B is 30, value of A is 20 and value of C is equal to 50 and you can also see here. So, on the environment window you can see that A equal to 20, B equal to 30, C equal to 50. So, you can get that one on the environment window also.

(Refer Slide Time: 26:06)



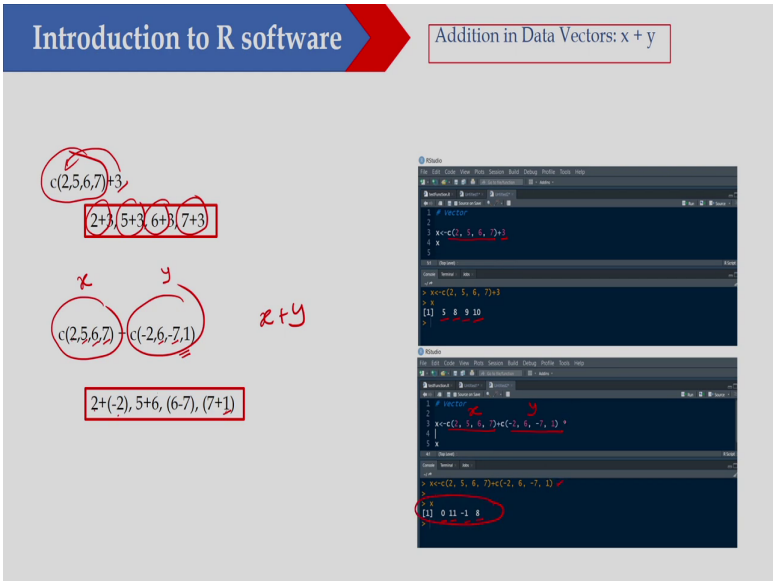
Now, let us see how R behaves with data vector ok. So, calculation with data vector. So, what happen? When a scalar is added or subtracted or multiplied or divided in a data vector ok. So, if I add a scalar to a data vector or if I subtract a scalar from a data vector if I multiply, if I divide it, so, what will happen? So, let us see that one.

(Refer Slide Time: 26:41)



Now, a vector is defined using this command, so; that means, how I will define that using C. So, C is suppose I am writing here C equal to 1, 2, 3, 4, 5. So, using this one and I am assigning that one to x. So, x is a vector actually here and the values are 1, 2, 3, 4, 5s ok. So, I am defining here. So, if I execute that one, so, I am getting this one and if I see what is x? x equal to so, it is 1 to 5; 1, 2, 3, 4, 5 ok. So, I am defining if I can define a vector using C ok. So, this is I am defining.

(Refer Slide Time: 27:30)



Now, let us see what will happen that C is a vector ok. So, this is a vector that is 2, 5, 6, 7; 2, 5, 6, 7 and I am adding another scalar that is 3 with this particular vector. So, what I will do basically? So, I am writing here this is C ok and C equal to 2, 5, 6, 7 and this is a vector and with that I am adding 3. So, what will happen?.

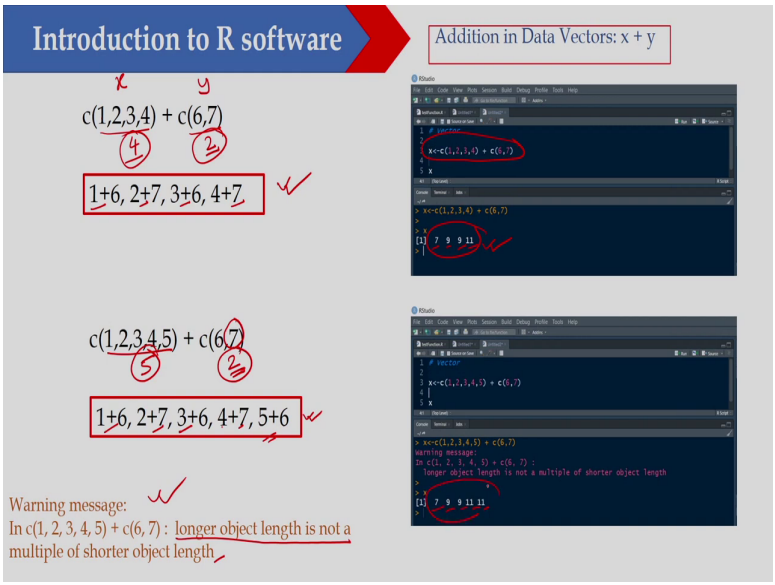
3 will be added to all these values; that means I will be getting 2 plus 3, 5 plus 3, 6 plus 3, 7 plus 3; that means, 3 will be added to this particular vector. So, all the elements of the vector. So, therefore, I will be getting 2 plus 3 that is 5; 5 plus 3 that is 8; 6 plus 3 that is 9 and 7 plus 3 that is 10 ok.

So, what I am doing here? So, I am adding this is a vector ok. Suppose this is x and this is another vector this is y. So, what I am doing? x plus y and you can see that how many elements are here? We have 4 elements and we have 4 elements here. So, in that case what

will happen? The corresponding element will be added; that means, first element of this x and first element of y will be added that is 2 plus minus 2.

Similarly, second element of x and the second element of y that is 5 plus 6, similarly third element of x and third element of y that is 6 minus 7 and the fourth element of x and fourth element of y will be added. So, you can see here. So, I have added 2 vector. This is the first vector, this is x and this is the second vector this is y. And what I am getting? That is 2 minus 2. So, I am getting 0, 5 plus 6 this is 11, 6 minus 7 minus 1 7 plus 1 is 8. So, I am getting this particular result. So, if I execute this particular line.

(Refer Slide Time: 29:48)



Now, if I add these 2 vector, so, this is x and this is y. Now, we just see the difference here. So, we have total 4 elements and here we have 2 elements ok. So, therefore, what will happen? So, it will add the first element. So, what will happen? The 6 will be added to 1 and

7 will be added to 2 ok. So, 6 is added to the first element, 7 is added to the second element then it will repeat again with third element it will the 6 will be added and fourth element the 7 will be added.

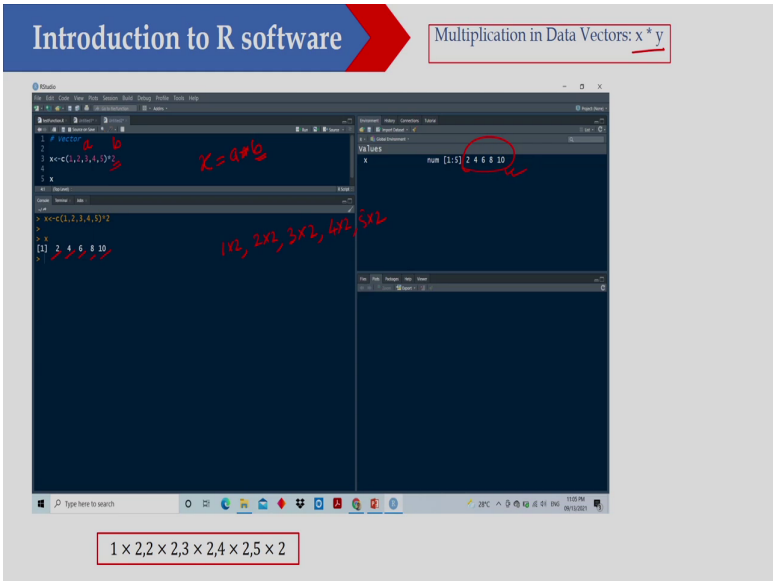
So, here what is the; we have 4 here and we have 2 here. So, I can also show you here. So, this is I have done in R Studio and you can see the first element is 7, second element is 9, third element is 9 and fourth element is 11 ok, so; that means, 1 plus 6 that is 7 2 plus 7 that is 9 3 plus 6 that is 9 again 4 plus 7 that is 11. So, I am doing that one. Now, you just see here what is happening, we have total 5 and this is 2 ok.

So, in this case what will happen? The one will be 6 will be added to 1, 7 will be added to 2 then it will repeat. 3 the 6 will be added to 3 and 7 will be added to 4 and 5 and on the last element 6 will be added to 5, but then 7 is again. So, there is there should be another element. So, in that case it will execute that one, but it will give a warning ok.

So, what is the warning here? So, warning here is that longer object length is not a multiple of shorter object length. So, it is not a multiple. In this case the longer one is a multiple of shorter one ok. So, therefore, there is no error, but in this case the longer one is 5 and shorter one is 2. So, it is not a multiple.

So, therefore, there will be an error message ok. So, otherwise this will also calculate ok, you can see. So, I am getting 7, 9, 9, 11, 11. So, I am getting this particular results.

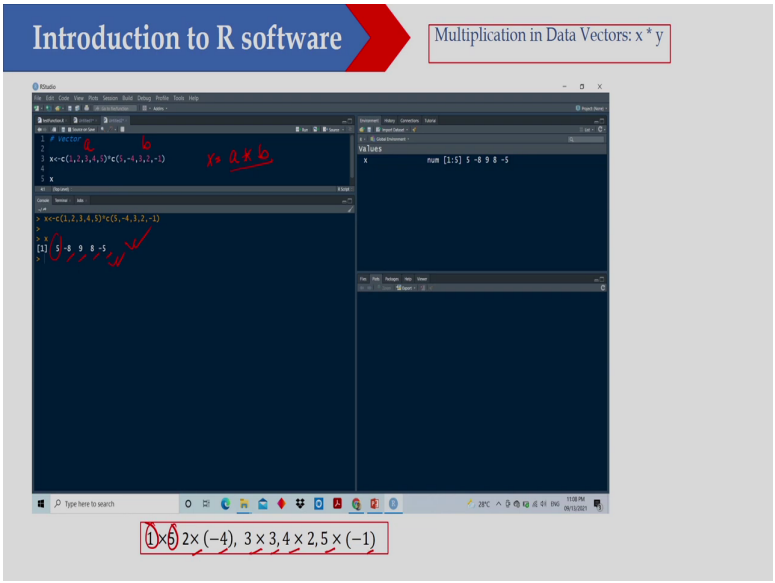
(Refer Slide Time: 32:16)



Now, if I multiply 2 vector ok, so, what I am doing here? I am multiplying a vector with scalar. So, here this is suppose a is a vector and b is a scalar and x equal to a star b ok. So, now what will happen? b is a scalar here. So, what will happen? All the elements will be multiplied by 2. So, therefore, what I am getting? This will be 1 into 2, then 2 into 2, 3 into 2, then 4 into 2 and then 5 into 2 ok.

So, how much you are getting? That 2 1 into 2. So, I am getting 2 then 2 into 2 I am getting 4 then 3 into 2 I am getting 6 then 4 into 2 I am getting 8 and 5 into 2 I am getting 10. So, all the elements will be multiplied by this particular scalar. So, results are also shown here and you can see that is 2, 4, 6, 8 and 10 I am getting this one. So, here I have multiplied a vector by a scalar.

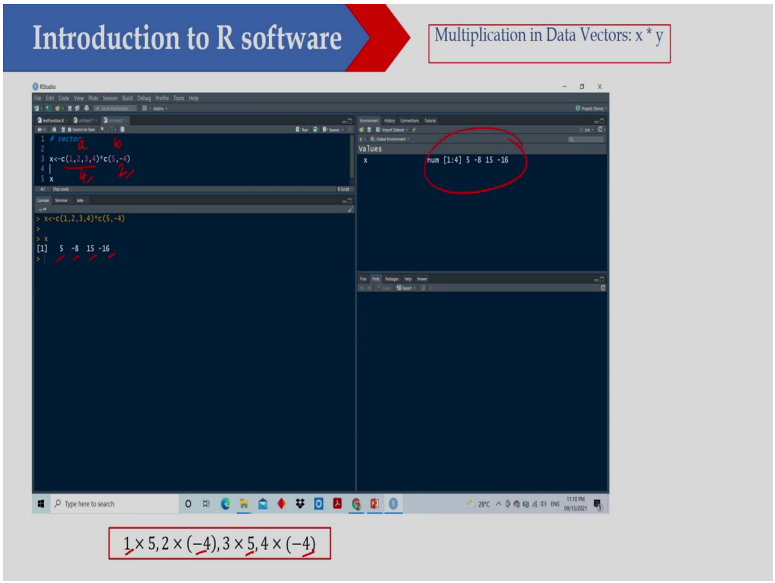
(Refer Slide Time: 33:34)



Now, you just see here in this case I have multiplied 2 factor of equal length. So, here it is a here it is b. So, a is also a vector and b is also a vector ok. So, I am getting the value of x here. So, here if you are multiplying 2 vector ok, so, in that case what you will get? So, the corresponding element will be multiplied. So, here what you are getting?.

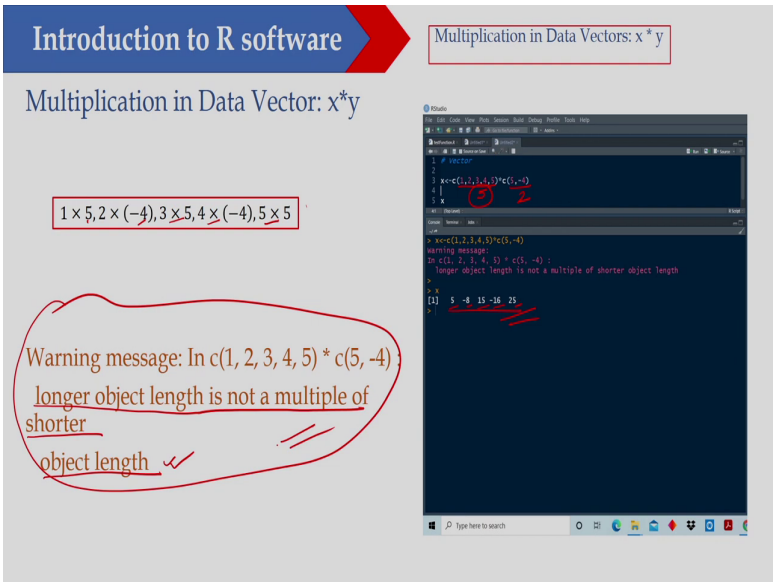
One is the first element of vector a and 5 is the first element of vector b; so, 1 into 5 2 into minus 4 3 into 3 4 into 2 and 5 into 1. So, what is the result here? That 1, then 8, 9, 8, 5 ok, so; that means, if you are using a star b then this is a multiplication of 2 vector. So, you are getting this particular results. So, I hope this is clear to you.

(Refer Slide Time: 34:43)



Now, so, here what I am doing? Here a is a vector and b is also vector, but length of the vector are not equal. So, length is 4 here and this is 2 here, but this is the longer one and this is the shorter one, but longer one is a multiple of the shorter one. So, what I will get? So, in this case the 5 will be multiplied to 1 then minus 4 then this will repeat again this is 5 and this is minus 4 ok. So, results are 5 minus 8 15 minus 16 you are getting and results are also stored here. So, there is no issue ok.

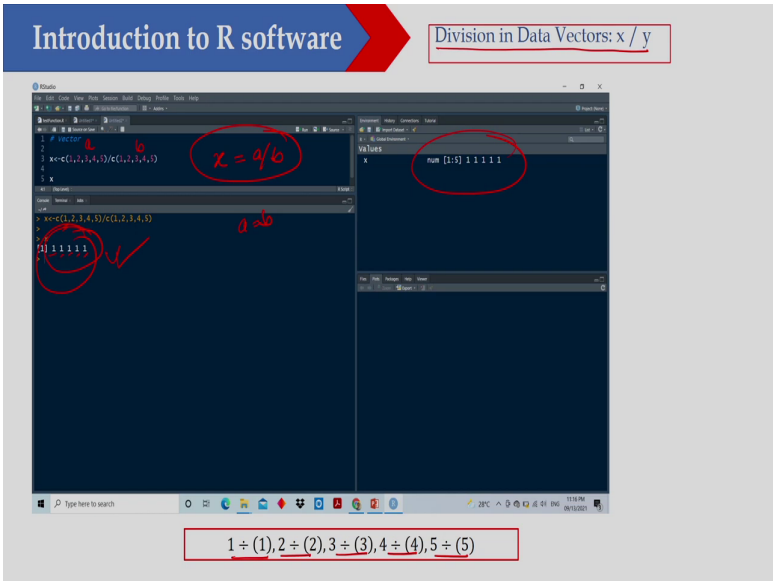
(Refer Slide Time: 35:28)



So, here what we are doing? So, here length of the vector is 5 and length of the vector is 2. So, in this case the longer object length is not multiple of the shorter object length. So, in this case the longer object length is not multiple of shorter object length. So, therefore, I will get the result. So, this is the result I am getting, but I am also getting a warning ok. So, this is the warning you will get.

So, here what you are getting? 1 into 5, then 2 into minus 4, then 3 into 5, 4 into minus 4, 5 into 5 and basically so, it is a as the length is not multiple of the shorter one. So, I am getting a warning, but I am getting; I am getting the results here that is 5, minus 8, 15, minus 16, plus 25.

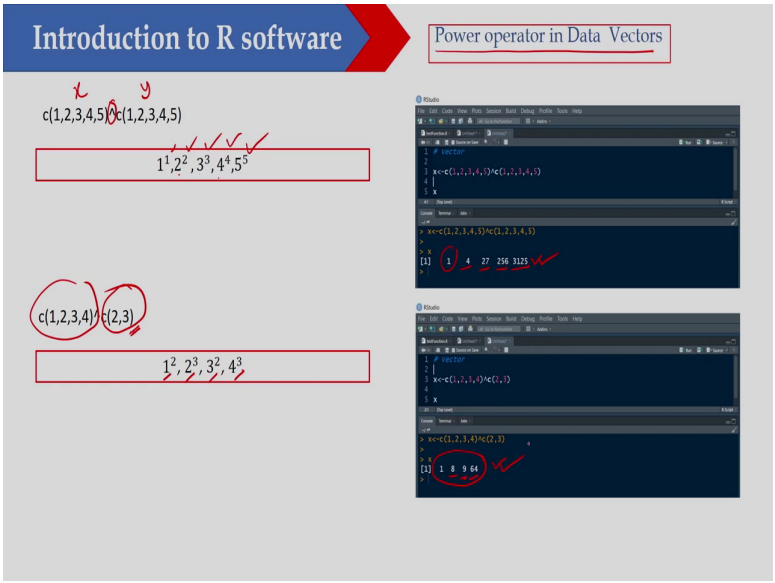
(Refer Slide Time: 36:28)



So, here I am dividing x by y; that means division in data vector. So, that is x is a vector and y is also a vector. So, x by y I am doing. So, this is suppose a and this is b. So, what I am calculating? x equal to a by b. So, if you are doing that one, so, in this case what will happen? The corresponding element will be divided ok. So that means, what will happen?.

1 divide by 1, then 2 divided by 2, 3 divide by 3, 4 divide by 4, 5 divided by 5. So, this is the same vector that here it is a equal to b. So, therefore, the result is 1 1 1 1; 1 1 1 1 1 1. So, this is the result I am getting ok. So, this is the result I am getting and the corresponding element will be divided ok.

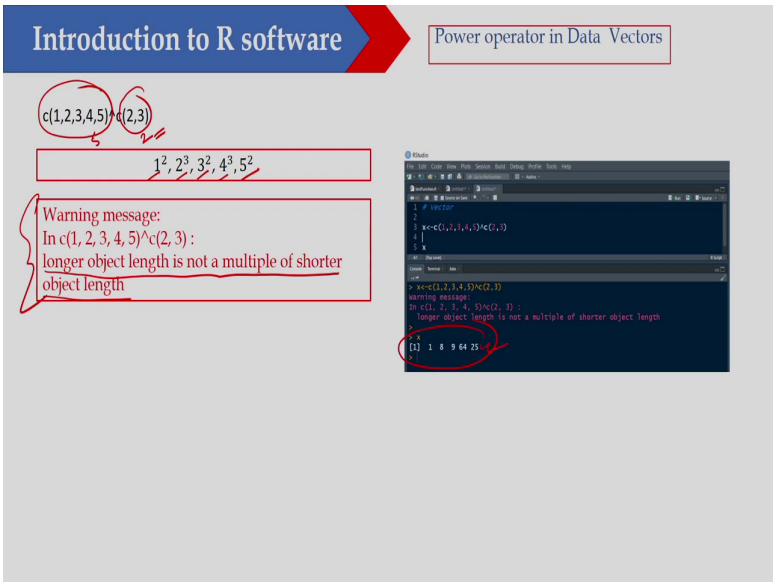
(Refer Slide Time: 37:24)



Now, if I want to calculate power, so, power operator in data vector. So, I have taken a vector. Suppose this is x and this is y ok. Now, what you will do basically? So, if you are using this power, so, it will be 1 to the power 1, 2 to the power 2, 3 to the power 3, 4 to the power 4, 5 to the power 5. So, what you are getting? You are getting 1, 4, 27, 256 and 3125 ok, so that means, 5 to the power 5 is 3125.

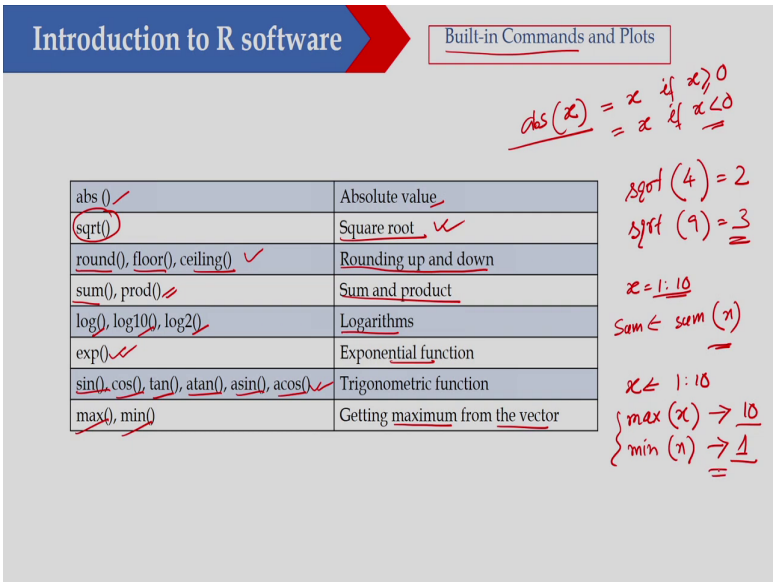
Now, if I am writing, so, here I am writing this is one vector and this is another vector. So, here the longer one is multiple of the shorter one. So, what I will get? 1 to the power 2, 2 to the power 3, 3 to the power 2 and 4 to the power 3. So, I am getting that one. So, this is the result I am getting here that is 1, 8 and this is 3 square 9 and 4 to the power 3 64.

(Refer Slide Time: 38:37)



Similarly, if the larger one is not multiple of the shorter one, so, here the length is 5 and this is 2. So, in this case again you will get an error, but it will be calculated like this. So, you will get 1 square, 2 cube, then 3 square, 4 cube, then 5 square, but you will get a warning message that the longer object length is not multiple of the shorter object length, but you will get the result that is 1, 8, 9, 64 and 25 ok. So, you will get the result in this case, but there is a there will be a warning.

(Refer Slide Time: 39:19)



Now, in R, so, there are some built in command ok. So, that commands are absolute, suppose so, what you can do? You can calculate the absolute values. Suppose x is a variable and I would like to calculate the absolute value of x. So, what I will do? I will write ab as x ok. So, it will give you. What it will give? It will give x, if x is positive; it will also give x, if x is negative ok, it will give negative. So, it will get.

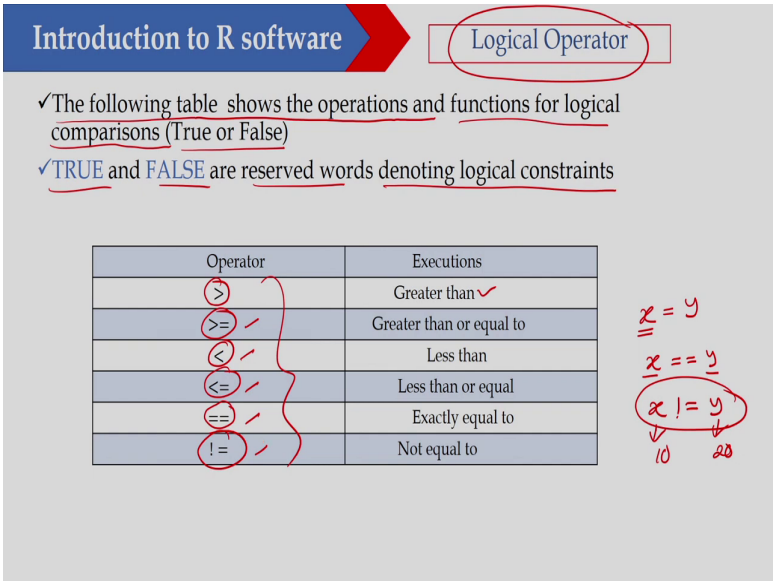
So, similarly I can calculate square root ok the square root I can calculate. Suppose if I write sqrt and I am writing 4. So, I will be getting 2 ok, sqrt this is 9. So, I will getting 3 ok. So, I can find out the square root of any variable using that one. Similarly, I can use round, floor, ceiling ok. So, rounding up and down basically. So, you can use that one then if you want to use sum and product ok.

So, suppose x equal to 1 to 10 ok and I can write that if I want to sum ok, sum is equal to that is sum, this is x. So, it will add all these at the vector ok. So, you can calculate, similarly you can also calculate the product that I that you can do product of two. Similarly, I can also calculate logarithmic.

So, log, then log10, log2. So, that you can do. Similarly, exponential function that you can write exp then trigonometric function that sin x, cos x, tan x, atan, asin, acos. So, you can use that one and max and min function is also there to get the maximum from a vector. Suppose, if x is a vector which is 1 to 10 then if I write max x. So, what I will get? I will get the value 10.

If I write min x, so, what I will get? I will get 1. Is not it? So, its a vector x is a vector which will have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. So, maximum is 10 and minimum is 1. So, I can use max and min function to calculate maximum value or minimum value of the vector ok. So, I can use this one.

(Refer Slide Time: 42:21)

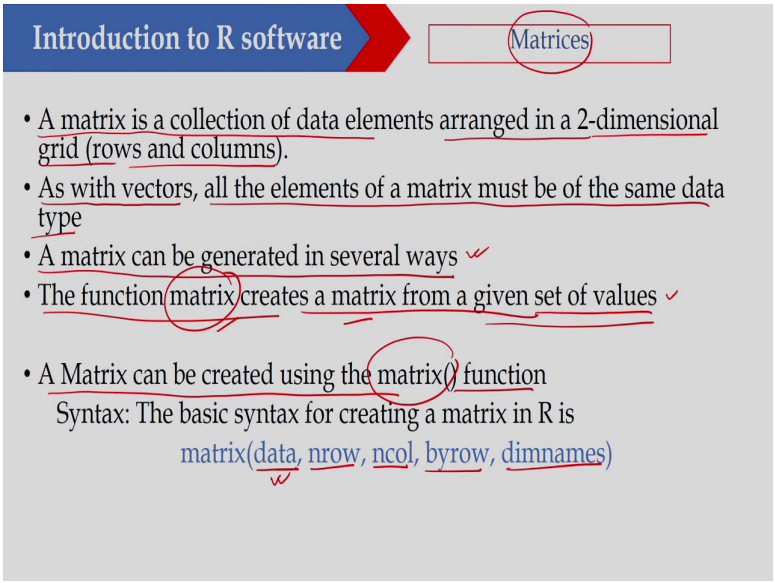


So, there are the logical operators. So, in programming I also need the logical operator the following table shows the operation and function for logical comparisons TRUE or FALSE. TRUE and FALSE are reserved word denoting logical constraints ok. So, what is this? This is greater than and this is greater and equal then this is less than this is less than equal. Then if you it is equal, suppose if I say x equal to y, so, what does it mean?.

That the value of y will be assigned to x, but if I say x equal to y, so, this is a logical statement; that means, variable x is equal to y if it is equal then it will return TRUE and if it is not equal it will return FALSE ok. And not equal. So, not equal; that means, if I write not equal to y. So, if it is not equal suppose x value is 10 and y value is 20. So, this is not equal. So, this operator will give you TRUE ok.

And if it is equal then it will give you FALSE ok. So, I can use this operator in my coding. So, I can use it if there is any logical operation greater than then greater than equal to this type less than less than equal then equal not equal to so that I can use this symbol.

(Refer Slide Time: 44:00)



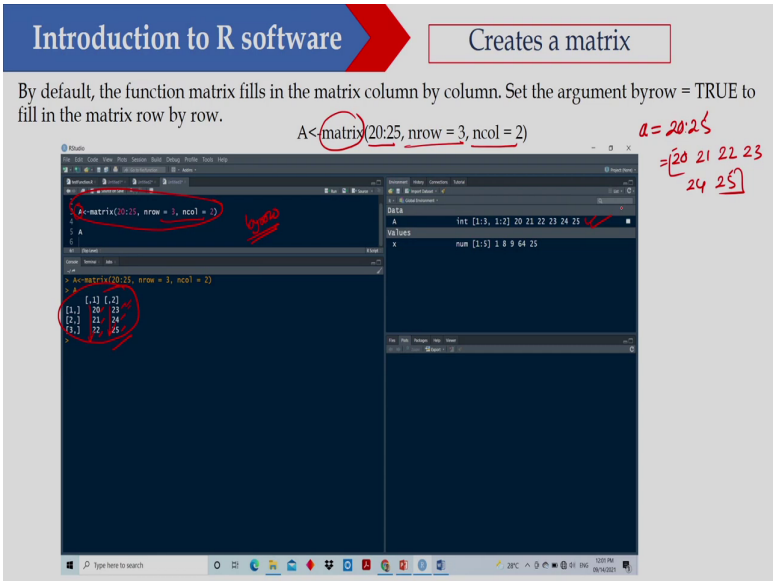
Let us see how you can define matrices in R. A matrix is a collection of data element under 2-dimensional grid column and rows ok. As with vectors, all elements of a matrix must be of same data types ok. So, it must be of same data type then a matrix can be generated several ways. So, we have different ways to generate a matrix. So, I am explaining here how you can generate a matrix. The function matrix create a matrix from a given set of values.

So, you can use the matrix function you can use the matrix function to create a matrix. So, that I will show you and if from a given values. So, I can use this matrix function to create a

matrix from a given set of values. A matrix can be created using matrix function as I said. So, this function you can use.

The argument of this function is data. So, you have to; you have to put the data here and data will put as a vector the number of rows number of column then by rows actually. So, by default it will be by column I will explain you and dimnames.

(Refer Slide Time: 45:33)



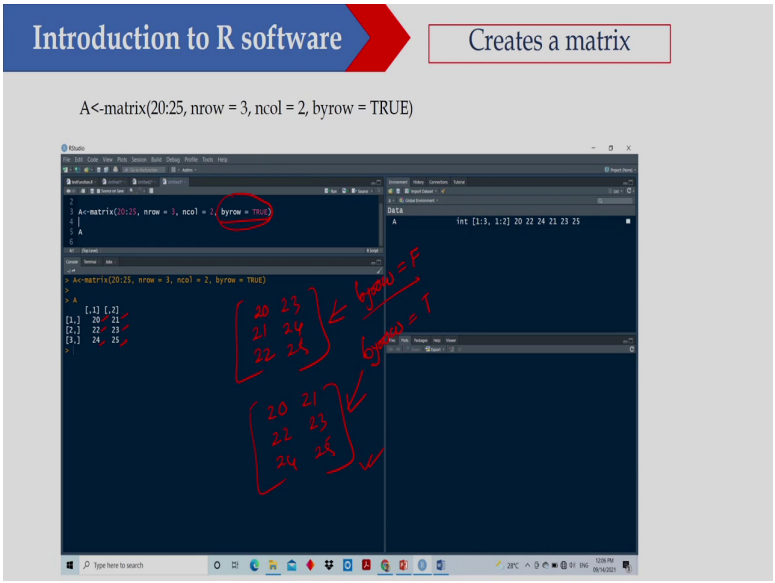
So, let us create a matrix here. So, as I said I can use this matrix function. So, here I am putting the data is a vector that is C. So, here I am putting the data suppose this is a equal to 20 to 25; that means, it will have 21, 22, 23, 24 and 25 ok. So, it will have these data and number of rows equal to 3, number of column equal to 2 ok.

So, if I execute, so, I have written this particular line here and whatever matrix I have created. So, I am assigning this matrix to the variable A. So, now, variable A is a matrix and it is dimension is 3 by 2; that means, number of row equal to 3 then number of column equal to 2 and this is the data.

So, you can see this is the matrix A here and by default it is basically by column. So, you can see that 20, 21, 22, 23, 24, 25. So, what will happen? First it will go to the 1st column and then it will go to the 2nd column; that means, the first element will be 1st row 1st column, second element will be 2nd row 1st column then third one will be 3rd row 1st column and then it will fill the 2nd column.

So, what I have done here? I have not written byrow ok, so, byrow. So, by default byrow is FALSE ok. So, by default byrow is FALSE. So, if you are not writing byrow if you are not writing byrow; that means, it will consider the byrow is equal to FALSE ok. So, I am getting this matrix and matrix is also stored here.

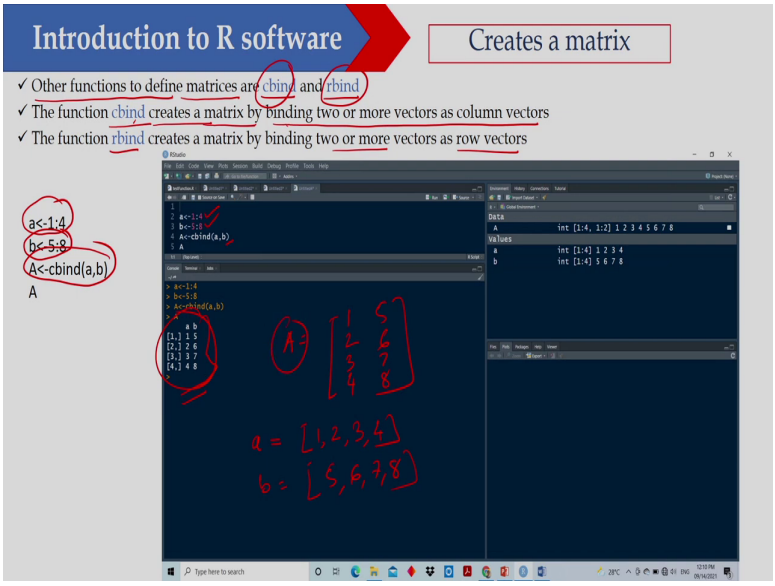
(Refer Slide Time: 47:38)



Now, is the C the I am writing byrow through basically. So, I can write I just I have just added this one just see the difference. So, here what is happening? byrow TRUE that means, it will fill byrow; that means, 20, 21, 22, 23, 24, 25. So, just look at. So, earlier one it was 20, 21, 22, 23, 24, 25. So, this was the matrix ok.

Now, this one is 20, 21, 22, 23, 24, 25, ok. So, this is; so, here what I have done? The byrow equal to TRUE ok. So, here I am not defining anything. So, if you are not defining then in that case byrow is equal to FALSE ok. So, byrow FALSE. So, it will fill the column y that 1st column then 2nd column then 3rd column something like that I hope this is clear. So, I can create a matrix.

(Refer Slide Time: 48:48)



So, there are some other function to define a matrix. So, this is cbind and rbind. So, this function also you can use to create a matrix. The function cbind create a matrix by binding two or more vector as a column vector ok. So, I will show you what cbind is doing and similarly rbind create a matrix by binding two or more vectors as a row vector.

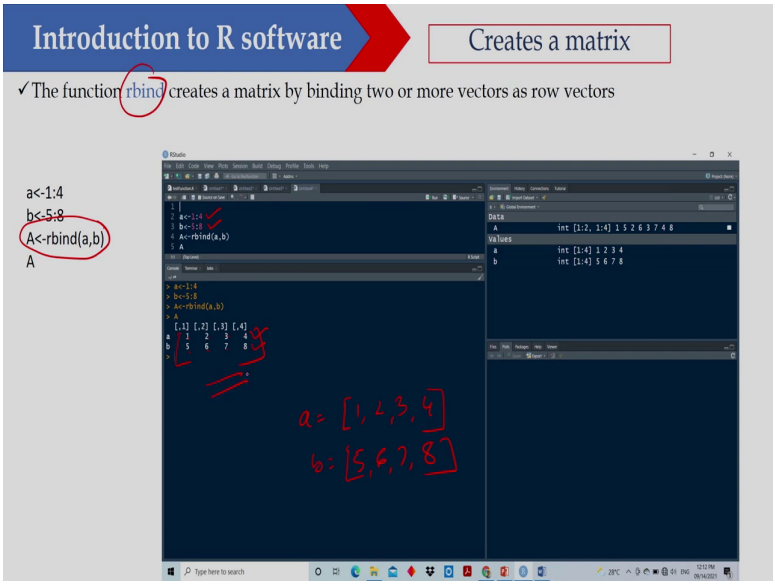
So, here you just see. So, what I am doing here? I am defining two vector; one is a one is b. So, what is a? a is 1, 2, 3, 4 ok. And what is? b is 5, 6, 7, 8 ok. Now, what I have done here? So, I am defining a, I am defining b. So, a is equal to 1, 2, 3, 4. So, this is a vector and b equal to 5, 6, 7, 8.

So, now, what I have done? I am using cbind. So that means, what it will do basically? So, it will create a matrix by binding two or more vector as a column vector. So, in this case what is

happening? It is a column vector. So, you just see you are getting that is you are getting a matrix which is A and this is a column vector.

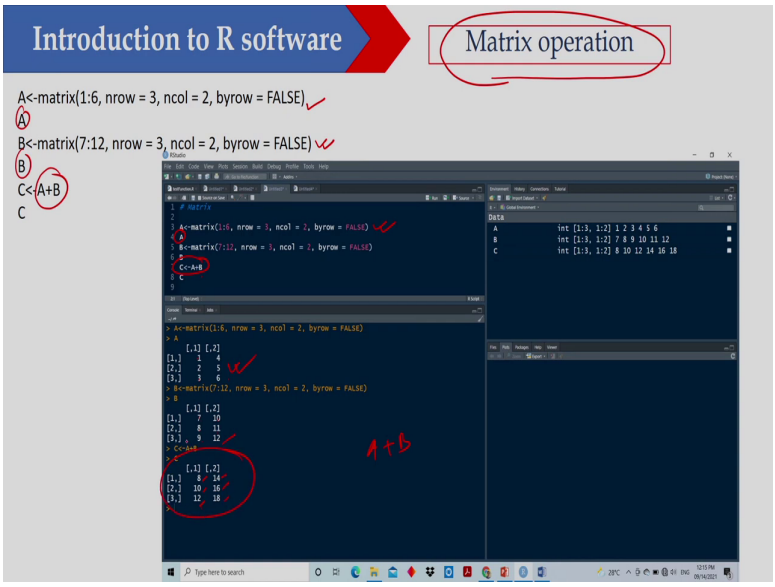
So, 1, 2, 3, 4, then 5, 6, 7, 8. So, this is a column vector. So, what I am using here? I am using the command cbind ok. So, you are getting the result here ok. So, this is the matrix I have created using cbind.

(Refer Slide Time: 50:46)



Similarly, I can also use rbind ok. So, now, the same problem ok. So, same vector I have used that a vector that is 1 to 4 then b vector that is 5 to 8. So, a equal to this is a equal to 1, 2, ok 3, 4 and similarly b equal to 5, 6, 7, 8 ok, so, 8. So, you can see now I have used rbind. So, rbind a, b. So, I am getting this particular matrix. So, we have two rows now two rows and there are four columns; 1, 2, 3, 4; this is the 1st vector; 5, 6, 7, 8 this is the 2nd vector you are getting the matrix operation. So, here I have added two matrix A plus B.

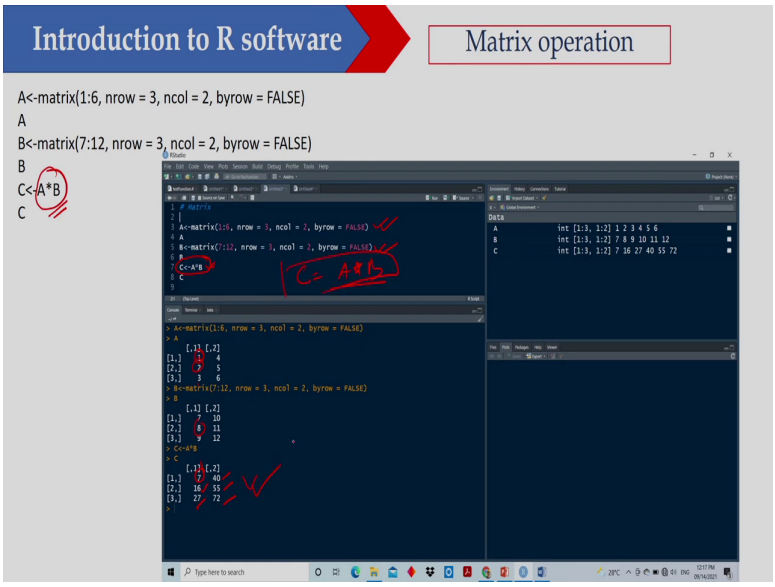
(Refer Slide Time: 51:35)



So, first line is creating one matrix that is A then this line is creating another matrix that is B and I have written A and B just to print that one. So, you can see this is the first matrix I have created then I have printed here and this is the matrix A that is 1, 2, 3, 4, 5, 6. So, this is the first matrix and this is the second one 7, 8, 9, 10, 11, 12 and then I have added these two matrix ok. So, that is A plus B.

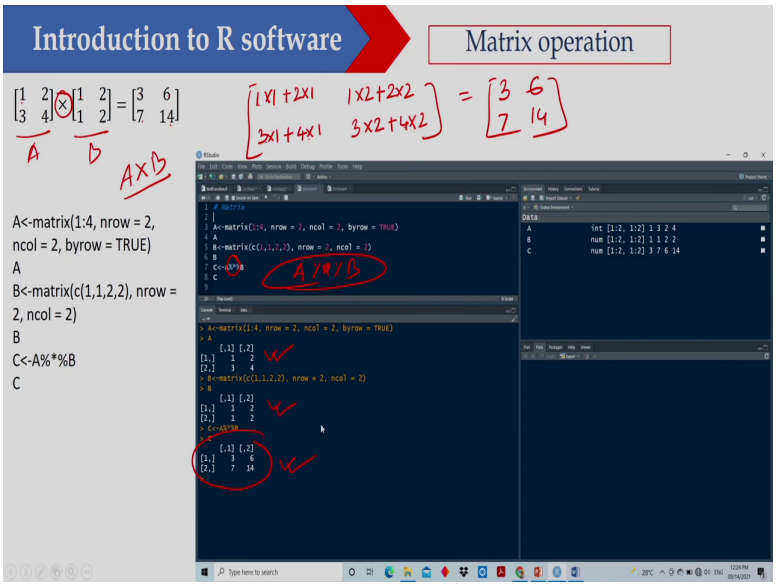
So, if you are writing A plus B A and then what you are getting? Corresponding element will be added; that means, the first one is 1 plus 7 that is 8, the second one is 2 plus 8 that is 10 then third one is 3 plus 9 that is 12, similarly 14, 16 and 18. So, I am adding two matrix here.

(Refer Slide Time: 52:37)



Now, if I use A star B the corresponding element will be multiplied. So, here 1 into 7 I am getting 7, 2 into 8 I am getting 16, 3 into 9 I am getting 27, similarly I am getting 40, 55 and 72. So, the first line is creating matrix A then this line the third line is creating matrix B and then the fifth line that is basically executing C equal to A star B ok. So, this is and I am getting this particular matrix C.

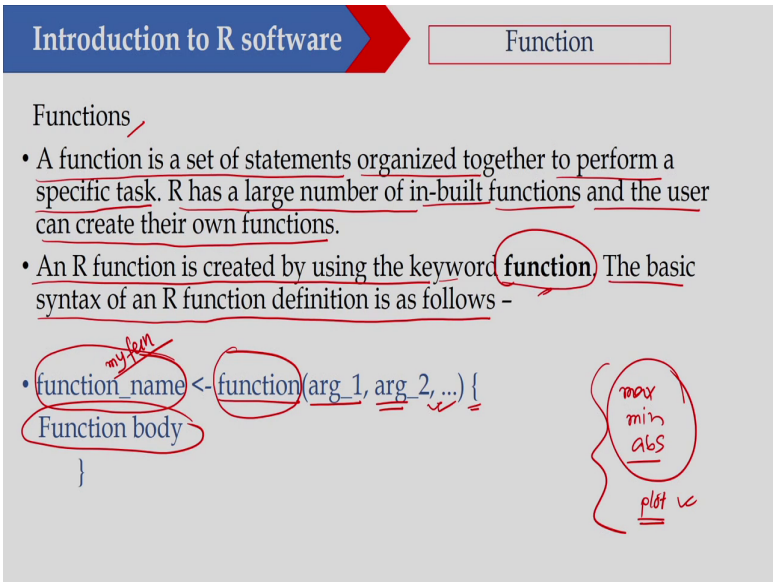
(Refer Slide Time: 53:17)



Now, if I want to do matrix multiplication; that means, this is the matrix A and this is B, I would like to multiply this one A into B ok. So, in that case what I have to do basically? So, the result is 3, 7, 6, 14. So, how I am getting? That 1 into 1 ok. So, this I am getting 1 into 1 plus 2 into 1 ok then I am getting 1 into 2 plus 2 into 2, I am getting 3 into 1 plus 4 into 1 and I should get this is 3 into 2 plus 4 into 2 ok.

So, this value is this is 1 plus 2 that is 3 this is 3 plus 4 that is 7 then this is 2 plus 4 that is 6 and this is 8 plus 6. So, that is 14 ok. So, I should get that one. So, to do that, so, what I have to do? I have to use this particular symbol A star percentage B ok. So, I can do the A cross B. So, finally, I am getting this particular solution. So, this is the matrix A, this is matrix B and I am getting the matrix C.

(Refer Slide Time: 55:04)



Now, in R I can also write the function ok. A function is a set of statements organized together to perform a specific task. R has large number of inbuilt function. So, it has already inbuilt function suppose you are using max, you are using minimum, you are using absolute abs absolute. So, these are all function inbuilt function and there is a function called plot also ok. So, I can plot it. So, these are all inbuilt functions and the user can create their own function. So, I can create my own function and once I am creating I can use this function just like an inbuilt function.

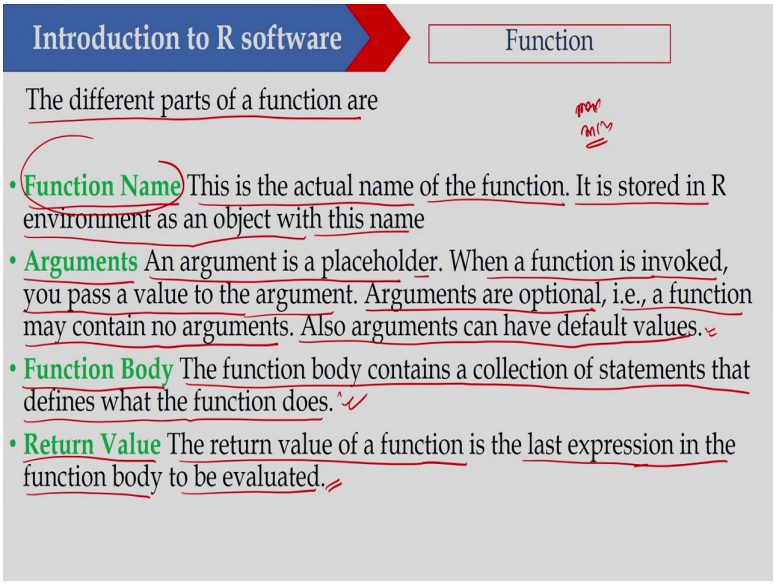
An R function is created by using keyword function. So, this is the keyword I can use to create a function. The basic syntax of R function defined as follows ok. So, what you have to give? You have to give a name of the function. So, this is function name then I have to use

this particular keyword. So, I cannot change that one. So, this is the keyword I have to I have to use that is the function, but this name you can give anything ok.

So, this is suppose I want to give my function myfun ok. So, this is whatever name you can put it and then this is the argument what is coming into the function ok, argument 1, argument 2, argument 3, something like that. So, whatever the argument you want to give. So, you can give the argument. So, this variables ok will basically will come to this particular function and then there is a function body.

Then I will use second bracket within bracket I can write the function ok. Suppose for example, I can define a function suppose anyway. So, I will go to the next slide and I will basically discuss how to create a function ok. So, this is user defined function that means I would like to create a particular function.

(Refer Slide Time: 57:14)



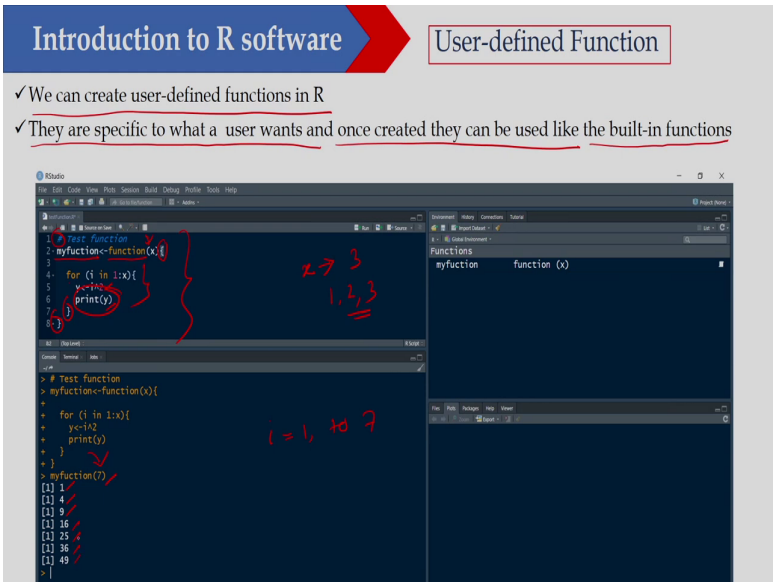
So, as I have discussed the different parts of the function are the function name. This is the actual name of the function. So, you can define this, but you cannot define this function as if suppose that name is reserved. Suppose I cannot define a function like max or min or something like that because this is the default function already it is available in R. So, it is stored in R environment as an object with this name ok. So, this name you can define, but some of the name you will not be able to take.

Then there is a argument. As I said an argument is a placeholder ok. So, when a function is invoked you pass a value of the arguments. So, what we are doing? We are passing the value of the argument to the function and arguments are optional ok. So, it means that there may be argument and there may not be argument; that means a function may contain no arguments also.

So, there may not be any argument on a particular function also arguments can have default values. So, sometime this arguments can have also default values. Then second part is function body. The function body contains a collection of statements that defines what the function does ok. So, as per our need, so, I can define this.

So, whatever the objective of this particular function, so, I can define the statements here and the return value the return value of a function is the last expression in the function body to be evaluated. So, it will return the last expression value. So, that will return by this particular function.

(Refer Slide Time: 59:10)



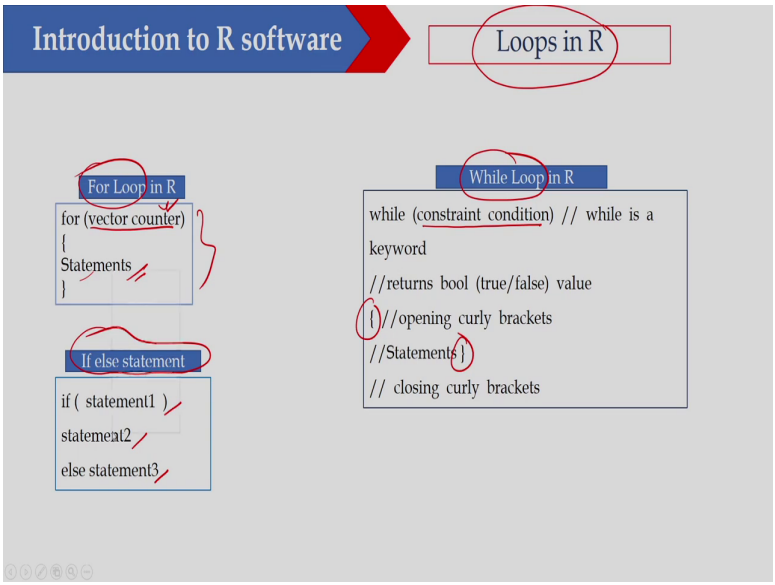
So, let us see an example. So, as I said the we can create a user defined function in R. They are specified to what a user wants and once created they can be used like a inbuilt function built in functions ok. So, once you are creating, so, you can use it as a built in function. So, suppose here I am defining one function. So, this is a test function. So, this is the comment line and name of the function is myfunction.

So, you can give any name. So, I have given myfunction and this is I cannot change, this is the function and this is the argument, argument is x ok. So, x is entering. So, x in this case this is a scalar value and x is entering into the function. So, we are passing the value of x to this particular function and then I am using this second bracket. So, within second bracket I am writing this particular body ok.

So, what I am doing here? For i equal to; I will also discuss this for loop later on, but right now you just see the for i i in 1 in x ok. So, 1 to x basically that x is suppose x value is 3; that means, 1, 2, 3 ok. So, 1 to x that y equal to i square; that means, it will square the value of i and then I am printing.

Now, if I execute that one, suppose now the argument I am argument is x and the value I am putting is 7; that means, what you will do? I will go from I will go from 1 to 7; that means, 1, 2, 3, 4, 5, 6, 7. So, what you will do? Suppose first time it will come 1, then 1 square that is 1, then 2 square that is 4, 3 square 9, 16, 25, 36, 49 and then print command will print the value of y ok. So, this function will return the value of y ok. Or here what you see? It is printing the value of y.

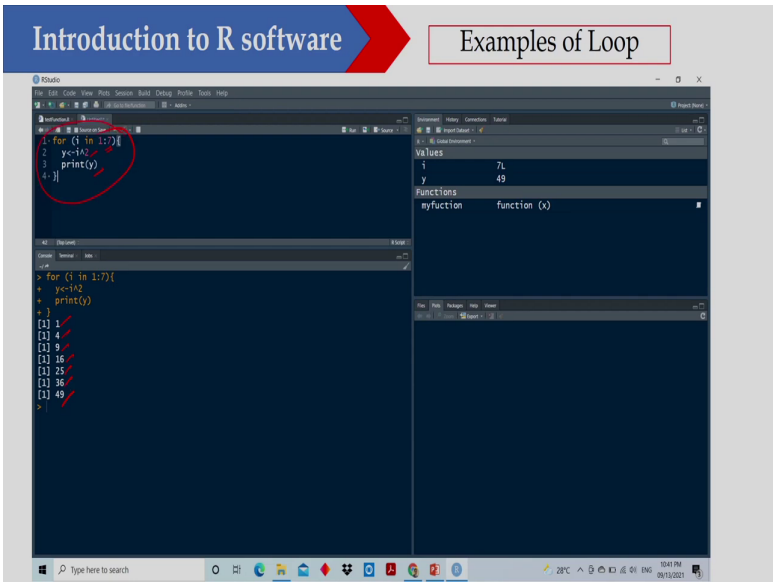
(Refer Slide Time: 61:27)



Now, let us see how you can execute the loop in R. So, here we can execute loop in R using for loop ok. So, this is one option. I can also use while low and I can also use else if statement ok conditional statement. So, here what we are doing for this is a for loop for vector count and then within second bracket this is the statement. So, this will be executed for this particular condition.

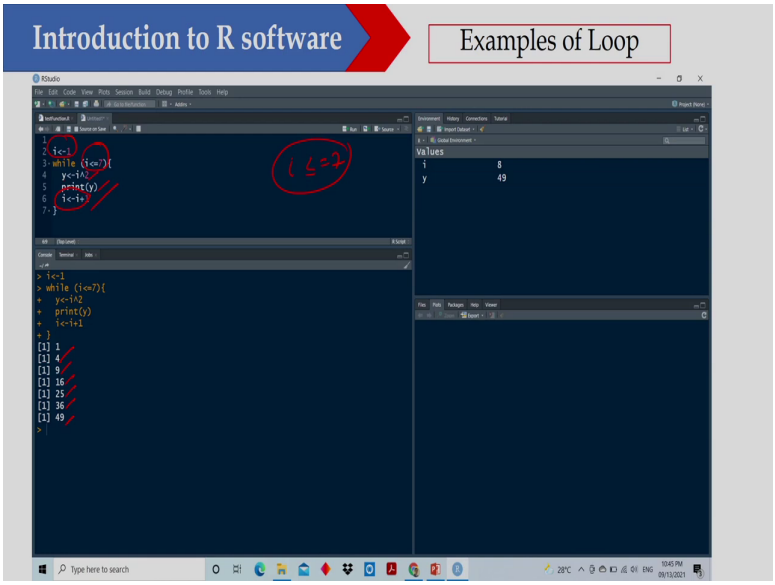
Similarly, if I write while that is the condition I am writing here what constraint condition I am writing here then it will return, so, what you are basically writing within this second bracket ok. And similarly I can also write the conditional statement if else statement that if this statement is TRUE then you execute this one else you execute the statement 3 ok. So, that is the if else statement.

(Refer Slide Time: 62:36)



So, what I can do? Suppose I would like to execute I would like to print the value of the same thing ok. I would like to print the value of y to do y which is equal to i square; that means, from i equal to 1 to 7 ok. So, now this is a for loop here. So, i in 1 to 7. So, y equal to i square and I am printing that one. So, you just see I am getting 1, 4, 9, 16, 25, 36, 49. Here I can use for loop.

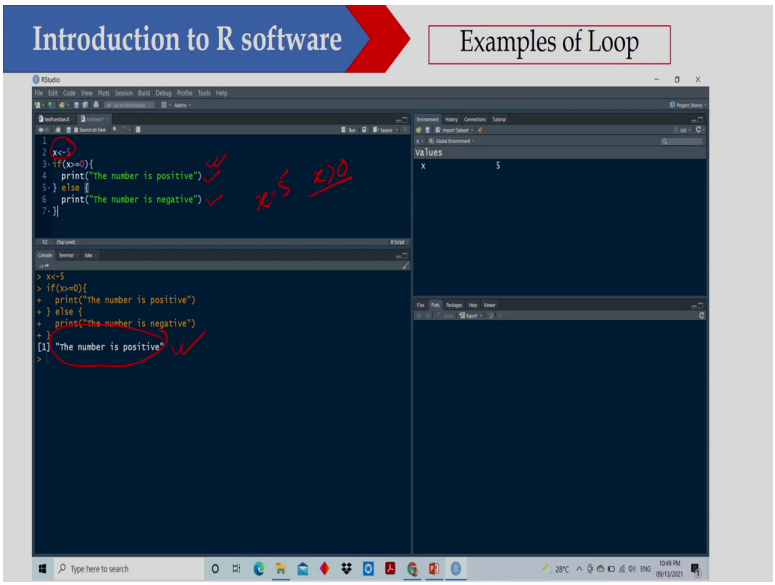
(Refer Slide Time: 63:12)



I can also use the while loop to do the same thing here. So, I would like to print y which is equal to i square. So, what I have done here? That i equal to 1 and then while this condition is TRUE; that means, i is less than equal to 7; that means, i is less than equal to 7 this condition is TRUE then what I am doing? i equal y equal to i square then print y and then there is a increment that i equal to i plus 1.

So, I am getting the same result; so, 1, then 4, 9, 16, 25, 36, 39. So, I can also use the while loop to execute this particular problem.

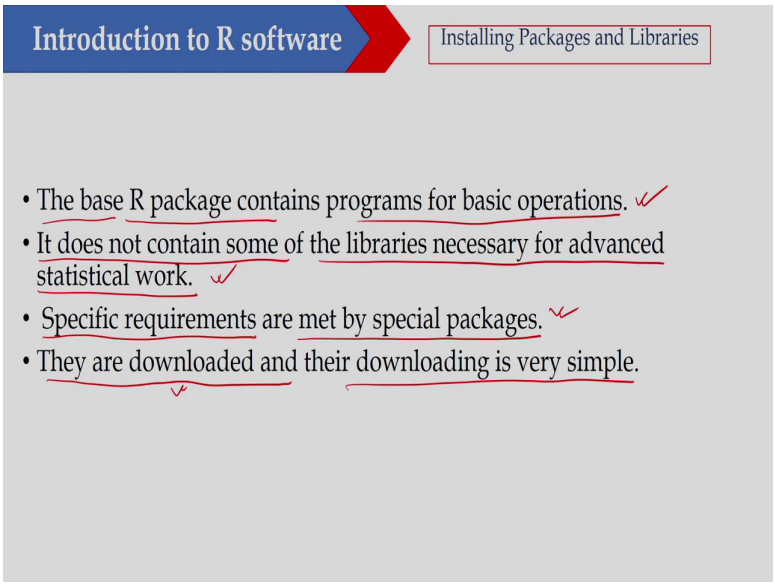
(Refer Slide Time: 63:58)



Now, let us see the if else statement. So, here what I am doing here? I am defining a value x equal to 5. Now, what I am doing? If x is positive then I am printing the number is positive else the number is negative. Now, if x is positive ok, so, in that case it will print positive else it will print negative.

So, in this case the I am putting x equal to 5. So, x equal to 5. So, this is positive. So, therefore, I am getting the number is positive. So, therefore, I can also use these if else statements to execute some of the statement like this.

(Refer Slide Time: 64:40)

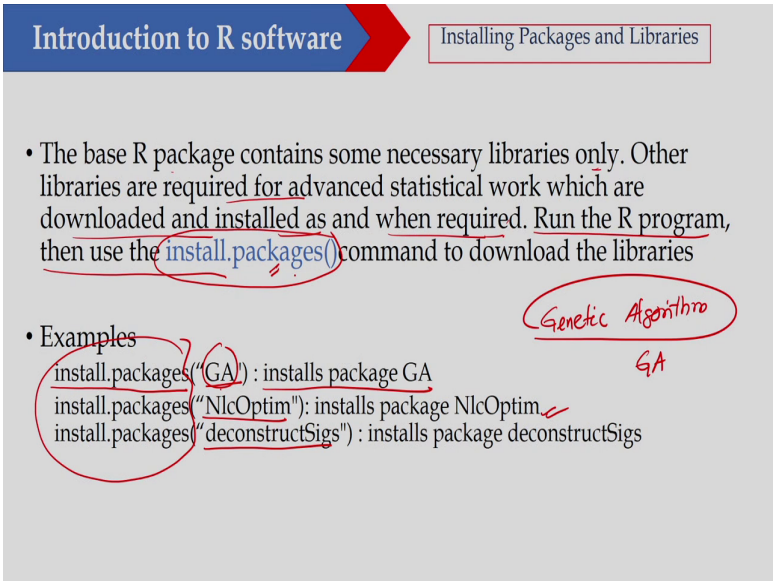


Now, as I said so, whatever I have; whatever I have discussed here, so, this is the basic R engine ok. So, as I said earlier the some of the advanced facilities are not available. Suppose I would like to work on GA ok. So, GA is not available here so, but GA package is available ok.

So, you can download that package you can install that package and you can include in your R platform and you can use it. So, the base R packets contain program of basic operations, so, that as I have discussed. So, it contains only the basic operation. You can write your own code. It does not contain some of the libraries necessary for advanced statistical work. So, that is not there.

Specific requirements are met by specific packages ok. They are downloaded and they are downloading is very simple. So, I can download these packages and downloading is also very simple and then I can install it ok.

(Refer Slide Time: 65:46)



The base R package contains some of the necessary libraries only. Other libraries are required for advanced statistical work which are downloaded and installed as and when required. So, as and when required, so, you need not do because sometime you need that one. So, at that time you can download it and you can install it. And run the R program then use install package.

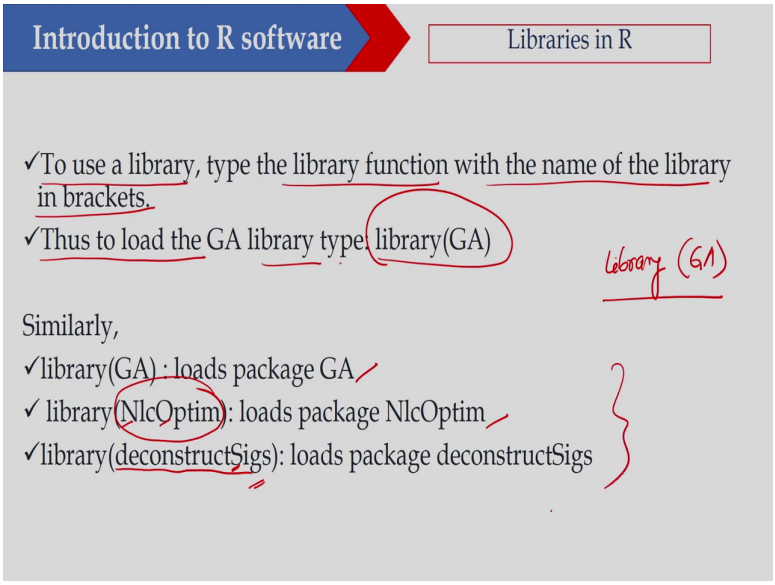
So, this is the command I can use to install a particular package ok. So, this command I can use. Suppose in the console itself I can write or in the program itself I can write that I would like to install this particular package. So, for example, I can suppose I would like to install

GA package. So, name of the package. So, this you have to know what is the name of genetic algorithm package ok.

So, what is the name of this particular package? So, name is GA. So, what you have to do? You have to write this particular command that is install packages and within code so, you have to write GA and this will install GA on your computer. Similarly, there is another package which is non-linear optimization toolbox actually automation package. So, this is NlcOptim.

So, it will you can install using this one and there is another one deconstructSig. So, that also you can use. So, you have to use the install package command to install the packages. So, what will happen? So, in the program itself you can write that one.

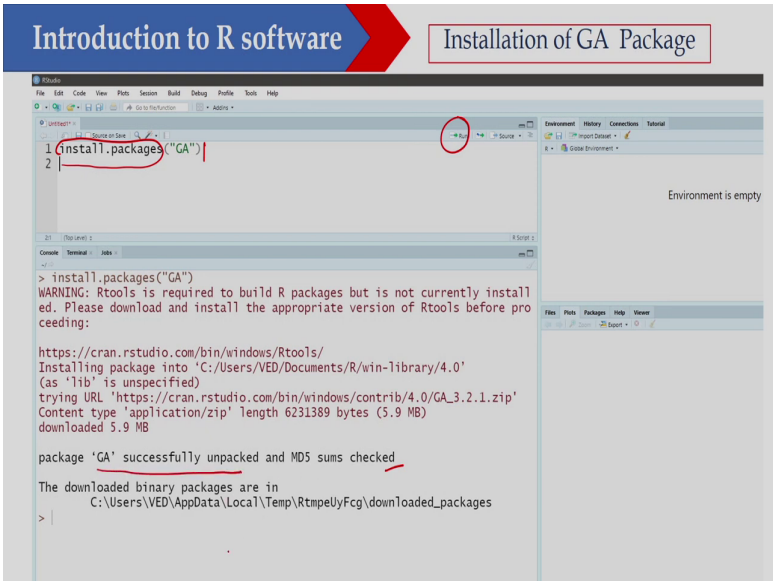
(Refer Slide Time: 67:32)



So, it will install the package and after installation what you have to do? You have to include that library to use the library type the library function with the name of the library in the brackets. So, thus to load the GA library type: library GA. So, in the program itself, so, what I will do? So, in the program itself I will write library GA ok. So, GA and this will include the library on your program and you can use it.

So, similarly, so, load this is GA, this is NlcOptim and this is deconstructSig ok. So, this is. So, you have to know the name of the library. So, you cannot say in suppose this is in capital letter you have to use capital letter and here it is S is capital. So, you have to use capital letter otherwise it will not work ok. So, you can include the library here ok.

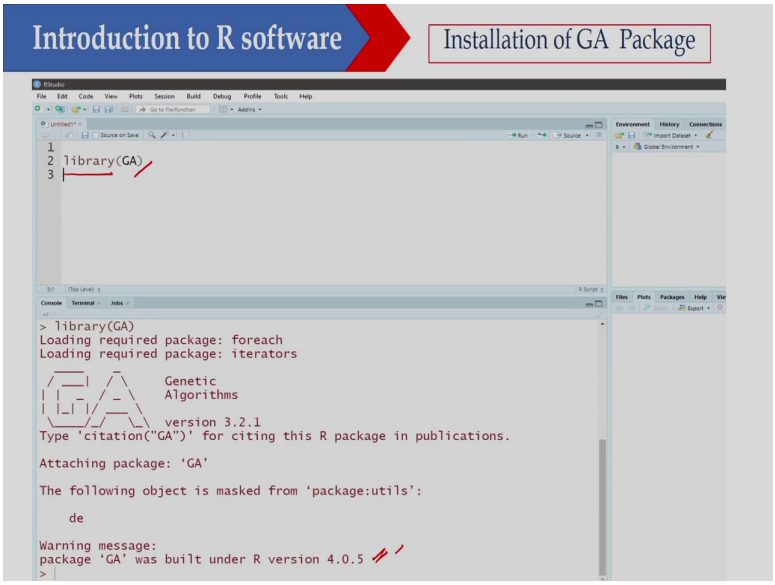
(Refer Slide Time: 68:37)



So, here you can see that as I said. So, I can install the GA package by using these particular command. So, install GA. So, if I execute this particular line using the Run button, so, you

can see that it will automatically. So, if you have the net connection on your computer, so, it will automatically download that one and it will install in on your computer. So, you can see the package GA successfully unpacked and it has been installed ok.

(Refer Slide Time: 69:07)



Now, then when we are using that one, so, I have to include that library. So, how I am doing that one? I am writing library GA. So, this will basically use or this will include the library on your program ok. So, on that particular program this will be or on your working environment this will be added. So, GA has been included here.

So, thank you very much. So, in this particular class, so, I have introduced you what is R, how you can download R, then again as I said that we are not working on R engine, but we are working using R studio. So, both are freely available. So, first you download the R engine and then you download R studio, but we are working on R Studio.

Then we have discussed the simple operation how you can create a vector, how you can create a matrix, how you can do the vector manipulation, how you can do the matrix manipulation so that we have discussed we also discussed how to create user defined function how you can create a function and this function once you are creating. So, you can use this function as a built in function. So, that you can use it and also we have discussed the different packages.

So, when I will solve the optimization problem using some of these packages, suppose I will solve the genetic algorithm using R. So, there I will show you how you can install the genetic algorithm package and then how you can include that one in your program and how you will use that one.

So, that I will discuss and so, this packages you have to download or you have to download and you have to install before using that one before including that one. So, if you are writing library, suppose without installation if I write library GA, so, in that case it will show you error because you did not install the library GA library on your computer. So, first you have to install it and then you can use that one. So, let us stop here.

Thank you.