Marketing Analytics

Professor Swagato Chatterjee

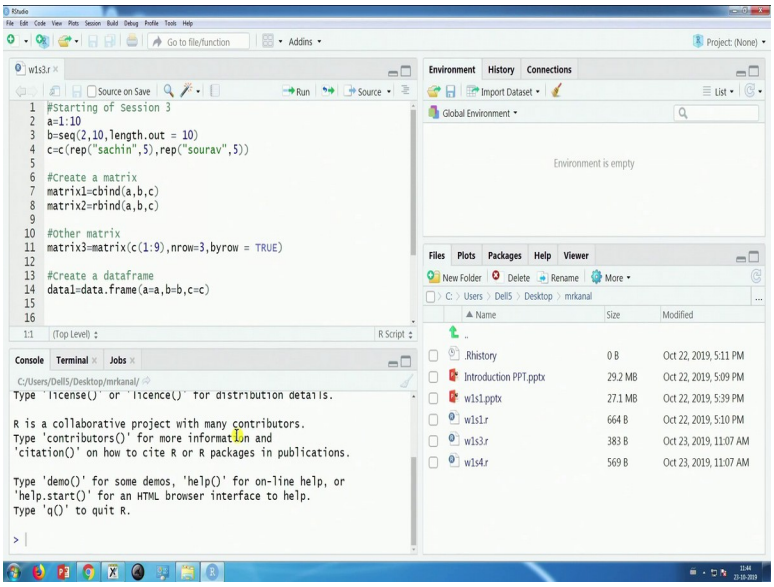
Indian Institute of Technology, Kharagpur

Lecture 03

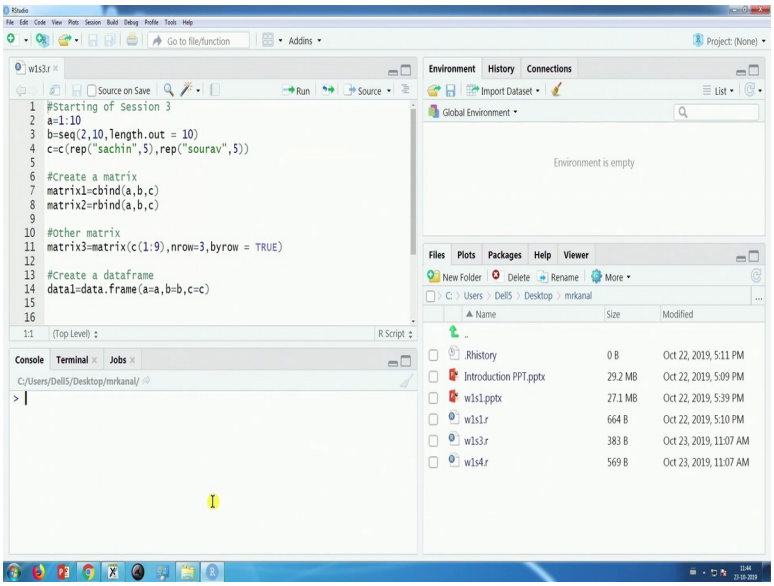
Introduction to R Programming (Contd.)

Hello everybody, welcome to session 3 of Marketing Analytics, this is professor Swagato Chatterjee I am from Vinod Gupta school of management, IIT Kharagpur and I will be taking this session 3 for you guys.

(Refer Slide Time: 00:36)



(Refer Slide Time: 00:43)



So there is another file called W1S3 dot r. I would ask you to open that file in R studio and it looks like this and I would ask you before you will start you should clean your console, pressing Ctrl+L.

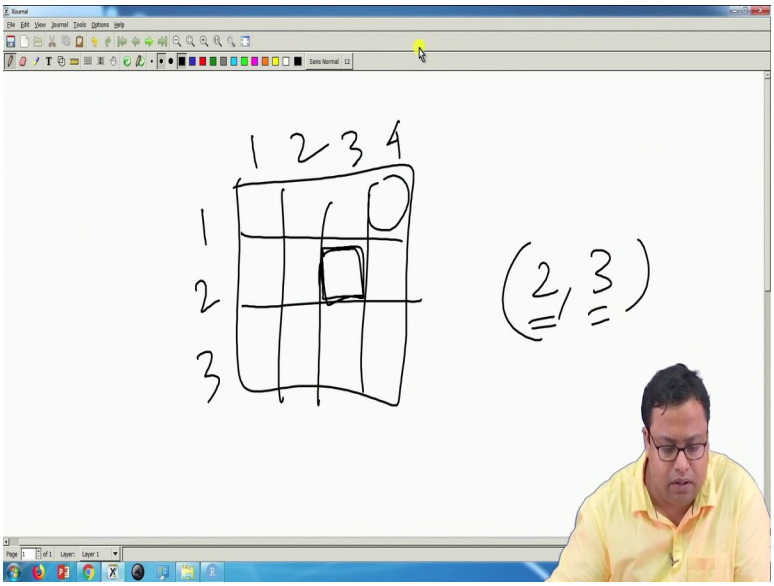
You should clean your global environments, so there is a small brush sign here you can click on that and it will clean the global environment and the later point of time I will show you how to save the workspace also, right now we do need that we are just practicing. But later point of time will show you how to save whatever is here. So whatever is here will be saved and any other files if it is open I told you on the first day that it is very important that you keep things clean when you are learning a little bit of coding.

So any other file if that is open here please close it, the only file that should be open is W1S3.r, the console should be clear, your global environment should be clear, if you are with me at this particular level then we can start. Now in the last class, we have talked about vectors, we have talked about numeric vectors, we have talked about character vector, we have talked about factor

vector and we have told you that we have discussed that how character vector and a factor vector is different from each other.

So in this particular class, we will try to know that a little bit more, other objects, the second important object is matrix and the third important object is called a data frame. So the first thing is that we will create a matrix. Now, what is a matrix?

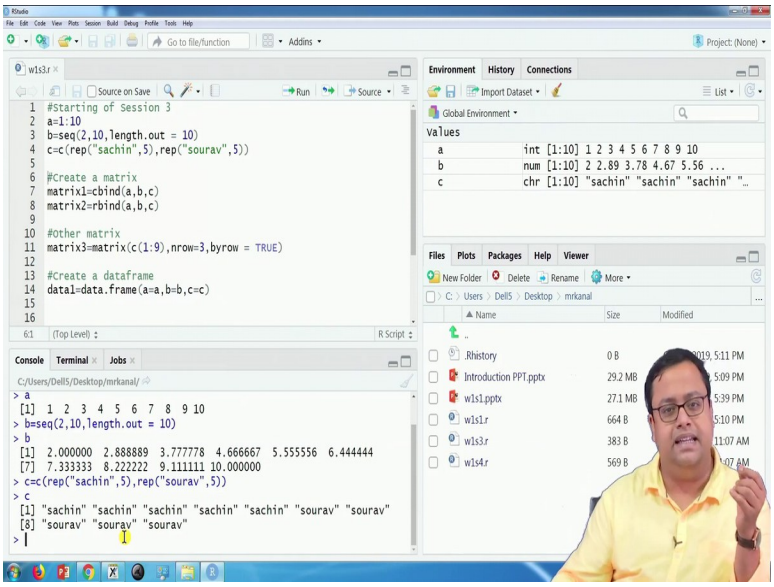
(Refer Slide Time: 02:15)



A matrix is something that looks like this, so a matrix is a tabular form, it looks like a table and each of these things is the cell and each cell will have to a continuous similar type of data. So if one cell is numeric the all other cells have to be numeric, if one cell is categorical or character all other cells have to be a character variable.

So otherwise, it does not work, so matrix all the elements in a particular matrix has to be same and these are the column numbers 1, 2, 3, 4 and this is the row numbers 1, 2, 3 and each cell is known by its row number first and column number second. So if I want to find out what is the average of this particular cell is 2, 3. 2 is the row number and 3 is the column number. So based on this particular idea we will actually learn matrix and how to create a matrix using R.

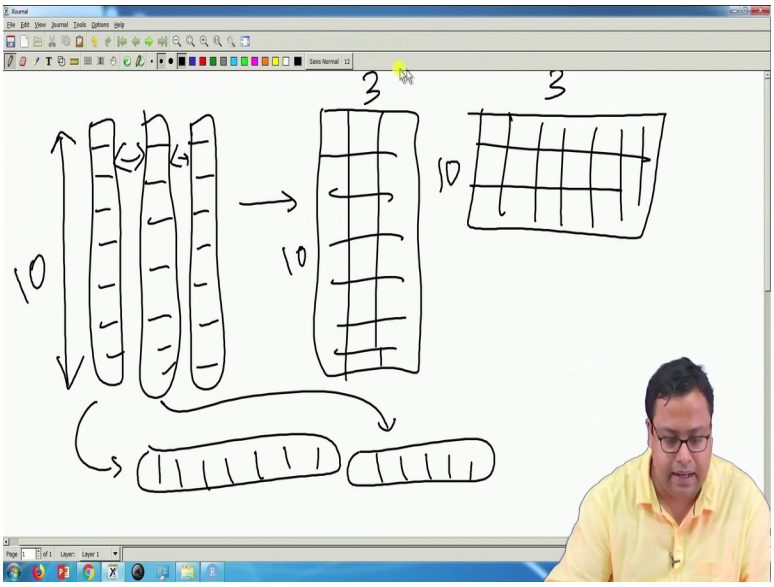
(Refer Slide Time: 03:16)



So the first thing that let us say I am creating 3 different variables, 3 different vectors and then I will join them to create a matrix. So the first vector, the first 2, 3, 4 lines you just run on your own and you will understand what these guys are doing, line number 2, line number 3 and line number 4. The first line is I am creating A 1 to 10, it is particularly an integer vector which starts from 1 goes up to 10. b is a sequence which starts from 2 ends at 10 and the length output is also 10, so 2, 10, length dot out is equal to 10.

So the moment I run that 2 to 10 these values are actually broken into probably 9 equal intervals so that you get 10 values or something like that. So 2 then 2.889 then 3.778 and so on, so that is basically a numeric vector it is not an integer anymore. On the other hand, if you see the C, it is a character vector where the first 5 entries of the character are Sachin, the next five entries of the character are Saurav. So this is C, now I will join this A, B, and C to create a matrix. Now I can join in two different ways.

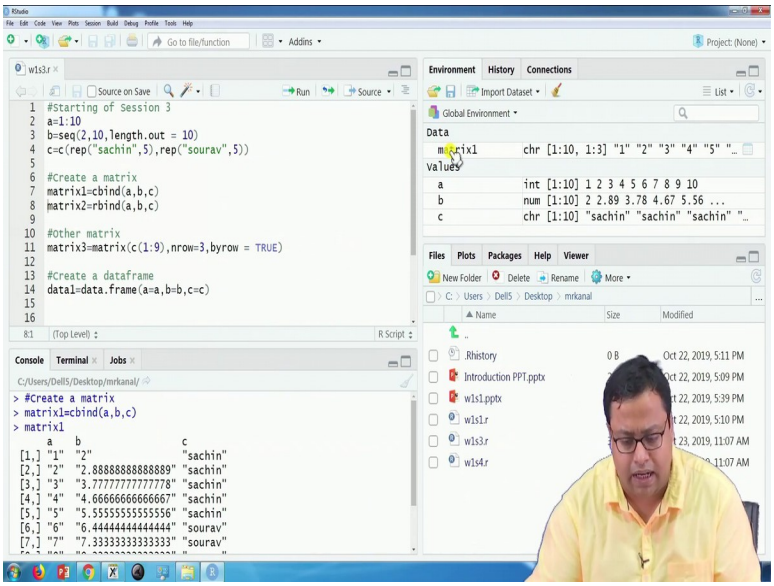
(Refer Slide Time: 04:44)



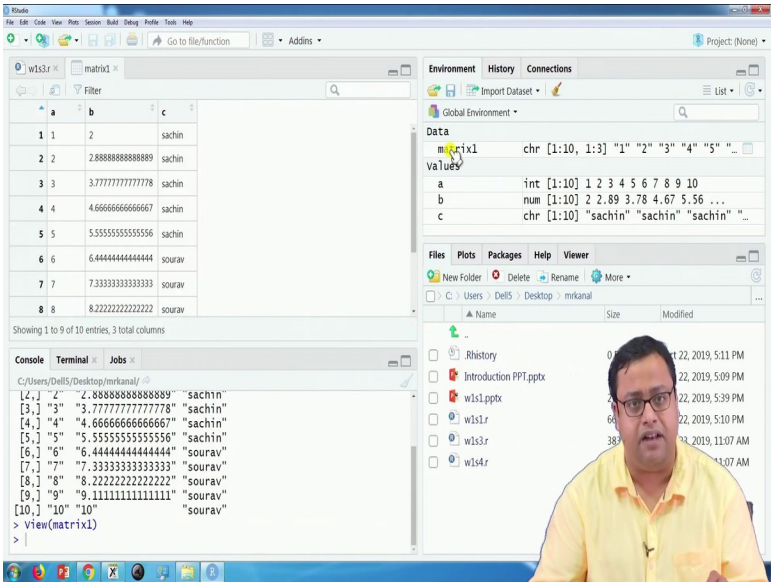
So let us say if I have 1 to 10 but one matrices which looks like this, similar to another matrix the same size and shape and another matrix which is the same shape. So I can actually bind them like this so side by side and create something which looks like this or I can rotate them, I can make this guy looking like this and this guy looking like this and so on. And then bind them one above other, so I can actually create something like this.

So if each of them has 10 entries and there are 3 columns then these guys will have 3 columns and 10 entries, on the other hand, these guys will have 10 entries, 10 rows, and 3 columns. So the first one is actually called a column bind and the second one is called a row bind and that is what we will be doing here in this particular hour. So I will be doing that.

(Refer Slide Time: 05:56)



(Refer Slide Time: 07:12)

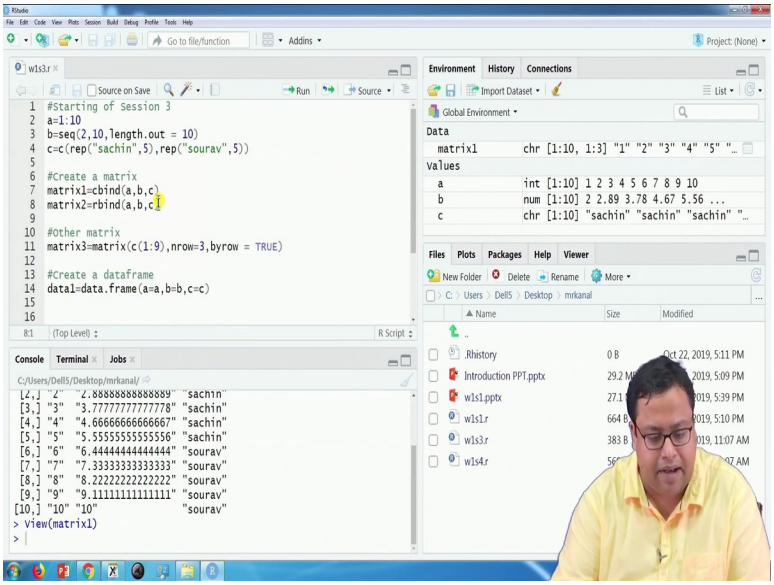


So the first thing that I will do is matrix 1, if you see, matrix 1 is equal to cbind, a, b, c if I run this you will see here in the global environment under matrix of type data, so that is why it is but there is a difference between matrix and data frame will talk about that. But it is coming as

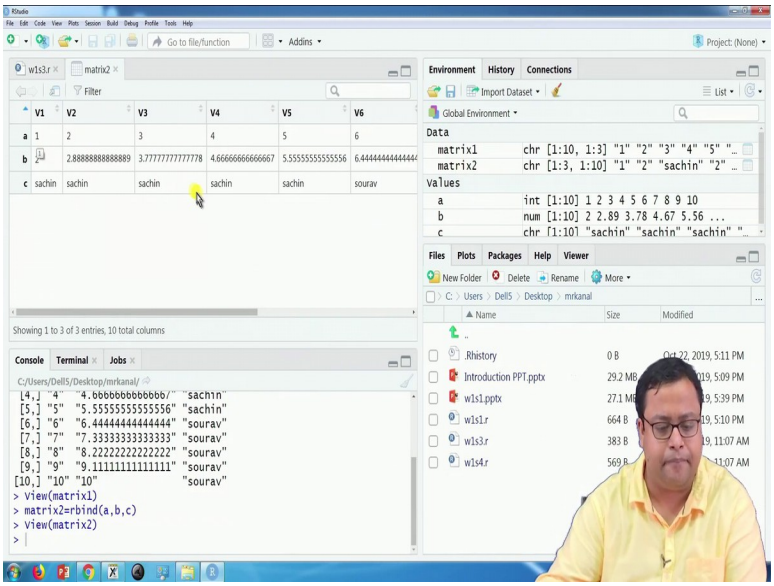
matrix 1 which has a character matrix 1 to 10 and 1 to 3 which means the rows rise from 1 to 10, the columns vary from 1 to 3, so there are 10 rows and 3 columns and there are some values.

Now I can just write matrix 1 here, you will see that all the values there was integer vector, there was a numeric vector and there was a character vector. But when I join them because character vector is superior to them and every character cannot be converted to a number but the number can be converted to a character. So they converted everything to a character and the matrix that is when created is a character matrix and it has been created in such a way that a, b and c are 3 columns so that is called cbind, column bind.

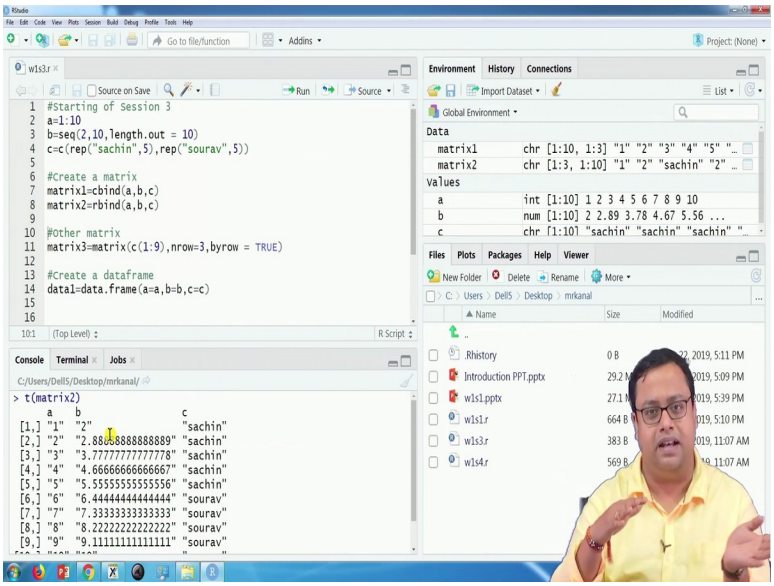
And I can also click here or I can write capital V I E W, view within bracket matrix and that will give me a spreadsheet-like view of the matrix. Here you will see that all the values are left-oriented means left all the values is actually align towards the left of the cell, so all the values are characters. (Refer Slide Time: 07:33)



(Refer Slide Time: 07:48)

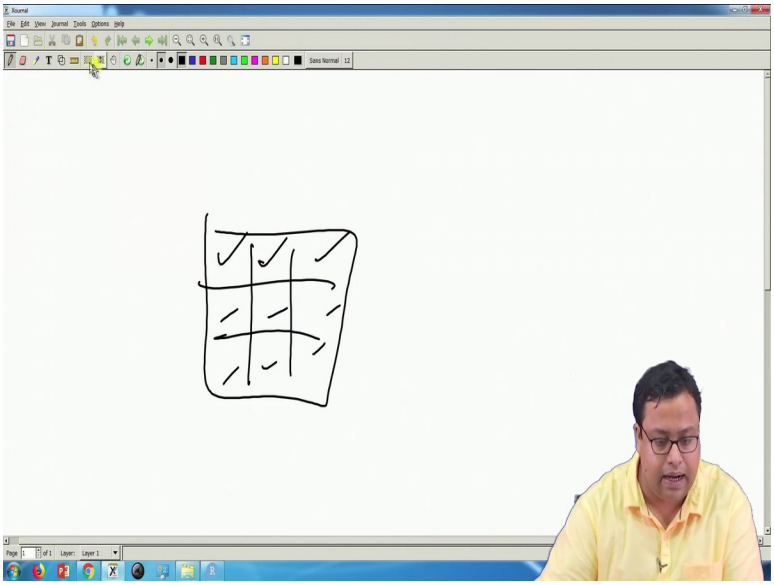


Similarly, I can do a matrix 2, where I have the only thing that I changed in line number 8 is instead of cbind I have written rbind so that is row bind it binds by rows. So if I click on that and if I want to see matrix 2 the same thing but now things are in a linear form so if the first one is 'a', first row is 'a', second row is 'b' and third row is 'c' and something like that, so it is nothing but a transpose, so you can also do a transpose in the matrix and you can say that okay. (Refer Slide Time: 08:09)



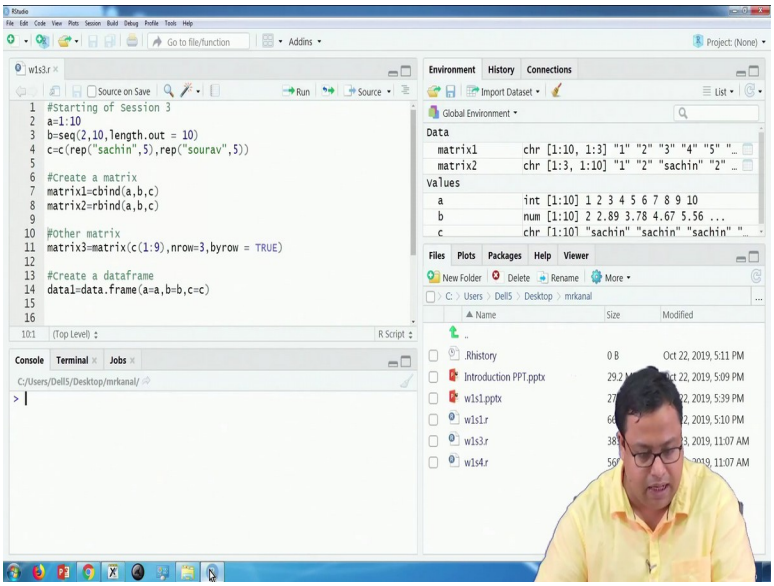
So if I do let us say transpose of T is the function transpose of matrix 2 and press an enter I actually get something like a matrix on its doing a transpose. So whatever is the in the rows I can get in the column and so on. So that is one way of creating a matrix when you create the matrix from 2 or 3 different vectors. Sometimes we populate a matrix also so we say, sometimes we say that a matrix will look like this.

(Refer Slide Time: 08:46)



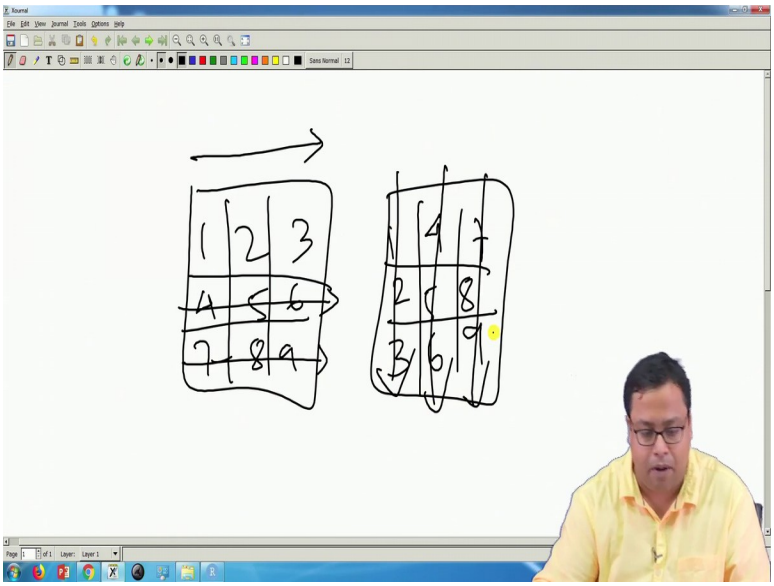
So this is my matrix and I will populate I will write a value here, write a value here, write a value here and so on. So if I go on writing the values here manually one by one I can create a matrix also. So sometimes that is required and will see when that is required and to do that we will use a function called matrix.

(Refer Slide Time: 09:07)

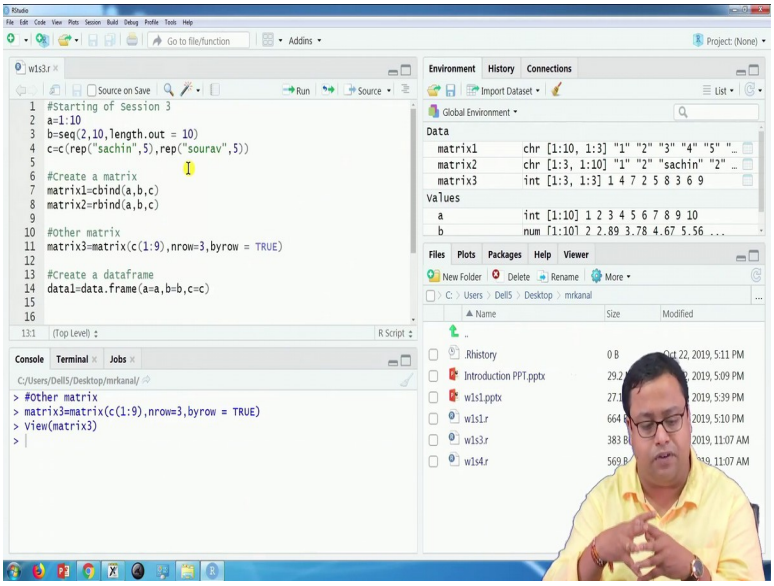


So here if I go in line number 11 you will see matrix 3 is equal to, so matrix 3 is the name of the matrix which I am giving is equal to the matrix function, the first syntax of this function is what are the values? So the values are C 1 colon n, so 1 to 9, 1, 2, 3, 4, 5, 6, 7, 8, 9 these 9 values will be is given which will be used to populate the matrix and how many rows will the matrix have? I have written n row is equal to 3. The moment I write that if there are 9 values and the number of rows is equal to 3 then how many columns do you want? 3 obviously, 9 / 3, so there will be 3 rows and 3 columns.

(Refer Slide Time: 09:57)



(Refer Slide Time: 10:30)

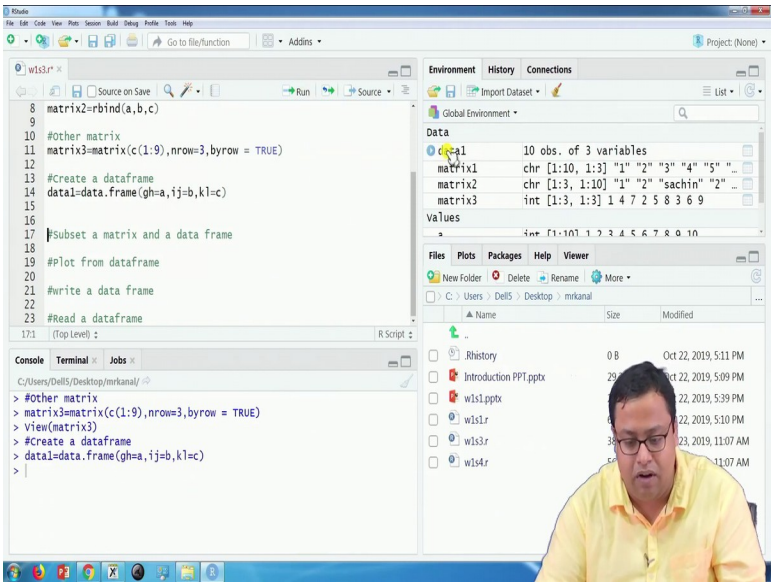


Now I can write these things when I populate it, I can populate it in 2 different ways if it is a 3 by 3 matrix I can write 1, 2, 3, 4, 5, 6, 7, 8, 9 or I can write 1, 2, 3, 4, 5, 6, 7, 8, 9. The first one is I am populating by row, so row-wise first row, then the second row, then third row and here it is column-wise first column then the second column then the third column. So I can populate it either row-wise or column-wise and that can be given by just writing 'byrow' is equal to true or

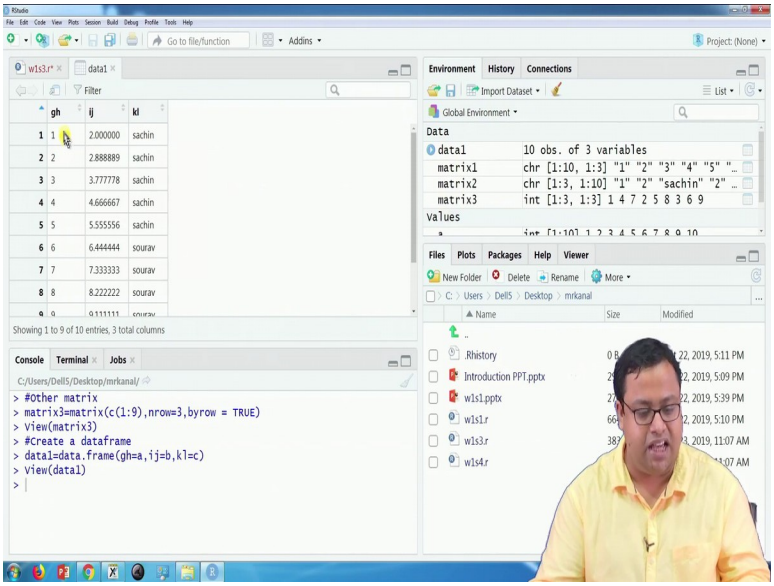
false. If it is byrow is equal to true then it will populate it row-wise and if it is false it is by default it is false. If I do not write anything it will treat it as false, so row-wise population will happen.

But if I write byrow is equal to true then row sorry if I do not write anything column-wise will happen and if I write byrow is equal to true then row-wise will happen. So if I see matrix 3 you will see that 1, 2, 3, 4, 5, 6, 7, 8, 9 it is coming row-wise. So that is another type of matrix formulation, 3 types of matrix formulations I have shown you, cbind, rbind and matrix as a function.

(Refer Slide Time: 11:13)



(Refer Slide Time: 12:13)

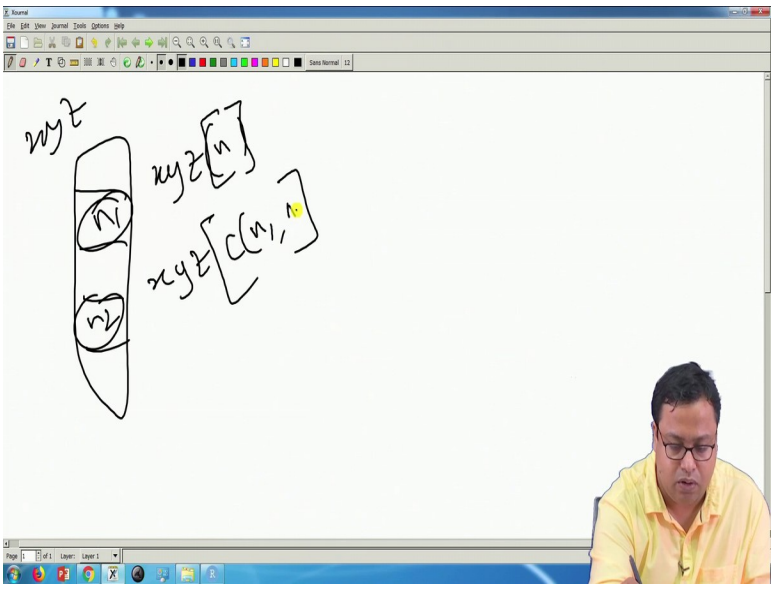


Now I will actually try to show you how to create a data frame, it is different. In a data frame, we write data 1, the name of the data is equal to the data dot frame and then I am in this particular case I am putting these 3 columns separately, I will not put it actually row-wise I want to put it in

column-wise. So I am writing a is equal to a, b is equal to b and c is equal to c, so these a, b, c the first a, b, c the left-hand side of this small things are the names that you want to put in the data set and these a, b, c are the vectors.

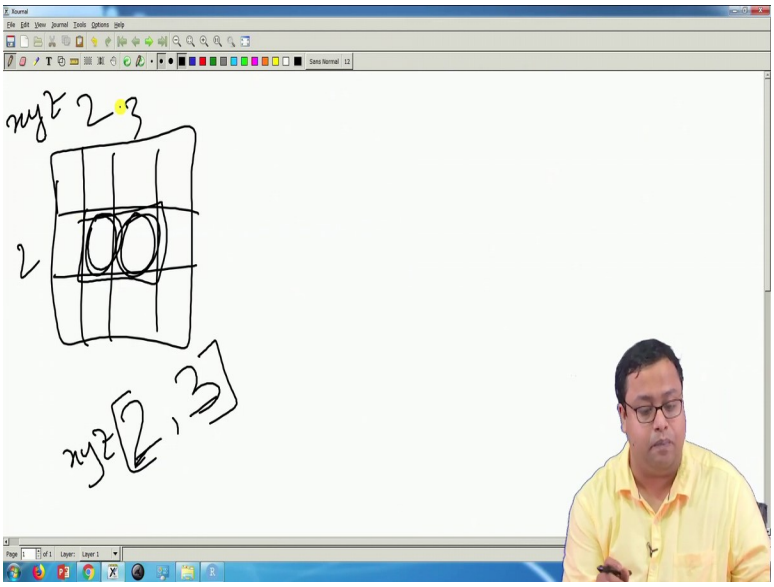
So you can also write instead of a, b, c you can write okay let us say gh, ij, and kl , so gh is the column name , ij becomes the second column name and kl becomes the third column name. So if I now want to see data, gh is the first column name and the entry of the first column name means whatever was there in the a vector. Similarly ij is the second column name and the entry of these values is whatever was there in the b vector and similarly, kl is the third column name.

So I am getting 3 columns with these column names and corresponding entries. So this is the data frame. You will see here the weight is written is also different. In case of matrix it is 1 to, 1 colon 10, 1 colon 3. On the other hand in the case of data set it is saying 10 observations of 3 variables. So the formulation is also different. But I will say what is the difference there. Now we will try to do a little bit of sub-setting of the data set. (Refer Slide Time: 13:05)

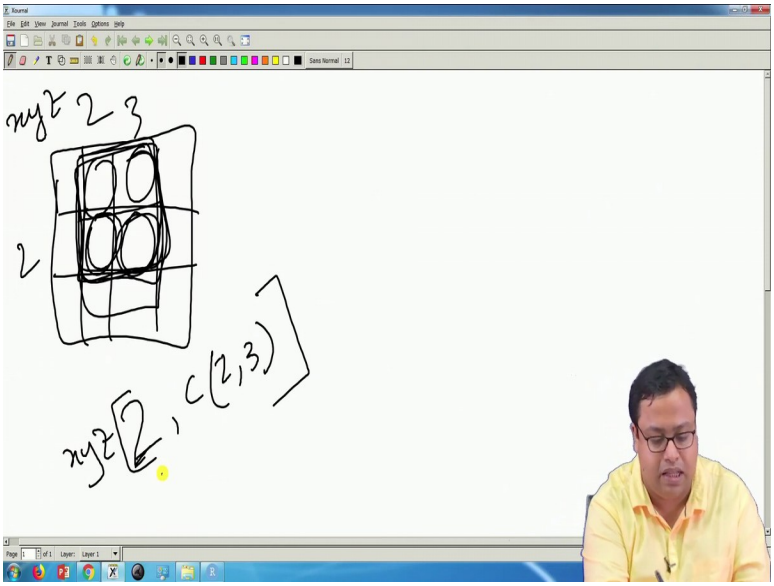


If you remember in case of a vector we told that if this is my vector which is x, y, z and this is the entry which is the nth cell I will write x, y, z, n and if you want to know more cells n1 and n2 you will write xyz , so that should be the third bracket, third bracket C n1, n2 so his is what you generally write when there is a vector and that the same thing the same idea is something that will write it here in case of a matrix also.

(Refer Slide Time: 13:39)



(Refer Slide Time: 15:08)

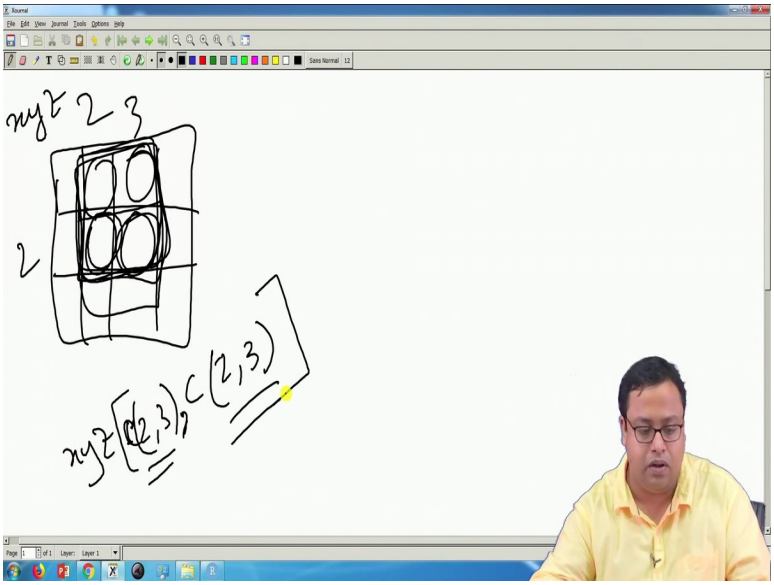


So let us say if this is my matrix, sorry if this is my matrix and I want to know which cell whatever is there in this particular cell, so this is actually second row and third column, and if the matrix name is xyz I have to write xyz, and then this is my address, and in the address, there

have to be two entries something has to be written before the comma, and something has to be written after the comma. So before the comma part is the row number, in this case, it is 2, and after the comma part is the column number in this particular case, it is 3.

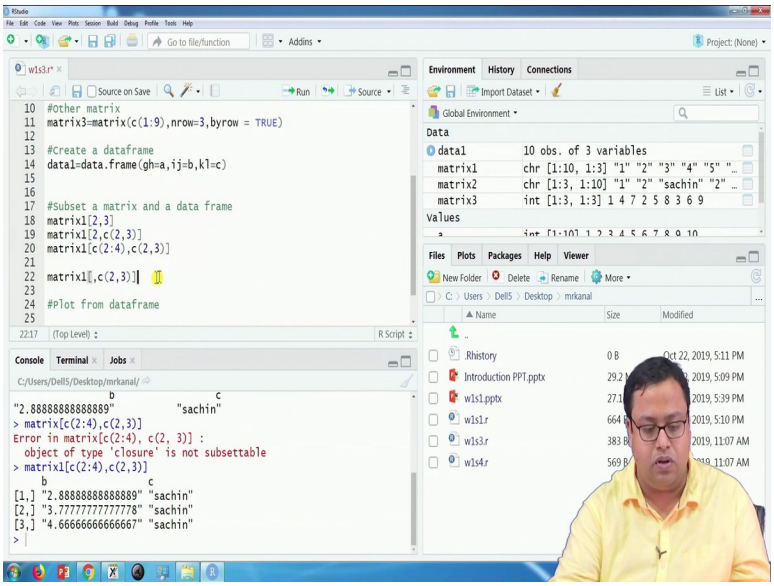
So 2, 3 that gives me this cell, so if I want more cells, let us say I want this cell and this cell both of this cell. So then I am actually asking you second row and second and third column, so in the columns space, I have to give two entries instead of 1. So what I will do here, instead of 1 entry, I will write here in this part 2 entries. I will write C 2, 3 two entries. Now if somebody asks okay, not only these 2 rows, 2 columns, I also want these values.

So this is my subset, this part is my subset. So if that has been asked to me then I am not even, you are not asking me 2 columns you are also asking me 2 rows. So here also I have to give 2 entries (Refer Slide Time: 15:11)

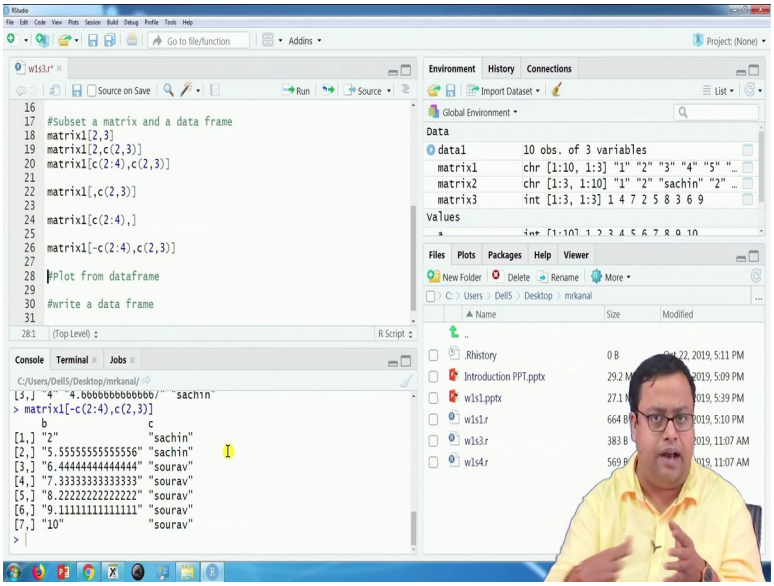


So I will instead of writing this, I will write C(2, 3). So something was written before the comma and something written after the comma will give me the subset of a particular column and that is how particular matrix and that is how you actually subset a matrix.

(Refer Slide Time: 15:42)



(Refer Slide Time: 17:48)



So we will do the same thing here in this particular case, so if I go here in line number okay I have to write it, so that is a matrix 1 and then I write 2 comma, so there are 3 variables so 2, 3 and if I run this much it will give me Sachin because second row third column is Sachin. If I

want a little bit more, if I write matrix1 and then 2, C 2, 3 that means see I have written 2, 3 after comma and before comma there is only 2 that means second row and second and third column if I run this it is giving me B and Sachin.

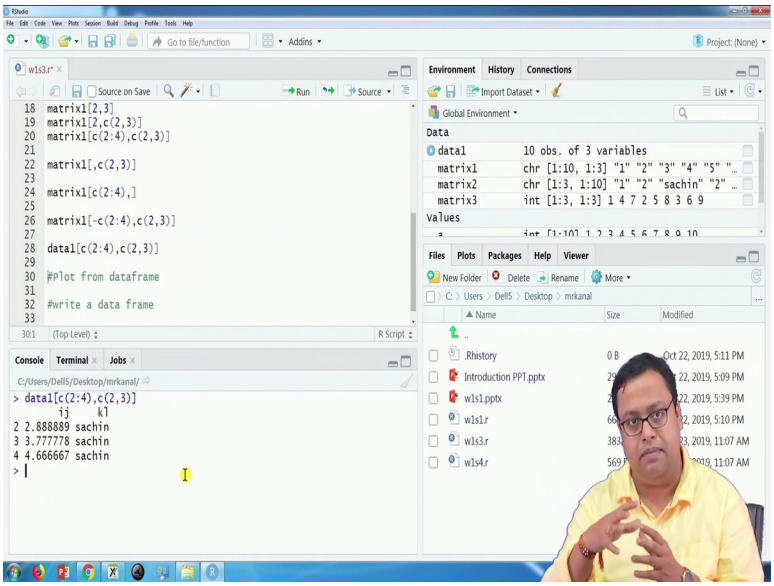
So second row whatever is there and second and third column. Similarly, if I write matrix c, let us say 2 to 4 comma C 2, 3 now check carefully. After comma 2, 3 that means second and third column, before comma 2 to 4 that means second, third and fourth row, if I run this okay sorry so there is a so it is a matrix 1 not matrix, so this is how error comes, this is a good thing that the error come, so the red error it will be written, so it is matrix 1, the wrong thing was there was nothing call matrix, there was a matrix 1 here.

So if I run this, you will get B second row third row, fourth row and second column, third column that is how you subset. Sometimes you may want to have all the rows, second column, and third column all the rows. So to do that, to do that, all the rows you write nothing before the column. So nothing before the column means all the rows, so I have written a comma here, nothing before the comma and after comma 2 and 3 that means second and third column but nothing before the comma means all the rows.

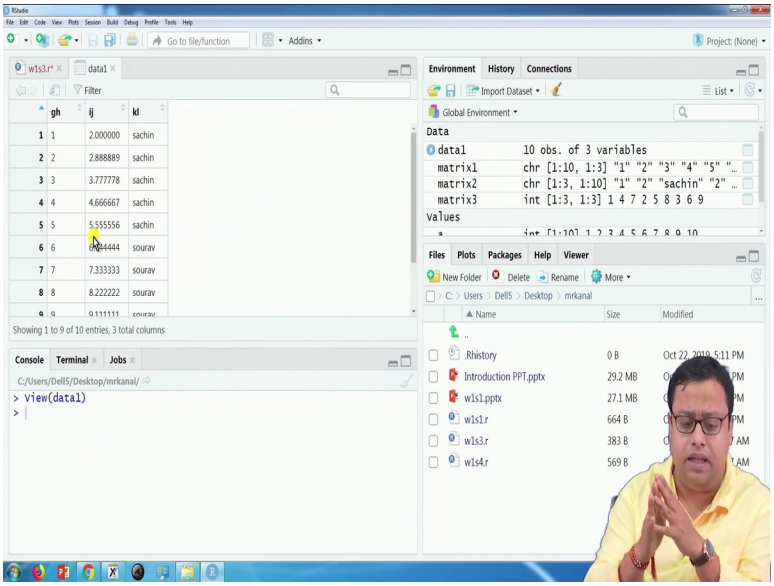
So the second and third column whole second and third column gets printed. Similarly, if I only want second and second to fourth row but all the columns, if I want second to the fourth row but all the columns, I will write nothing after the comma. So if I run this nothing after comma, second to the fourth row, all the rows are getting printed. Another interesting thing is, let us say I want to print all rows, other than second to the fourth row, all I have to write is minus, minus 2 to 4.

So it is a very interesting thing, so what it does, it does not print second to the fourth row, it prints all other rows because before comma I have written minus C 2 to 4 and after comma 2 to 3. So it is actually printing all of them. So I would suggest that I am going a little bit fast it is very good if you pause at it each point and run each line and see what is happening for you guys also. So that will help you to understand it properly. Now all of these things that I have done here on the matrix can be done for a data set as well. I can create a subset of a data set by this as well.

(Refer Slide Time: 19:14)



(Refer Slide Time: 20:05)

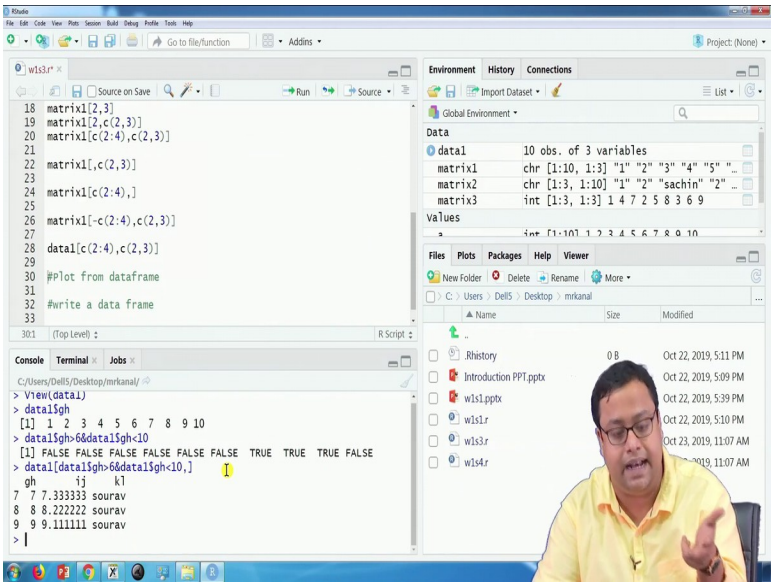


So I can just write the same thing I will just show one, I will write data 1 and then whatever I wrote here, I will copy and paste it here, everything that I did here can be done here as well.

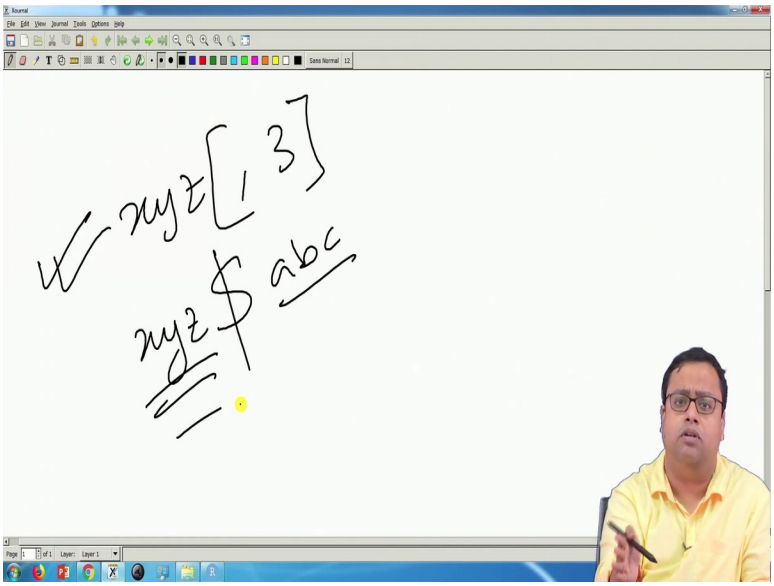
So again, I am getting second to the fourth row and second and third column in my data set. So I can subset it like this way. I can subset it also; sometimes, I can subset matrix and dataset also by putting a little bit of let us say conditions. For example, let us say I told you that I want such rows to be printed where the first row that means the first column that is gh is between 6 to 10, you listen to it carefully. So I want all such rows, the whole rows to be printed where the value of gh the first column in my data 1 is between 6 to 10.

So what do I have to do? I have to first find out where the value of GH is between 6 to 10, use that as my row numbers, and column numbers have to be everything.

(Refer Slide Time: 20:38)



(Refer Slide Time: 21:08)



So for my gh is between 6 to 10, so this is data dollar, data 1 dollar gh that is how you call up at this is only applicable for data frame not applicable for matrix. You cannot call a matrix's column with the name of the column. You can only call a matrix's column using the location of the column.

But that means you can write okay, you can write in this one, you can write if this is your x y z, x y z comma 3 that means the third column of xyz but even if the xyz column has some name in case of matrix you cannot call it like this whether this is applicable for both data frame and matrix, but for dataframe, you can also write xyz that is a data frame name dollar some column name let us say abc, abc is one column name, so this is the data set name and this is the column name.

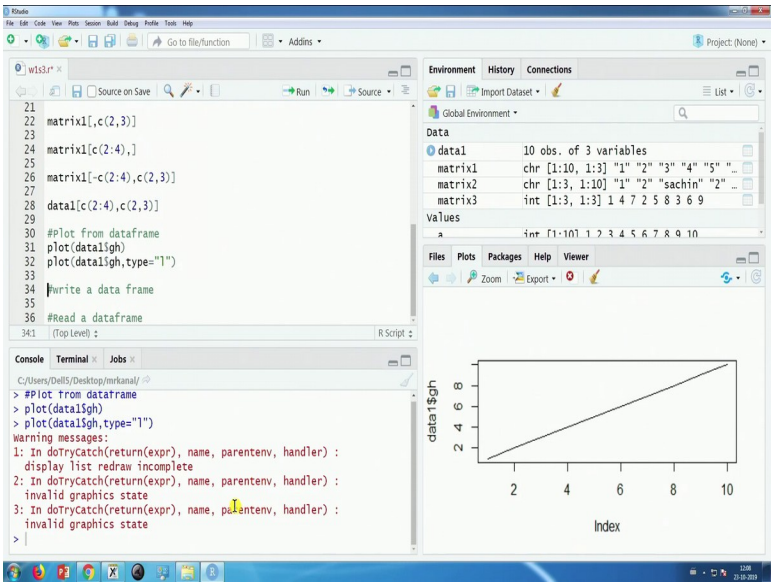
So I do not know the location of the column, but I know this column is there I can call it in this name. So that thing is not applicable here. So here I am writing data 1 dollar gh, data dollar gh looks like this. Data 1 dollar gh is higher than 6, and data 1 dollar gh is lower than 10. So a

greater than 6 and a smaller than 10 we did it in last class that gives me certain false, certain true. So there are 3 values that are coming true, and I want to use this as my row number.

So if I write carefully value c, if I write data 1 and how will I want to subset? I want all the columns; I want to print the whole row, so I want all the 3 columns, so nothing after comma, what is before comma? I want only such rows where the value of gh is between 6 and 10, what will I do? I will just copy this and paste it here. So the false and trues come up wherever the value is true it will get printed.

Now, if I just run it, you will see that 3 rows get printed where the GH value is between 6 and 10 that means 7, 8 and 9 so these 3 values are getting printed. So that is how we can do conditional subsetting as well; there are other things we will learn at a later point of time.

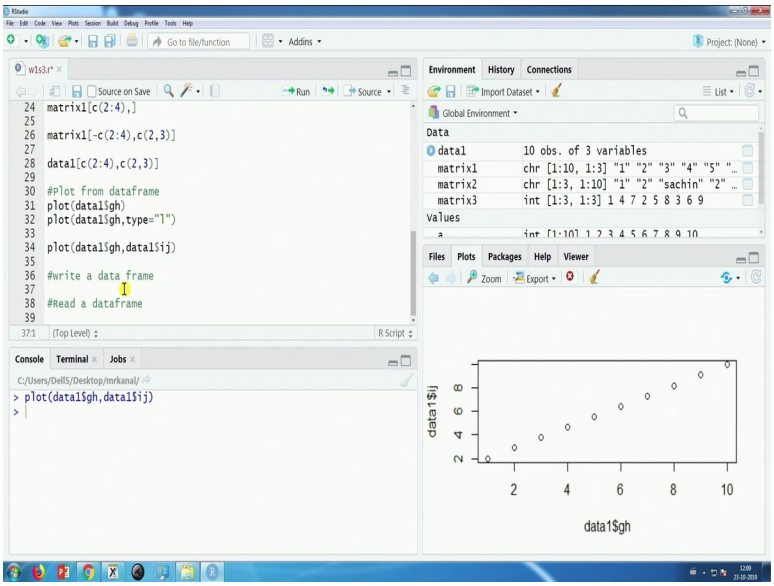
(Refer Slide Time: 23:18)



We can also write plot dataframes, so for example the easy one is there is a plot data 1 dollar gh. If I just run this one, see I am showing you another use of some of the tabs here. So if I use a particular data 1 dollar gh you will see that this part, oh sorry, so this part you will see I can actually zoom it up outside. So here this curve has been zoomed up and I am showing it here right now. So here you will see that 1 to 10 it is a linear curve, no so it has been printed, dotted lines have been printed, and you can plot something else also, you can plot the line curve also, you all the thing that you have to do is plot data dollar GH probably type is equal to l, and if I run this so you will get a line curve, there are certain warning messages.

Warning messages is something if it is not something which you cannot understand; you have to think a little bit about whether it has some meaning or it is relevant; otherwise, you can ignore it as well. So I get a line curve, there is a very basic line curve that I am getting.

(Refer Slide Time: 24:38)

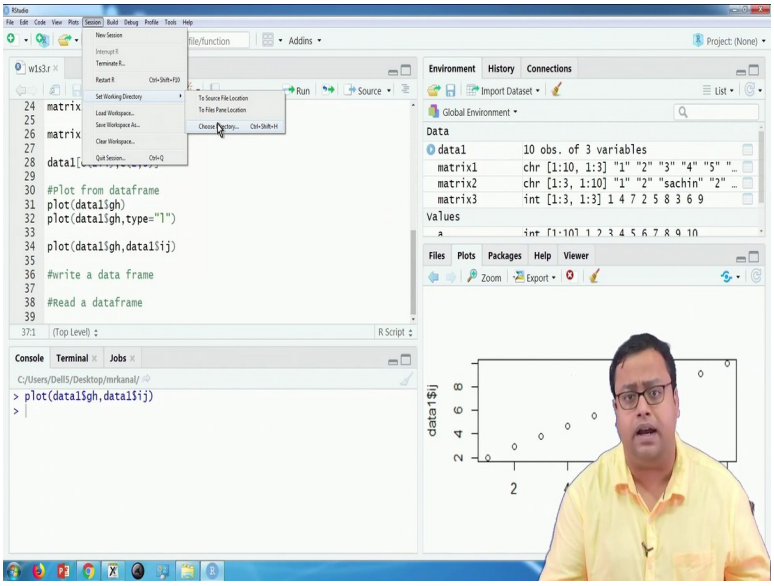


Sometimes we also want to get a scatter plot, so we will write plot let us say data dollar gh, data 1 dollar gh comma data 1 dollar ij and here I get a scatter plot, I get gh as x, ij as y and this is the scatter plot of that particular thing, and I can put it line plot as well.

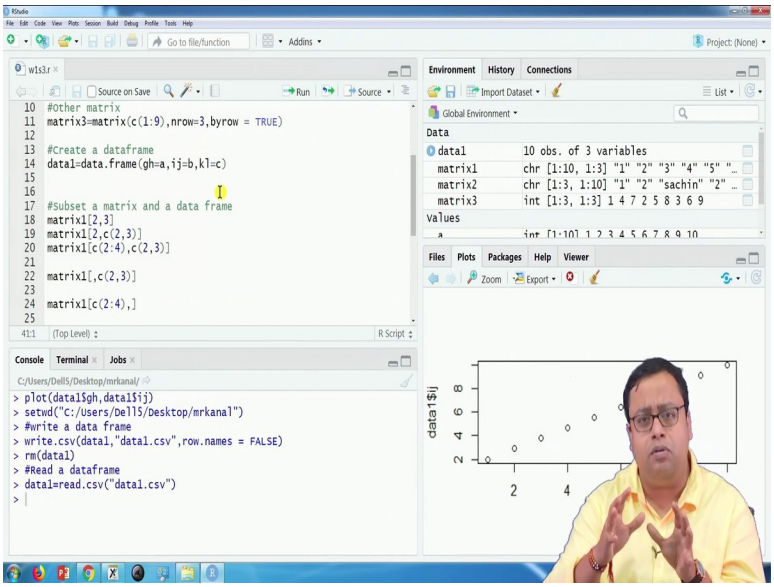
So these are there are various types of plots that can be plotted, all I am trying to say is that some basic plots you should know how to plot. So, for example, the bar chart is something that you should know, you should go and search how to do it, it should be done in introduction to business analytic kind of courses, how to plot bar chart, how to plot a box plot, how to plot a Pi chart these are some of the basic things that you can learn, and I will not go into that.

The next part I will just show you how to save a data and how to read a data, till now we have created a data on our own but sometimes we have to save our subset of the data that we have created, and sometimes we have to read our data set from somewhere it is saved. So this is something that we will show you. First, I will save a data whatever dataset, this data1 , I will save it and then I will read it. So again, good practice is that you put a working directory.

(Refer Slide Time: 26:13)



(Refer Slide Time: 26:59)



Sometimes we also want to get a scatter plot, so we will write plot, So you go to the session, set working directory, and you can choose any directory, directory means basically a folder in your

system. So you can actually choose any folder in your system by clicking on that, I personally try to keep my codes and the data set in the same place. The code, this is my personal opinion, you can choose any other directory as well.

But I generally put my code and the data file in the same place, so chose working directory to source file location, if you click that remember before clicking that everything has to be closed here, only this particular file has to be open and this has to be saved. After that, you go to sessions, set the working directory, to the source file location, wherever is the source location that means whatever W1S3 dot r is saved, the corresponding location is coming, and that is your working directory.

The working directory is a folder what the data sets and etc. can be stored, from where it can be read directly, it is placed where all your work is going on. After setting up the working directory, what we write is the, to write a data set, we have to write dot CSV, write dot CSV data 1 comma let us say data 1 dot CSV and row names is equal to false because we do not have any row names. So carefully see what did I write, the first syntax is the write dot CSV is the function name, the first syntax is the data set name that is there here data 1, the second syntax is whatever name you want to save it with data 1 dot CSV or whatever name you want to give that dot CSV.

And then row dot names are equal to false means I do not have any row names. I only have column names. So row dot names are equal to false I run this one. You will see in the folder there is a data 1 dot CSV here, I automatically saved, it was not there before, it got saved, it just got saved now. Now, if I want to read it back once more what I will do, I will first remove data 1 rm data 1, rm data 1 will remove data 1 from my memory, form my global environment, see there is no data 1 right now here.

I want to read it back, I write data 1 is equal to read dot CSV and then within quotes data 1 dot CSV, so this data 1 dot CSV is the name in which the CSV file is lying in your working folder, and this data 1 is the name you can give any other name is the name in which you want to store it in your memory, in your global environment. So this is if I run this one, it will read it from the same, remember before doing write or read whatever you have set working directory properly.

It should be that directory where the data file is lying, so data 1 is equal to read dot CSV data 1 dot CSV, and the moment I run this, this particular file is again read. So in this particular video, I have shown you how to create a matrix, how to create a data set, how to subset a matrix based on location, how to subset a data set based on location and based on conditions, how to read, how to write, how to create certain basic plots. Thank you I will actually continue with a little bit of data manipulation in the next video, thank you for being with me.