Data Analysis for Biologists
Professor Biplab Bose
Department of Biosciences and Bioengineering
Mehta Family School of Data Sciences and Artificial Intelligence
Indian Institute of Technology Guwahati
Lecture 16
Reading and Writing Data

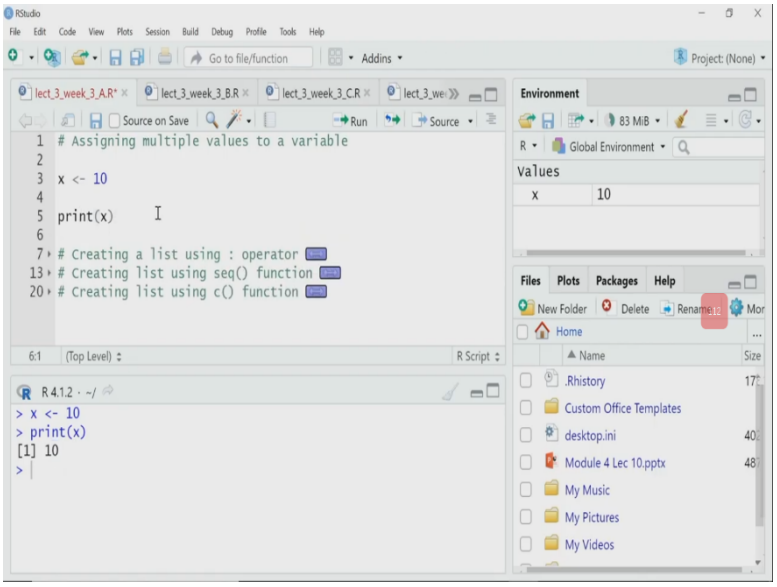
Hello everyone. Welcome back. For data analysis you need data. The data either should be in a file in a table format, data can have a complicated format also and sometime you may have to clean the data before you go into analysis. But, whatever it is, you need some data, either it is a table or a matrix or a simple list. So, in this lecture what we will discuss?

We will discuss how to read and write data. That is the key first step of any data analysis work that you have to perform. I will start with assigning list type data, vector type data to a variable. In one of our earlier lecture we have seen how we assign a numerical value to a variable.

Now, what I will do today first is that I will show, suppose you want to manually want to create a list of numbers. So, suppose you are doing, performing some calculation and there a list of numbers will be created and that will be assigned to a particular variable and then you will do some subsequent analysis. So, this list is equivalent to a vector also, you can call it a vector also.

So, we will start with that and then I will move into how to import or read a ready-made data file. Then we will show how to extract a part of that data. We will also learn how to write a data file. So, let us start it to do the whole work today I will use R studio but you can use native R also.

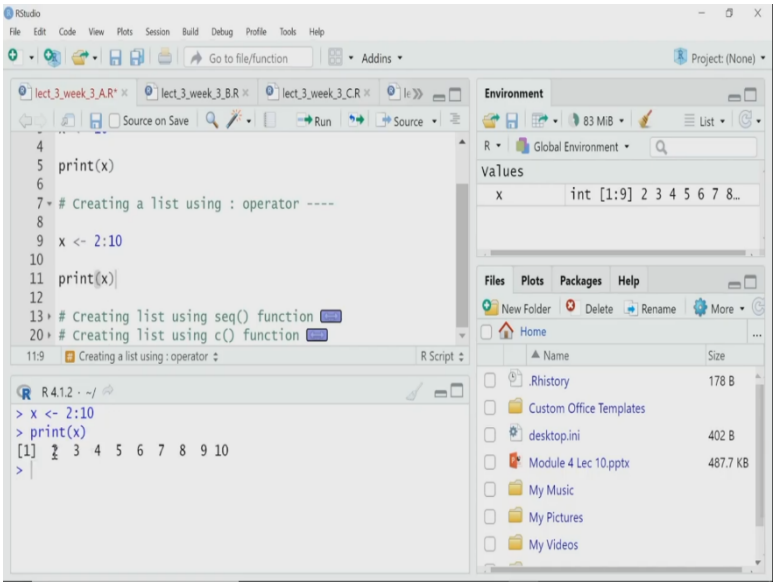
(Refer Slide Time: 02:04)



So, the first thing that we want to learn today is assigning multiple values to a variable. So, if you remember what we have done earlier, we have earlier said, ok, I can use this assignment symbol, this arrow(←), and then I can say, x ← 10 . So, I can do that. So, if I do that and then if I run it. Then x becomes 10.

You can see this environment pane here on the right hand side where it is written x = 10 or you can write also print(x) and then you can run it. It should say x = 10. So, this is the way, you actually assign a particular numerical value to a variable. But as I said today in the beginning that sometime you may have multiple values assigned to a variable. That means you want to create a vector. One column or you want to create a list. So, how should we do that? We will do that today using three methods.

(Refer Slide Time: 03:08)



The first one is we will use an operator, this colon(:) operator we'll use and then we will use a readymade function in R called a sequence function and then we will use a combination function called c. So, let us start with the first how to use colon operator to actually create a vector or list.

So, suppose I want to assign integer values 2 to 10. 2 to 10 to x. So, x is my variable and I want to assign 2 to 10. That means 2 3 4 5 6 up to 10 these values as a one-dimensional array or a vector or list to x, so that I can write in this way,

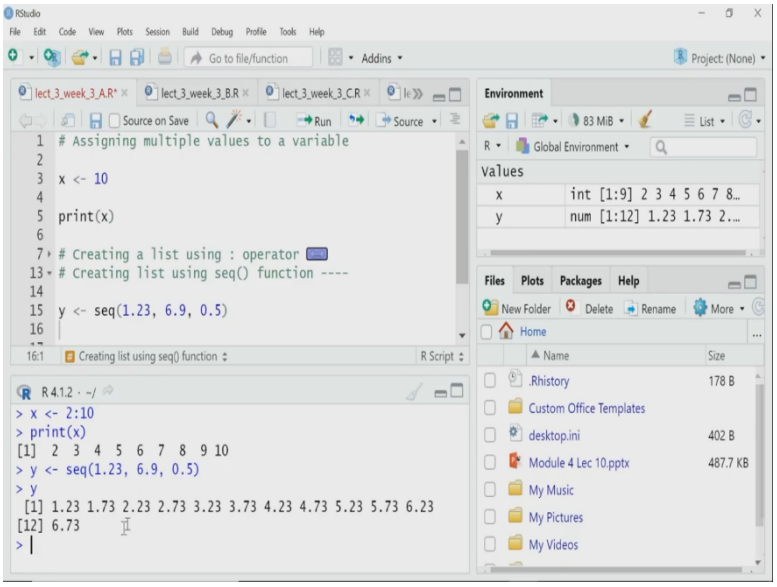
$$x \leftarrow 2:10$$

The lowest value 2 followed by a colon symbol and then the final value 10. So, let me run this. Let us see how what R does to that.

So, if I do on the environment pane, you can see it is saying that x is an integer, starting from, it has nine elements 1 to n and those values are 2 3 4 5 6 7 up to something like that. I can write here print(x), so that should print the value of x. So, it has printed. You can see we have integer value starting from 2 up to 10.

But, remember this colon operator that I have used, is used to represent a range and it is representing the integer range. So, I cannot say to create an array of numbers from 2.2 to 3.896 something like that, wherever the decimal values are there. So, to do that we have to use something called a sequence function. Let us go to that.

(Refer Slide Time: 04:57)



So, the sequence function is seq, you write,

$$Y \leftarrow seq(1.23, 6.9, 0.5)$$

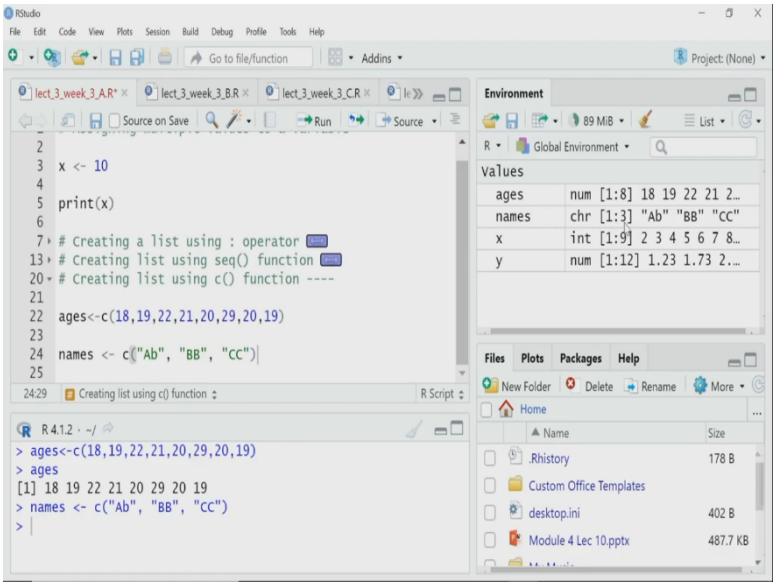
seq and then you put the round bracket, just like any other function. So, what I am doing here, I want to create a number sequence or rather a list or a vector, starting with 1.23 this is the lowest number and then I keep on increasing it by 0.5 and up to 6.9.

So, the sequence function seq take three arguments. The first argument is here, is the initial the lowest number or the first number. The second argument is the last number of the list and the third argument is the increment by which you want to increase it. So, what I am doing I am starting from 1.23 going up to 6.9 with increment of 0.5. Let me execute it.

Now, let me print(y). You can see y has these numbers. It has started from 1.23 because that is what I have asked for and then the value has got increased by 0.5. 1.23 plus 0.5 is 1.73 and then you get 2.23. So, the increment is by 0.5 and the last number is 6.73. This you should note very carefully. I have asked R to create a list up to 6.9.

But, it has not created up to 6.9. It has stopped at 6.73 because the increment I have asked is 0.5. So, if I if it has added 0.5 to 6.73 that should have crossed 6.9. But it is not supposed to cross 6.9 because the highest value, the last value I have said 6.9. So, it has stopped there. So, this is the way you can create a vector or a list of numbers and assign that to a particular variable, the way we have done it here for y.

(Refer Slide Time: 07:01)



Now, I come to the third option that you can use to create a list or array. And this is very beautiful, because in this case, not just number I can use characters also. For example, name of people, name of genes. You want to create a list of name of genes. How will you create that?

So, that is what we will do by using this c function, c with the round bracket. So, take the first example. So, what we I have shown here? Suppose, I want to create a list of numbers, which are ages of people. For example, 18, 19, suppose, I have I am collecting data from people in university and I have taken the age of each individual student. So, 18, 19, 22 and so on these are the ages of individual students. And I want to combine them together.

Remember, I have to combine them together to create a list or a vector. So, that is why the c function come, for combining. So, how I will write?

$$ages \leftarrow c(18,19,22,21,20,29,20,19)$$

I will write these numbers, numerical values that I want to combine as arguments inside the round bracket and I will write c before that because c is the name of the function. And then I use the assignment sign and I will assign this whole thing to my variable. I have given the name of the variable as ages.

So, let me run that. And now, if I want to know what I have got, I can check here in the environment pane. It is telling me that ages is a list of number. It has 8 numbers, 1 to 8 it is

shown in the square bracket. And the numbers are 18, 19, 20, 22, something like that. I can write here ages in the console and if I press enter then also all those eight numbers are printed. So, I have assigned a list of numbers to these variable ages.

Now, come the second part. I want to assign some characters, some names to or strings to a particular variable. For example, suppose, I have three genes and their names are Ab, BB and CC and I want to assign these names to a variable called names. So, again I will use the c function. So, c in the bracket I will write these names as arguments.

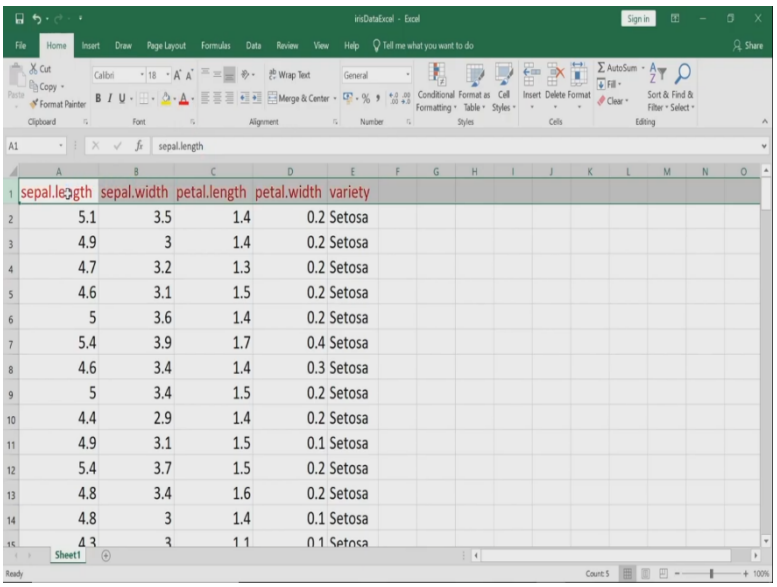
Names ← c(“Ab”, “BB”, “CC”)

But, be careful as these are not numbers, you have to put them inside the apostrophe. So, you have to this apostrophe ab then comma then the second character second argument, again a character within the apostrophe and so on. So, if I run this, I will create a variable, in the environment pane you can see, the variable, its name is name and what type of variable R is telling me that this is a character type, chr mean character type variable having three elements 1 to 3. And what are those? Ab, BB and CC.

So, this is very easy and this way you can actually create a list of numbers or characters which are very hand, will come very handy in our subsequent analysis. So, now, I will move into reading a data file. You are not writing the data directly in the inside the code or creating the list yourself but you are reading a ready-made data file.

Now, before I go into reading and show how I will read a data file and do extraction of data from that, let us discuss some crucial point in creating data tables. The best way, the good practice of storing data is to store them in a table format. And I will show one example to explain what do I mean by good practices.

(Refer Slide Time: 10:42)



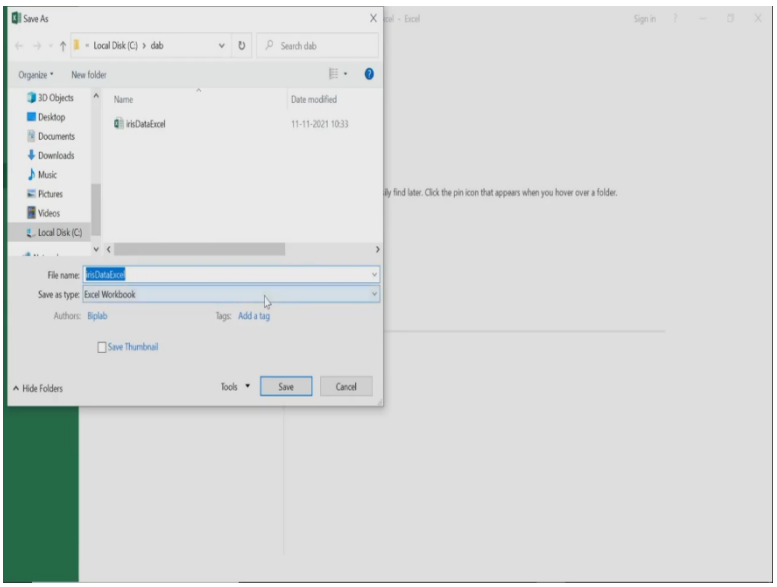
	A	B	C	D	E
1	sepal.length	sepal.width	petal.length	petal.width	variety
2	5.1	3.5	1.4	0.2	Setosa
3	4.9	3	1.4	0.2	Setosa
4	4.7	3.2	1.3	0.2	Setosa
5	4.6	3.1	1.5	0.2	Setosa
6	5	3.6	1.4	0.2	Setosa
7	5.4	3.9	1.7	0.4	Setosa
8	4.6	3.4	1.4	0.3	Setosa
9	5	3.4	1.5	0.2	Setosa
10	4.4	2.9	1.4	0.2	Setosa
11	4.9	3.1	1.5	0.1	Setosa
12	5.4	3.7	1.5	0.2	Setosa
13	4.8	3.4	1.6	0.2	Setosa
14	4.8	3	1.4	0.1	Setosa
15	4.3	3	1.1	0.1	Setosa

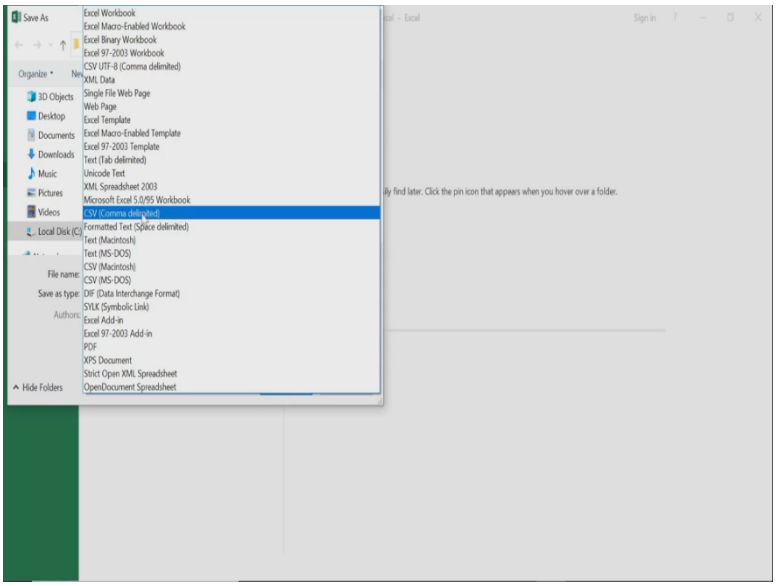
I have downloaded the iris flower data set which is very common in machine learning and I have created a excel sheet. Many of you doing experimental biology in lab, may have stored your data in excel. If you are storing the data in table format that way I have shown here, the first practice that you should always do, you should add a header line here. This line like let me highlight it with red, these are the headers, these are the name of the variables.

The first variable is sepal length, the last variable in the column e is variety. So, whenever you are storing data, please, put it, put a header and then arrange the data. If you keep the data in an organized fashion with a suitable variable names, then data analysis become very easy. You can actually import a excel file to R, but what I will show you in R studio here, I will import these same data, but not directly from excel file but from csv file Comma Separated Values formatted file.

So, this is csv file format is very common in data analysis. Many programs algorithms will output data in csv format. Many machines that you use in your lab may output data in csv format. You can open a csv file in excel. And what I will do I will create a csv file out of this excel file and then I will import that in my R studio. So, how can you create a csv file from excel sheet.

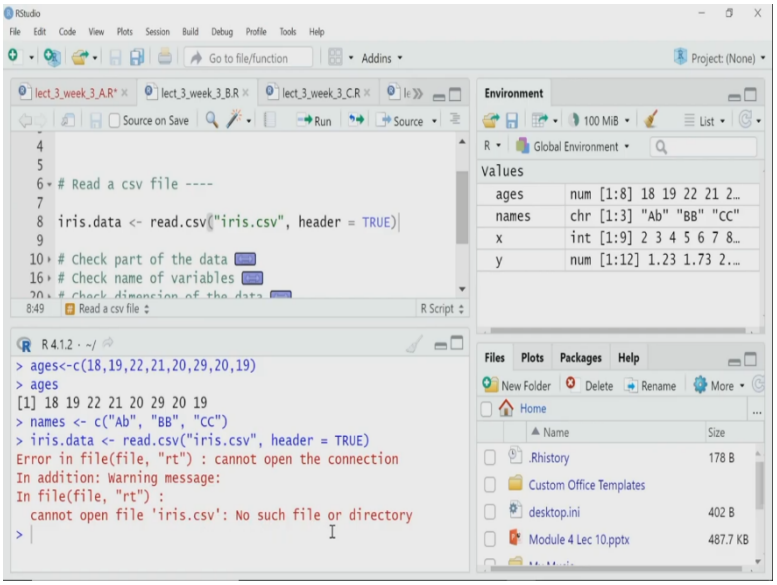
(Refer Slide Time: 12:27)





So, go to file section go to save as and then suppose, you want to go to a location where you want to save and then save as type you will find csv, this is written here csv comma delimited. You select it and save it. I have already saved it. So, I will not save it again but it is easy to do. So, now, let me go back to R and show you how I will import these csv files just that I have created earlier for iris flower data and extract some section of that data.

(Refer Slide Time: 12:50)



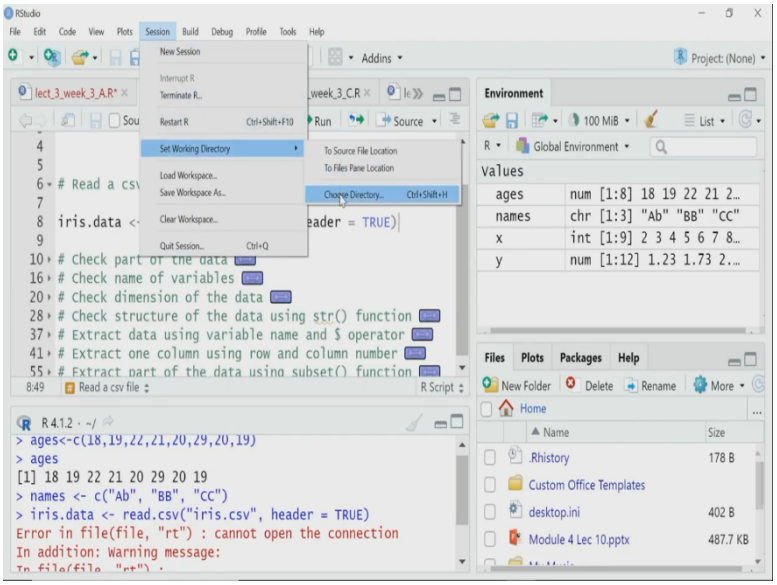
iris.data ← read.csv("iris.csv", header=TRUE)

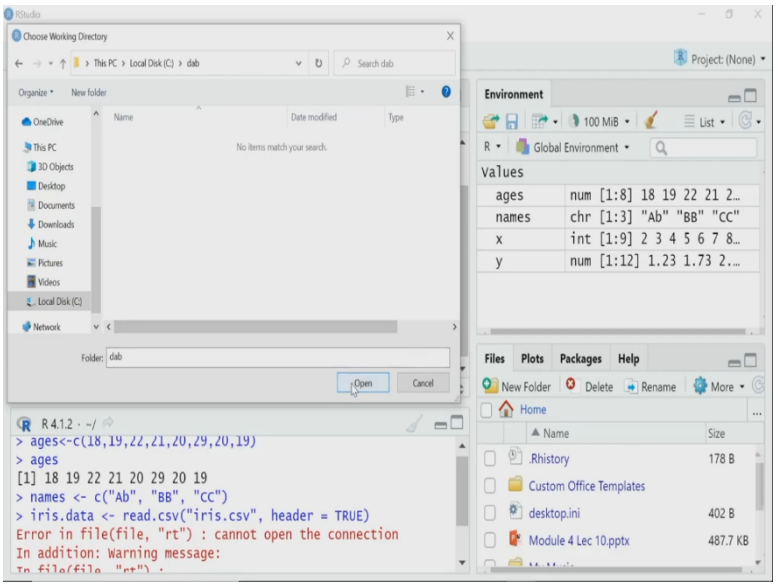
So, I want to read or import data from a csv file. And to do that what I will do? I will use a function called read csv, read.csv. This is a function in R and I will use that to read the csv file. So, read csv file, read.csv this function will take multiple argument which are present inside this round bracket. The first and foremost argument should be the name, name of the file that you want to read and that should be inside this apostrophe. Remember, it is the character, name is our characters. So, you have to put inside the apostrophe and do not forget to write this dot csv extension.

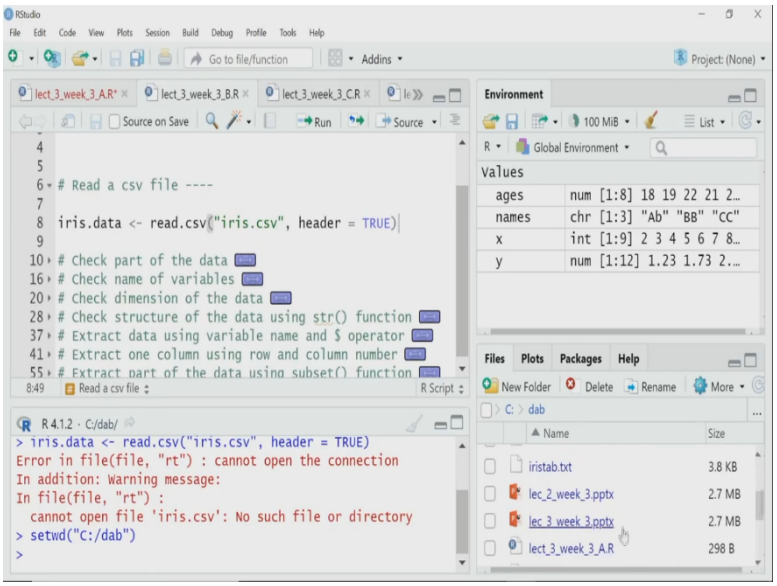
And then the second argument that I have used, I have written as header equal to TRUE. You could have avoided it but it is better to write it. That means I am telling R that explicitly that, ok, my data file has a header and read that header line as a header line to put the variable names. So, I will use this read dot csv to read this iris dot csv file. And then I will assign whatever I extract by reading or importing, it will be assigned to this variable called iris dot data.

So, let us do that. Ok I have got an error. I have got an error here in red color. R is saying that it cannot open the file iris.csv because no such file or directory exist, where I have done gone wrong? Okay, I am working in a folder or directory where that particular data file does not exist. That means I have to change my working directory. How can I do that?

(Refer Slide Time: 14:41)

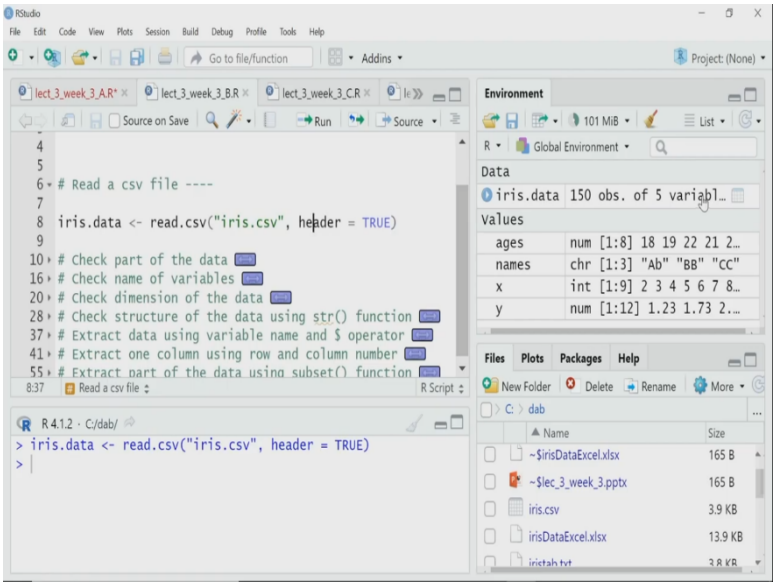

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
New Session
Interrupt R
Terminate R...
Restart R Ctrl+Shift+F10
Set Working Directory
Load Workspace...
Save Workspace As...
Clear Workspace...
Quit Session... Ctrl+Q
To Source File Location
To Files Pane Location
Choose Directory... Ctrl+Shift+H
Addins
Project: (None)
lect_3_week_3_A.R*
Run
Source
Environment
Global Environment
Values
ages num [1:8] 18 19 22 21 2...
names chr [1:3] "Ab" "BB" "CC"
x int [1:9] 2 3 4 5 6 7 8...
y num [1:12] 1.23 1.73 2....
Read a csv
iris.data <-
eader = TRUE)
# Check part of the data
# Check name of variables
# Check dimension of the data
# Check structure of the data using str() function
# Extract data using variable name and $ operator
# Extract one column using row and column number
# Extract part of the data using subset() function
8:49
Read a csv file
R Script
Files Plots Packages Help
New Folder Delete Rename More
Home
Name Size
.Rhistory 178 B
Custom Office Templates
desktop.ini 402 B
Module 4 Lec 10.pptx 487.7 KB
R 4.1.2 · ~/
> ages<-c(18,19,22,21,20,29,20,19)
> ages
[1] 18 19 22 21 20 29 20 19
> names <- c("Ab", "BB", "CC")
> iris.data <- read.csv("iris.csv", header = TRUE)
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:


Choose Working Directory
This PC > Local Disk (C) > dab
Search dab
Organize New folder
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C)
Network
Name Date modified Type
No items match your search.
Folder: dab
Open Cancel
RStudio
Project: (None)
Environment
Global Environment
Values
ages num [1:8] 18 19 22 21 2...
names chr [1:3] "Ab" "BB" "CC"
x int [1:9] 2 3 4 5 6 7 8...
y num [1:12] 1.23 1.73 2....
Files Plots Packages Help
New Folder Delete Rename More
Home
Name Size
.Rhistory 178 B
Custom Office Templates
desktop.ini 402 B
Module 4 Lec 10.pptx 487.7 KB
R 4.1.2 · ~/
> ages<-c(18,19,22,21,20,29,20,19)
> ages
[1] 18 19 22 21 20 29 20 19
> names <- c("Ab", "BB", "CC")
> iris.data <- read.csv("iris.csv", header = TRUE)
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
lect_3_week_3_A.R*
lect_3_week_3_B.R
lect_3_week_3_C.R
Source on Save
Run
Source
Environment
Global Environment
Values
ages num [1:8] 18 19 22 21 2...
names chr [1:3] "Ab" "BB" "CC"
x int [1:9] 2 3 4 5 6 7 8...
y num [1:12] 1.23 1.73 2....
# Read a csv file ----
iris.data <- read.csv("iris.csv", header = TRUE)
# Check part of the data
# Check name of variables
# Check dimension of the data
# Check structure of the data using str() function
# Extract data using variable name and $ operator
# Extract one column using row and column number
# Extract part of the data using subset() function
8:49
Read a csv file
R Script
Files Plots Packages Help
New Folder Delete Rename More
C: > dab
Name Size
iristab.txt 3.8 KB
lec_2_week_3.pptx 2.7 MB
lec_3_week_3.pptx 2.7 MB
lect_3_week_3_A.R 298 B
R 4.1.2 · C:/dab/
> iris.data <- read.csv("iris.csv", header = TRUE)
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
cannot open file 'iris.csv': No such file or directory
> setwd("C:/dab")

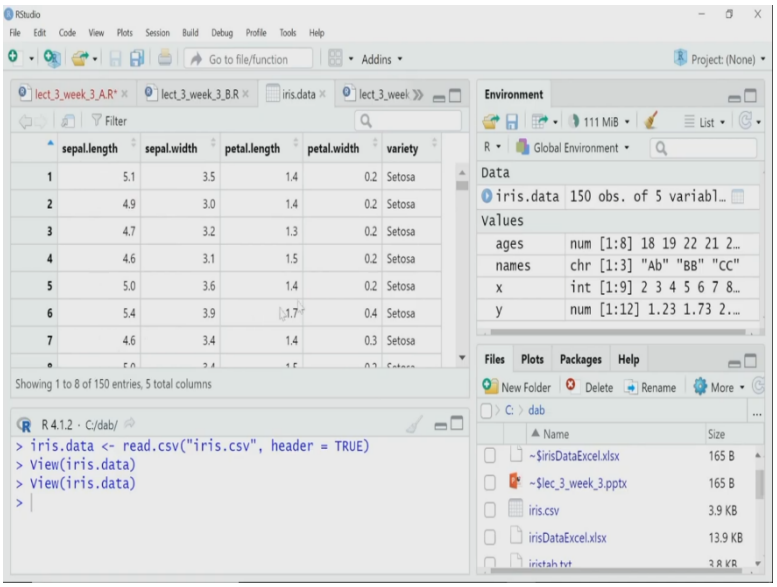
I can go to this session option in R studio, then to set working directory and then choose directory then I can choose where my data are. This is in this folder. So, I have changed my directory. Here in the file section now, you can see that directory or folder has opened and all the files are there. You can see here. I have this iris.csv file. Now, it should be able to read it. Again, let us call this read.csv to read iris.csv file.

(Refer Slide Time: 15:15)



Fine, it has read now. In the environment pane, you can see it has shown that in the data there is a iris.data variable. It has 150 observation and 5 variables. If I double click on this, that data will open.

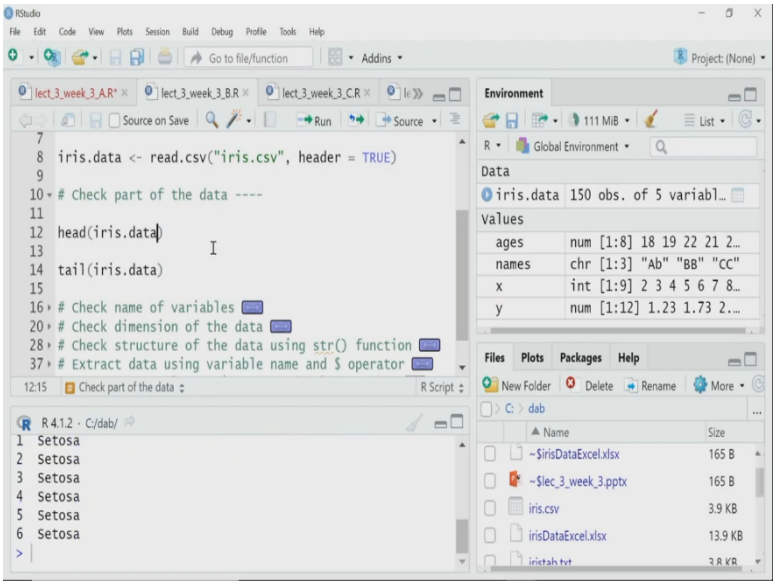
(Refer Slide Time: 15:30)



In this pane you can see, that data is in table format. It has been shown. The first variable is sepal.length. It is in the first column and so on. I have read the data. Now, I want to check the part of the data. When I am using R studio, it is very easy that I will double click on that data here in the environment, for example, like this and I can then scroll and check the whole data set.

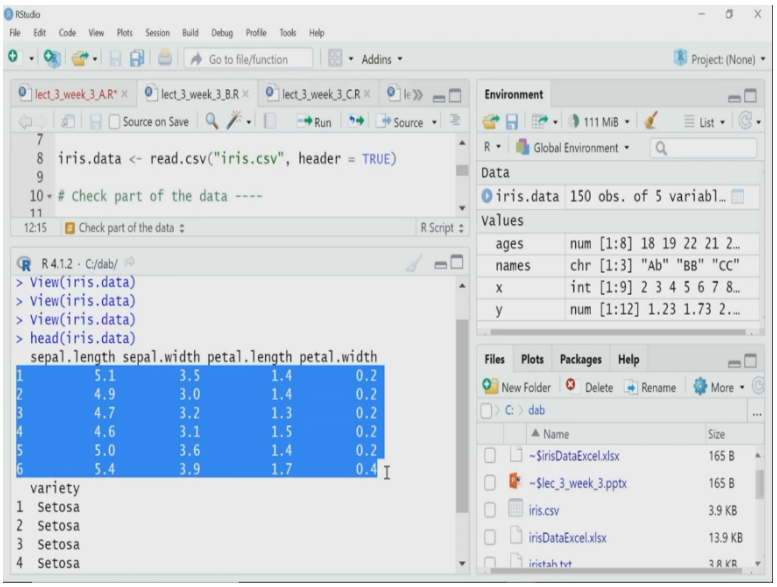
But, if you are using native R, in the R console, you have no option to open this whole data file or the data variable that you have created iris dot data that I have created now and see it. Because, it has 150 rows. So, whole console will get overflown by that data. So, that is cumbersome, so that is why there are some functions by which I can actually see or check part of the data and I will get an idea what data do I have in this file. So, for that I have some function for checking the data.

(Refer Slide Time: 16:25)



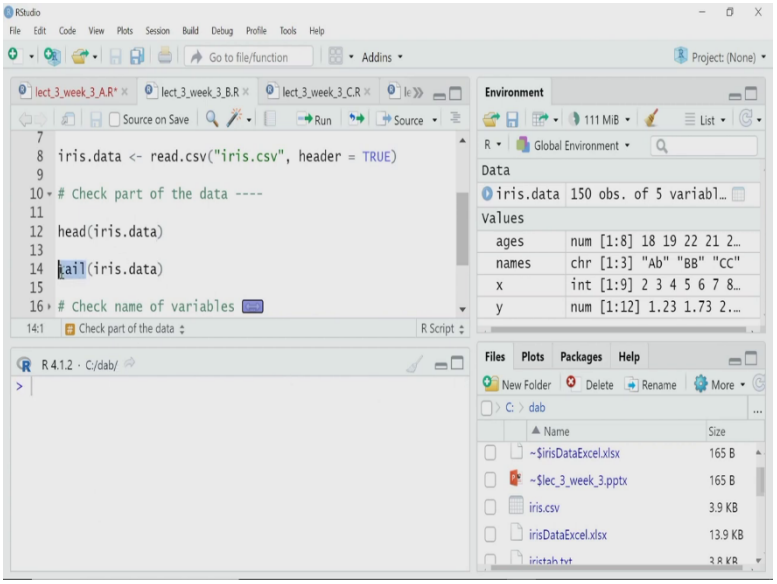
So, first function is head. Head function will show the upper part of your data file. So, if I say head(iris.data). This is my data variable that I have assigned.

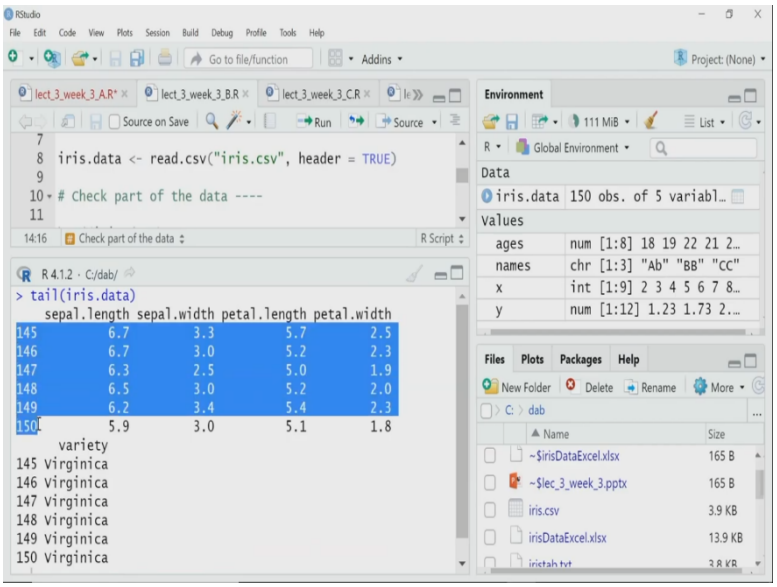
(Refer Slide Time: 16:44)



So, then it shows me here in the console, you can see, it shows the upper part only, up to 6 rows, 1, 2, up to 6. So, I know now, it has these variables. The name of the variables are sepal.length, petal.length and so on. And this is the head portion, the top most portion of my data that I have imported.

(Refer Slide Time: 17:09)



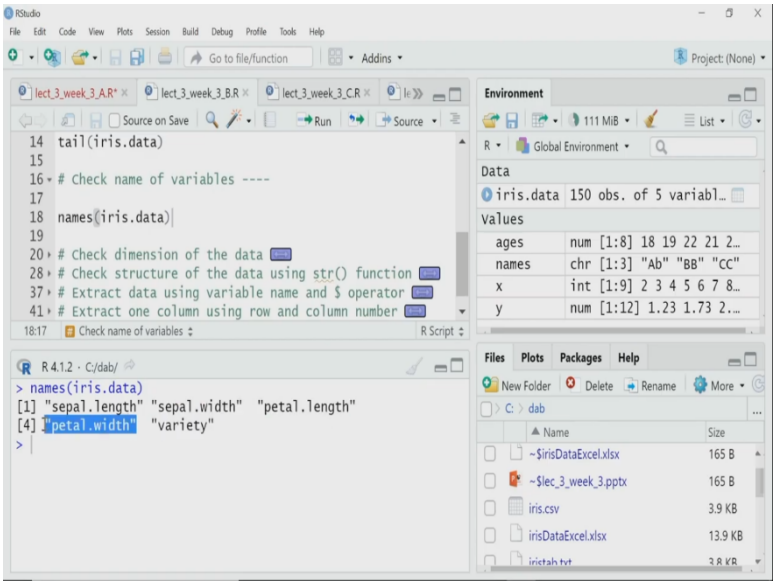


Now, let me go to the other option. Suppose, I want to look into the tail part, in the lower part of my data. So, I will use then the tail function and again I will put iris dot data the variable that where I have assign the whole data after reading from my csv file as an argument and I will run that. Then it should show me the lower part of that.

tail(iris.data)

Here, you can see easily that it is showing the last 6 line, starting from 145 to 150, because this data set has 150 rows or 150 lines and 5 columns. You can see the column headings also. So, this is very handy way to check part of the data start and end, so that you can understand what type of data you have in this file.

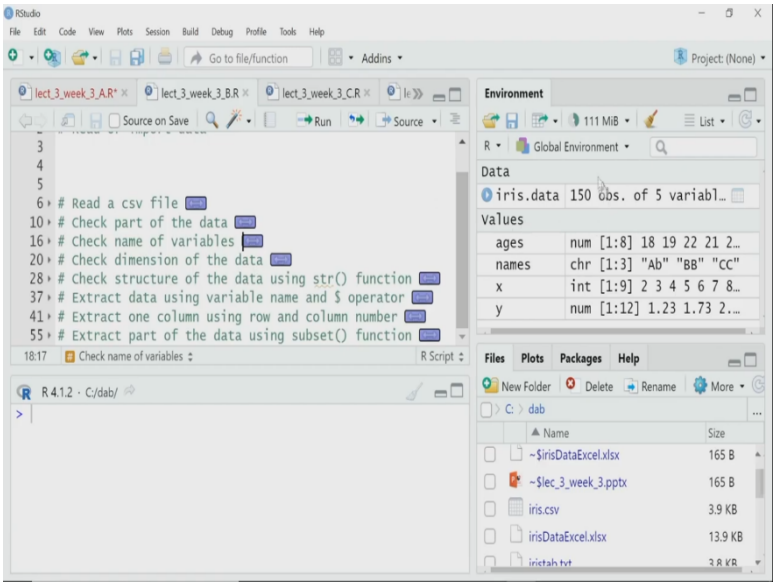
(Refer Slide Time: 17:56)



This brings us to another variable call names, which is very useful for us. By names, I want to know what are the names of the variable in my data set. And if you remember, I said at the very beginning that when you are creating a data table, please, give the names of the variable as header.

So, this name function is actually reading the header. So, the argument is obviously the name of the variable where the data has been imported that is iris.data. So, if I call names function for iris.data then it tells list me all the name of the variables, variables of all the names. So, I have five variables and it has listed them as names. So, what I have done?

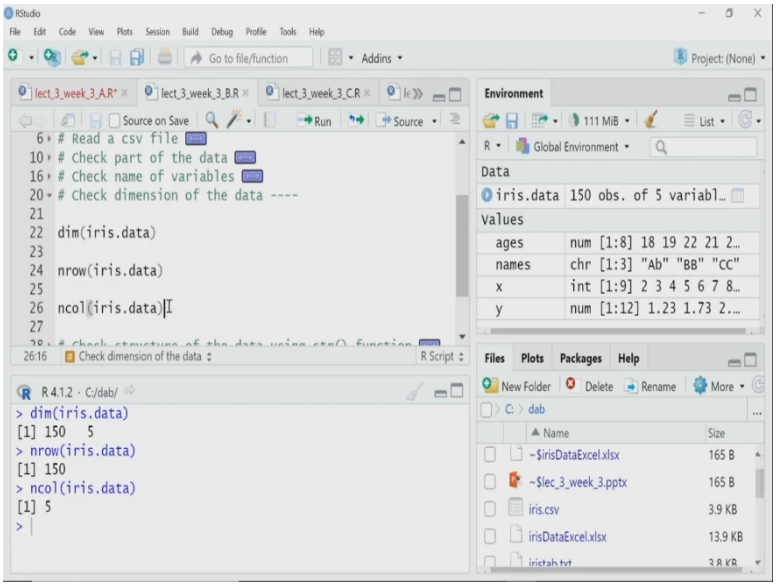
(Refer Slide Time: 18:42)



I have imported my data. I have checked the part of the data. I have checked the name of the variables. So, now, I want to know the dimension of the data. So, obviously, if you look into this environment part, it is already saying that iris dot data has 150 observation and 5 variable. But, suppose, you are not using R studio, you are using R, so that in that case, you need some way to know the dimension of your data set.

That means you want to know how many rows and how many columns are there in this table. Also sometime it may require that you want to use row numbers, number of rows in your data as a variable and use that variable in some calculation. For that also you have to use some function to do that.

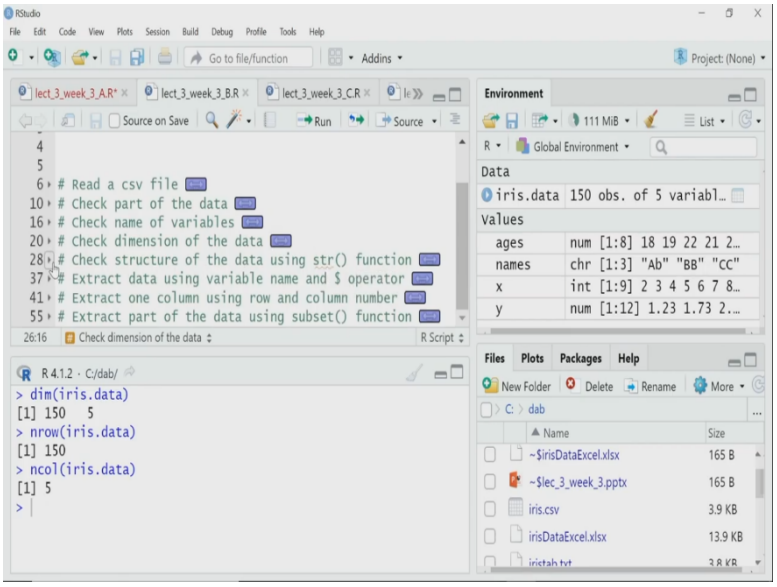
(Refer Slide Time: 19:34)



The function for that is dim. So, if I write dim(iris.data), it should give me the dimension. So, dimension is 150 by 5. Then you can ask number of rows in nrow function to get the number of rows in your data. So, that is 150 then you can use ncol, the number of columns in your data that function ncol function to get number of column in your data and that should be 5 in this case. So, this is the way you can check the dimensions of your table, tabular data.

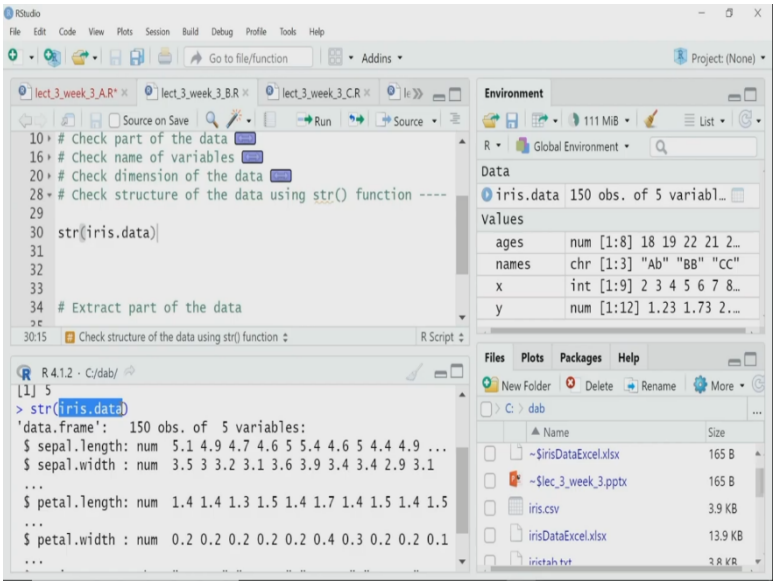
Now, what I have done till now, I have read the data from csv file, imported it and stored in a variable iris dot data. And now, I want to extract a part of that data. I want to extract a part of that data and assign it to a new variable because I want to analyze that section of that data. What do I mean by part? I mean a particular column or data for a particular variable or two variables. How can I do that? I can do that in multiple way.

(Refer Slide Time: 20:40)



So, one way of doing that is that you can use this operator. This is the dollar($) sign and this operator can be used to extract a particular object or variables information from a data. Whereas, I can use row number, column numbers, because your table has rows and column, we can use that. I can also use this subset function which is very powerful and I will show that.

(Refer Slide Time: 21:08)

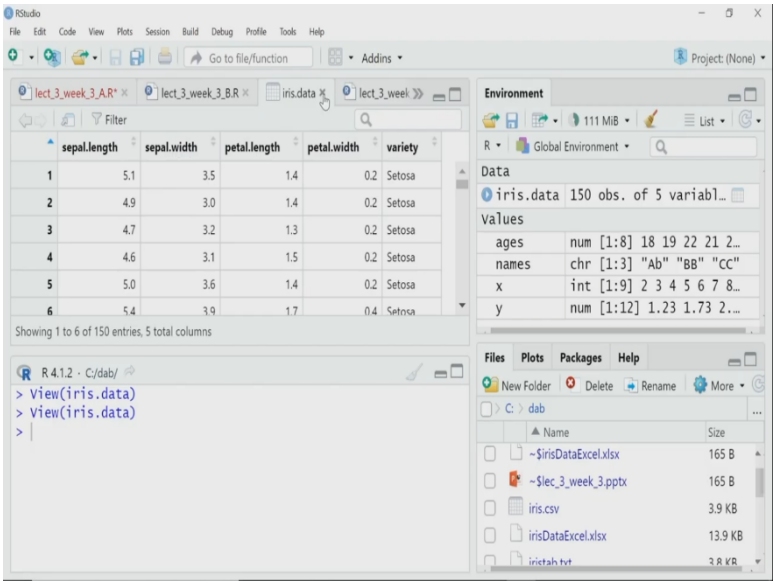


But, before I jump into it, another function which is very useful is called structure function. That gives me details about structure of the data. So, its function is, str, structure and then you

have to give the name of the variable where the data is stored, iris.data as an argument. So, if I run that str(iris.data), it gives me details which are very useful in data extraction.

So, it says that it is a data frame. So, iris.data is a data frame. Data frame is a type of table in R. It has 150 observation 5 variable and it has listed all the variable name here, sepal dot length, sepal dot width and that so on. And remember, it is using a $ sign before it. So, this $ sign with the variable name I can use to extract data for that particular variable and I will show that.

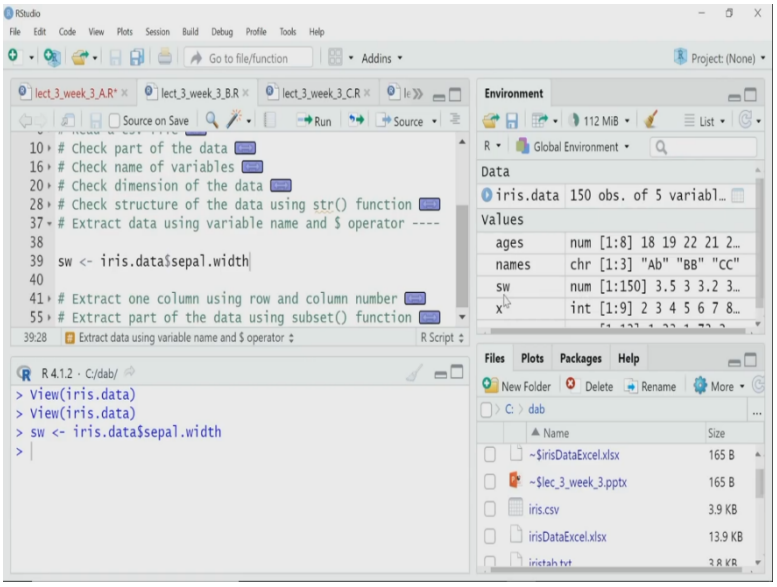
(Refer Slide Time: 22:02)



So, the first thing I will do I will extract a part of the data using that $ sign operator. So, what I want to do? I want only the data for sepal.width variable, width of the sepals. So, if I open the data here, see second column, the second column of my table has the data for sepal.width.

And I want to extract that part of the data from iris.data variable and assign that to a new variable sw. So, how I have written iris.data then the operator sign, the dollar sign and then name of the variable, name of the object. That is in this iris.data variable has and then I use the assign symbol to assign this whole thing to a new variable sw. So, let me execute that.

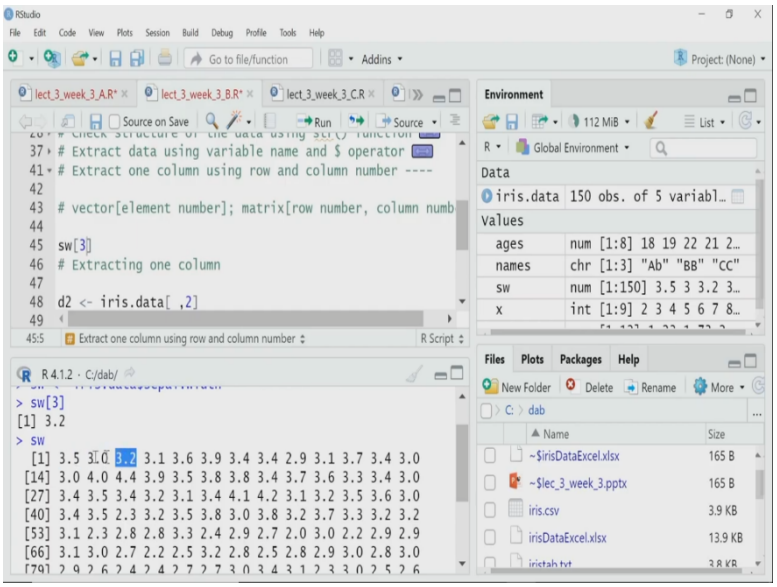
(Refer Slide Time: 22:52)



Now, here in the environment section, you can see, a sw variable has been created and R is saying, it is a number. It has 150 element starting from 3.5, 3, 3.2, so on. So, this is the way I have extracted the second column width where the data for sepal width is stored and I have created a new variable.

Now, I can do any operation on that new variable sw. I can extract the same thing in a different way, specifying the column or row number. How should I do that? So, before I do that, I explain that how we should use the column number, row number to extract a part of a data. Let us understand how we write the row number and column number for a vector.

(Refer Slide Time: 23:42)



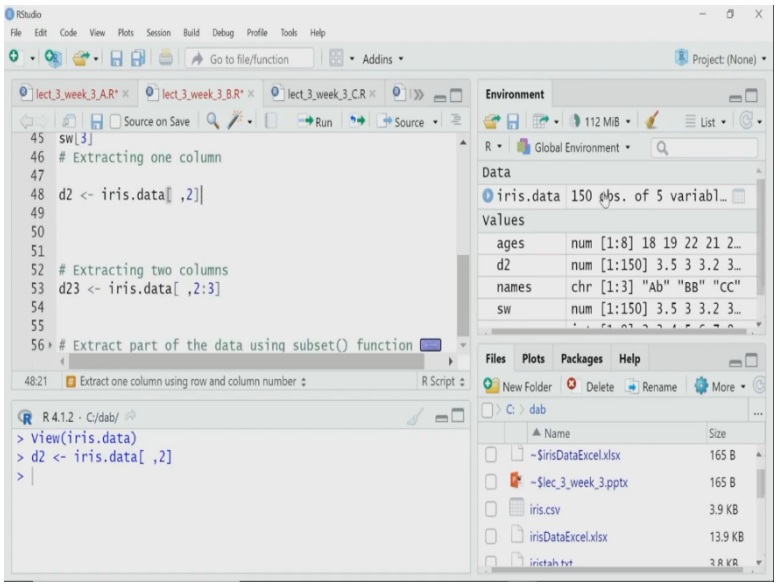
So, if I have a vector which is nothing but a list of numbers suppose, 10 numbers are there. So, the first number is 1, second number is the 2 and so on. So, you have, these are the elements. So, you use the, put the name of the vector and then you put a square bracket and in the square bracket you put the number of that that particular element position.

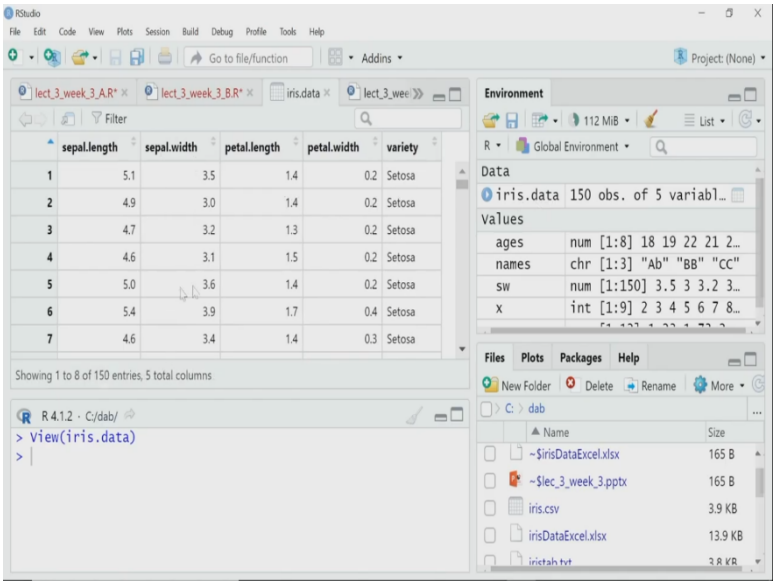
So, it is the first element, you should put 1, if is the second element, you put 2, something like that. For a matrix, you have row number and column number. So, you will put the name of the matrix then inside the square bracket, you write row number and column number. Let me write down to some example.

For example, I have sw is a vector. Just now we have created that vector, is the list. And if I put a square bracket and then if I say I want the third element and then. So, I will write 3 inside the square bracket and then if I execute it, it should give me 3.2. Let me open; let me see the whole sw data. See the third data for sw, the first one is 3.5, second is 3, the third one is 3.2.

So, when I written, when I have written s[3], the third, value of the third element has been taken. So, this is the way you assign the element number. In case of a matrix, what I we have to do? I have to specify the row number as well as the column numbers.

(Refer Slide Time: 25:19)





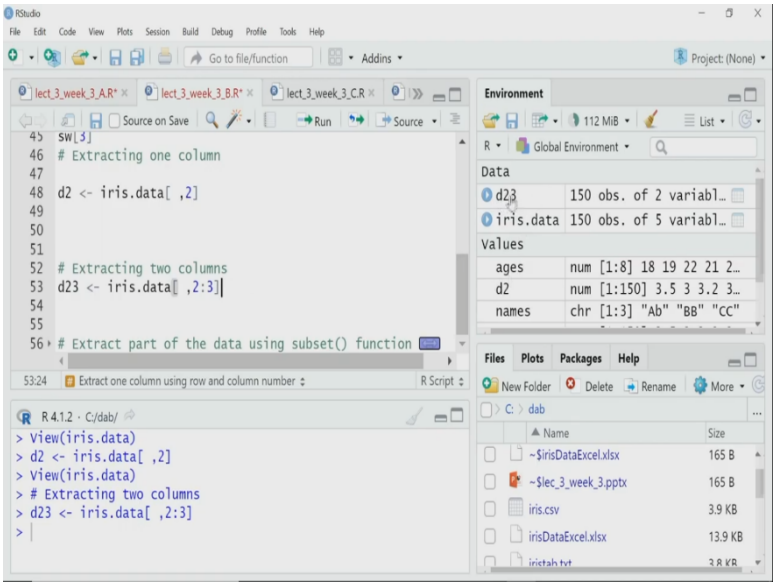
Now, in my iris data, I want to extract the second column. So, what is my iris data? Second column is this one, sepal width. I want to extract this whole column. That means I want all the rows and only that column, second column. So, what I am doing?

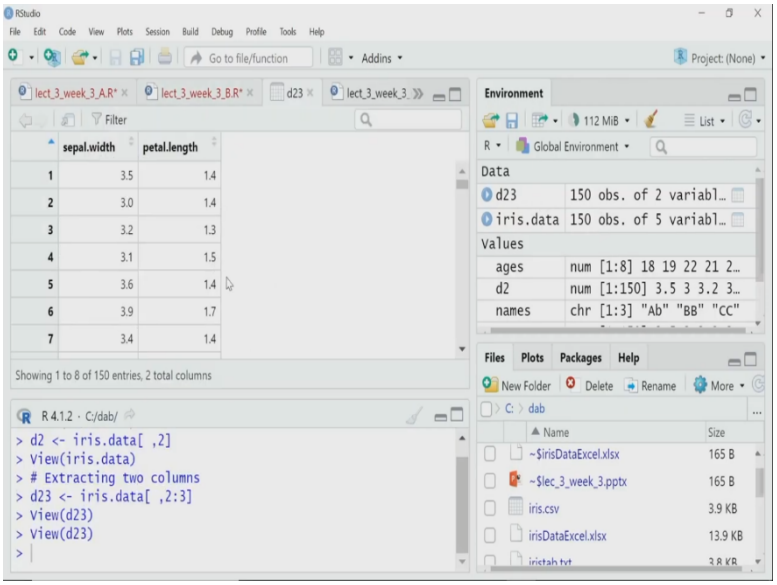
$$d2 \leftarrow iris.data[\ ,\ 2]$$

I am writing iris dot data, that is the variable name, name of the table. And then in the square bracket I am writing 2, before that I am putting a comma because first element should be here, the number of the row, row number and comma and the column number. But, I want all the rows that is why I have left this row number empty and then I put a comma and the column number. So, what I want?

I want R to extract the second column value for all rows and assign that to this new variable d2. Here, I execute and in the environment pane you can see, a d2 variable has been created which is list of numbers having 150 element starting from 3.5 then 3 and so on. Now, suppose, I want to extract two columns. Suppose, I want to extract two columns means sepal width and petal length for all the rows. So, if I again I will use that row number matrix and the column number option. So, what I will do here?

(Refer Slide Time: 26:50)





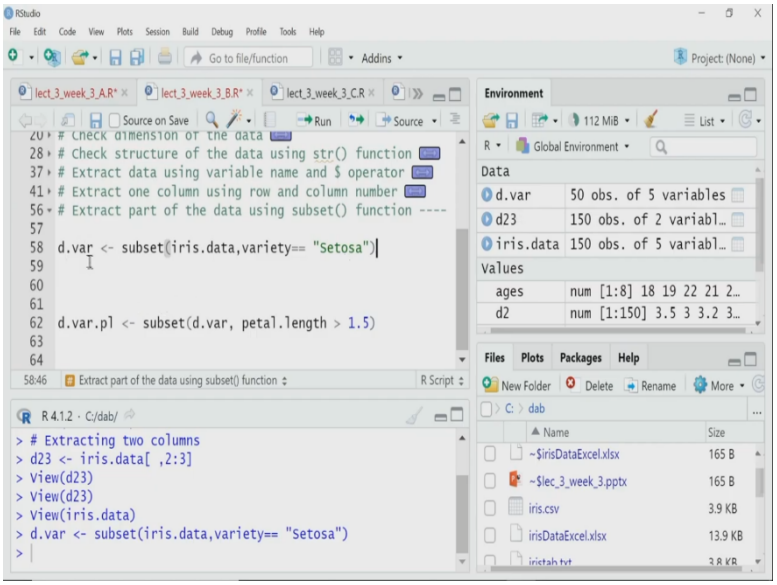
Iris.data is the name of the variable; from which I want to extract is the name of the table. And then inside the square bracket the first thing should be row number. But, I have left row number empty because I want all the rows to be extracted then comma and then I use the range option, the colon operator, 2 to 3. So, I want R to extract all the rows for column 2 to 3 and assign that to new variable d23.

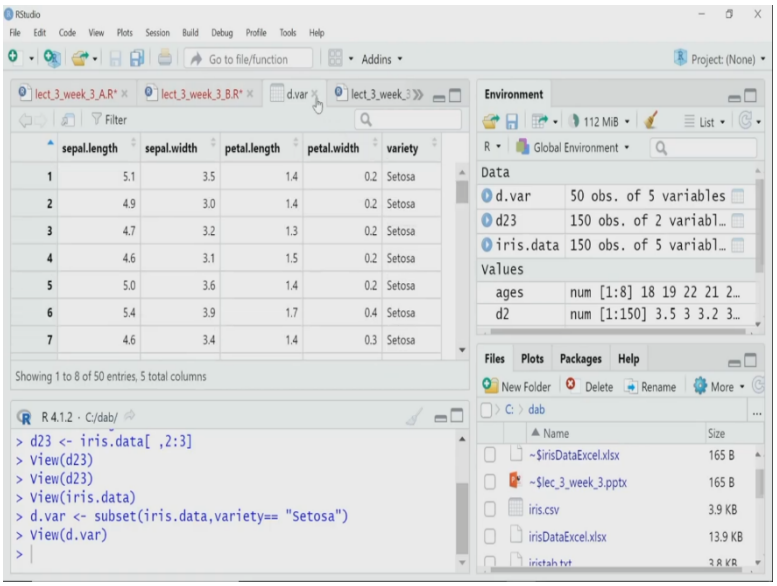
d23 ← iris.data[,2:3]

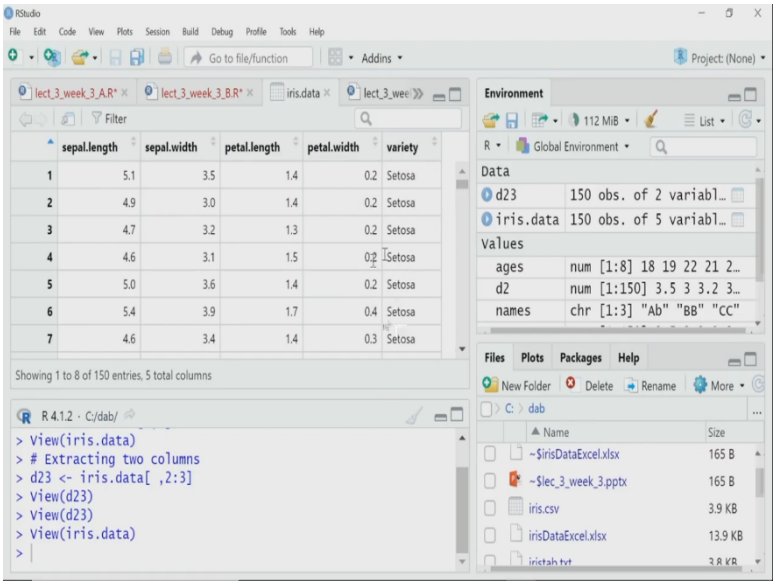
Here, I execute it. Now, you see, I have got a data which is actually a matrix because it has 2 variables and 150 observation. It has 2 columns 150 rows, if I double click it, here, you can

see, it is a table. So, it is actually a matrix. This brings me to the third option of extracting data. And I will use the subset function which is very powerful.

(Refer Slide Time: 27:55)







d.var ← subset(iris.data, variety == “Setosa”)

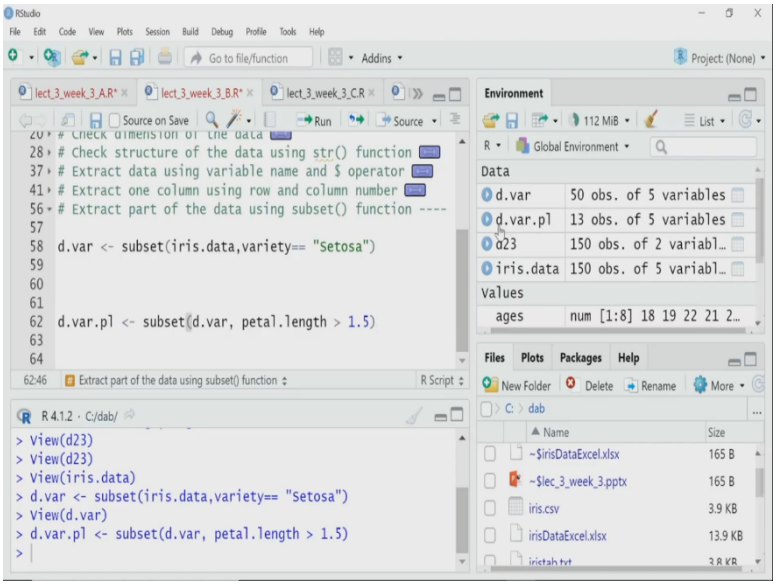
As the name suggests, subset function will extract a subset of the data. So, subset function, it will take some argument. The first argument will be, obviously, the name of the variable of the data frame or the table from where you are extracting a subset. And then interestingly here, I am not specifying any column number or row number, what I am specifying, I am specifying a logic operation. What is the logic operation?

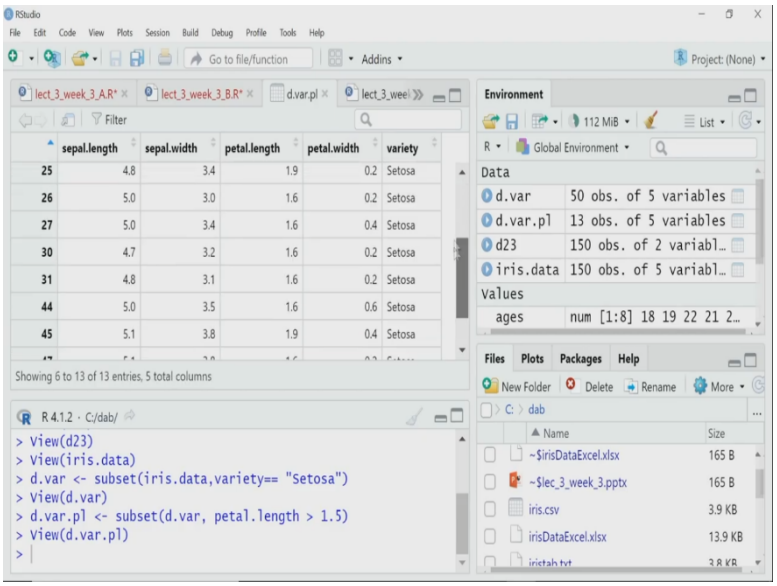
If you remember my data set has five variable names, five types of variables. One variable name is variety. So, I want that variety has to be equal to Setosa, so that is why I have used an equal to operator. So, I want, let me open the original data set to explain, I want the data, those rows from this data where the variety is Setosa, nothing else.

There are other variety of flowers also, I do not want that. So, I only you want the data, the full data set only for the variety Setosa. So, what I am doing? I am specifying here using this logic operation that the variety should be, the variety variable should be equal to Setosa. Let me execute that.

So, now, it has created a new variable d.var because I have assigned that to d.var. If I open it, you can see, it has 50 observations. You go down; you can see you have 50 observations. For all observations variety is Setosa. So, I have taken a subset. I have three four different flower types data; from that I have extracted data for the variety Setosa. Now, I can create a subset of this subset also.

(Refer Slide Time: 29:41)





d.var.pl ← subset(d.var, petal.length > 1.5)

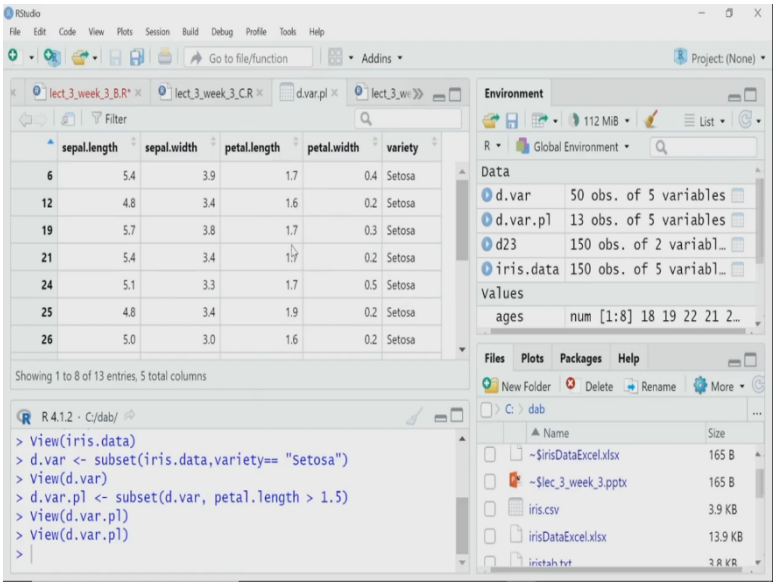
That is what I will show you here. So, now, I want that from this subset, because the subset data is in d.variable, from this d.var subsetted data, this subset data, I want to extract only those data's part where the petal length, the petal length is bigger than 1.5. Again I am using the subset function. So, subset, name of the variable is the first argument.

Now, name of the variable is not the original variable. It is the d dot var because already I have created this where only the Setosa data is there. So, d.var comma again I use a logical operation here. The logic operation here is that the petal length variable should be greater than 1.5.

So, if I run this, I get another variable that I have assigned d.var.pl and it has 13 observation for 5 variable. You see the 13 observation here. And you can see the petal length for all of these are bigger than 1.5. So, this subset function is very powerful. Using this we can actually extract one section of the data set using logic operation.

Then again we can implement the subset and by step by step, I can zoom into a particular part of my data set. So, I have extracted the data. I have started with reading the data file, extracting the data file. Now, you must be wondering, how can I write that data file, maybe as a csv file. I will go for that.

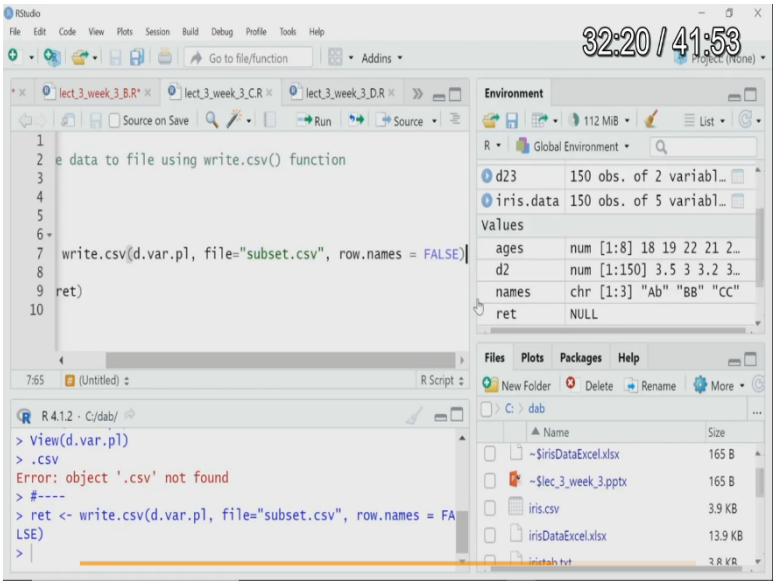
(Refer Slide Time: 31:16)



write.csv(d.var.pl, file = "subset.csv", row.names = FALSE)

So, now, I will learn how to write a csv file for a particular data. So, I will use write.csv function for that. And what I will write? Just now, I have created a data d.var.pl where you have data for variety Setosa and the petal length is bigger than 1.5 and I want to write it as a file, output file. I want to store it in my computer.

(Refer Slide Time: 31:40)

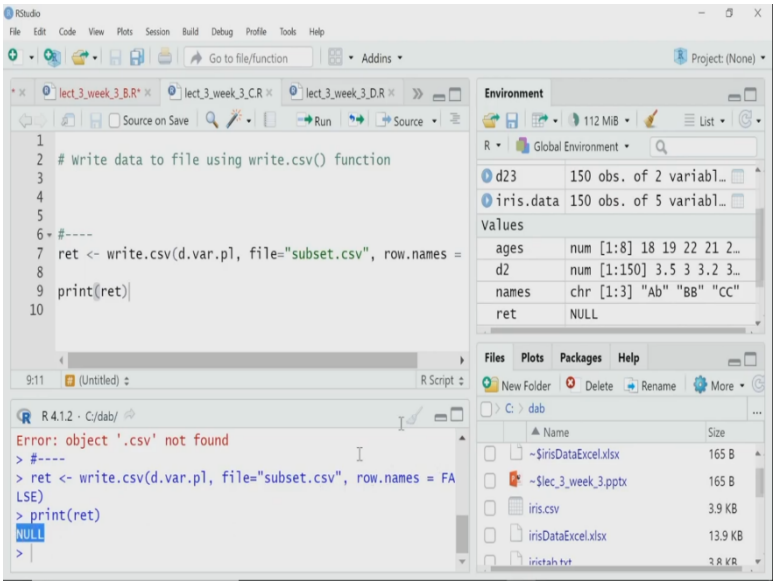


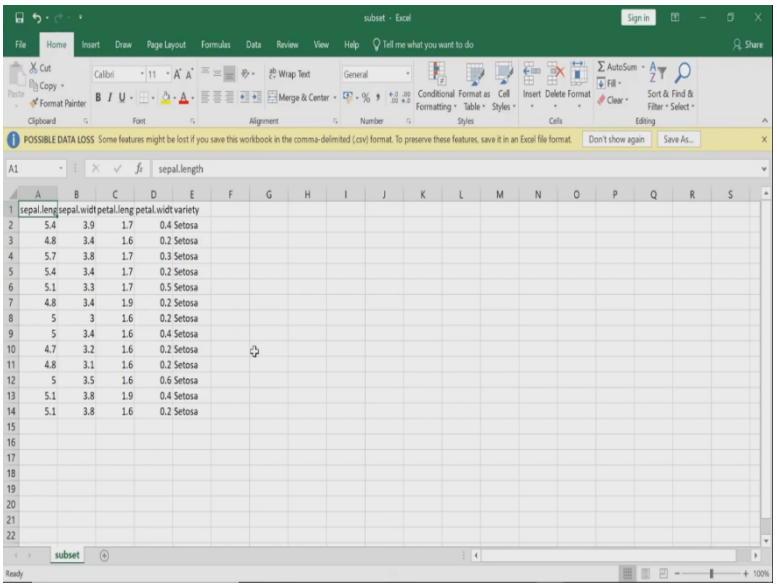
So, I will use the write.csv function. I will use the write.csv function. write.csv function will take the name of the variable as the first argument that is d.var.pl then I will put the name of

the output file where the file data will be written. So, it will be file equal to, I have given a name, subset.csv.

Again the name of the file is within the apostrophe and do not forget to write the name of the extension dot csv and then I have given written row.names as an argument and I have assigned that as FALSE. I will explain why I am doing it. So, let me run this first. So, that has been run and a new variable ret has been created. But the file has not been created yet. So, what I have to do? I have to call the print function to write that file.

(Refer Slide Time: 32:35)

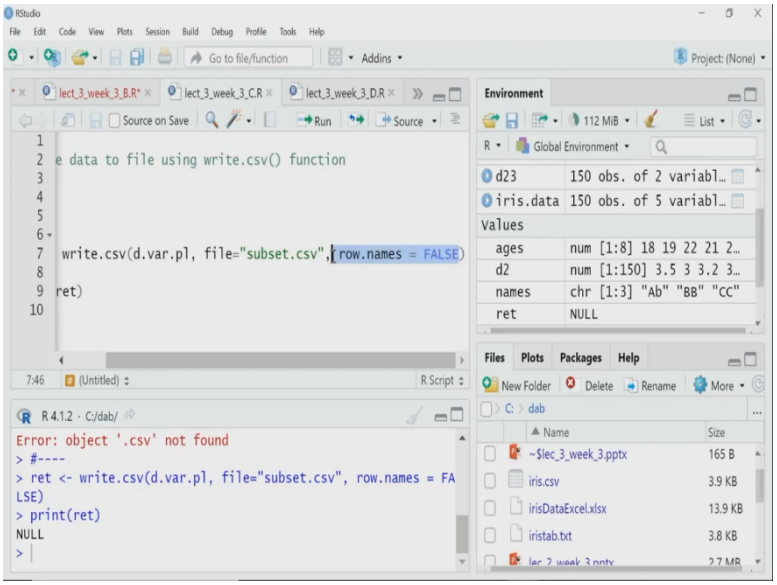


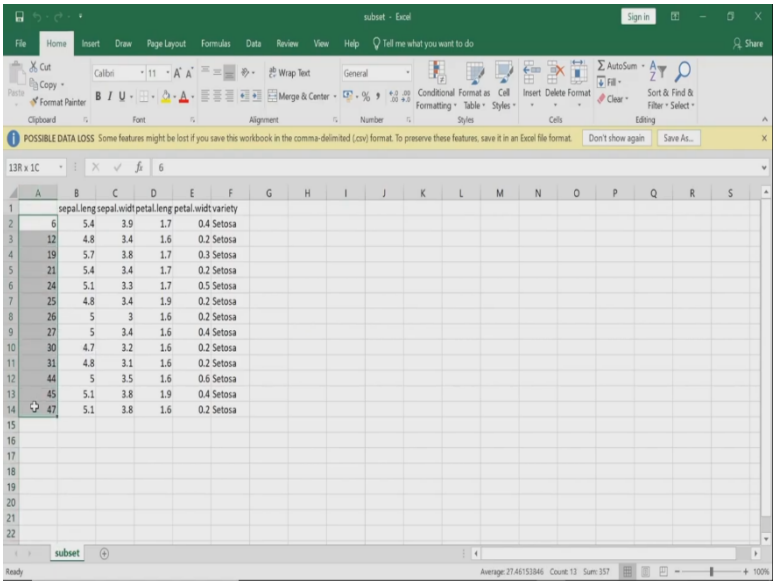


So, you have to write print and you have to give ret as the argument because I have assigned this write.csv output to this ret variable. So, I will run this print(ret) function. And I should

get this null output and I can go back and check in my folder whether I have got that file or not. So, here is that subset.csv. I am opening it and you can see that data has been written. I have for 13 rows. The first row is the header.

(Refer Slide Time: 33:16)



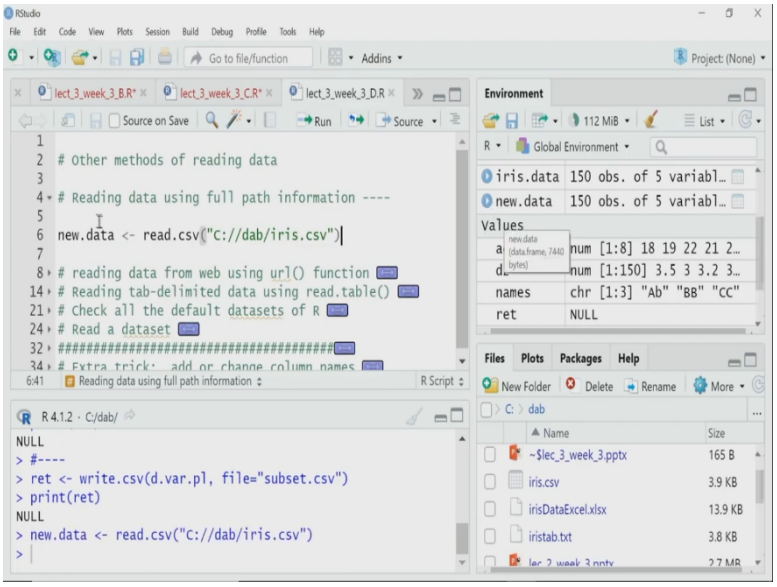


So, now, I will remove this row name argument. And I will run it again. So, again the ret variable is created. I will print that print ret. So, null output has come that means the file has been written. Let me go back to the folder. Here is the subset file written.

Now, you see what is there in the first column. The first column is the row numbers for these data set in the original file. But you may not need that. If you need that then you should not keep that row dot names option there, otherwise, it is better to remove it by setting that

argument equal to FALSE. Then it will not write the row numbers. Suppose, you want to read the data using the full file path name.

(Refer Slide Time: 34:10)



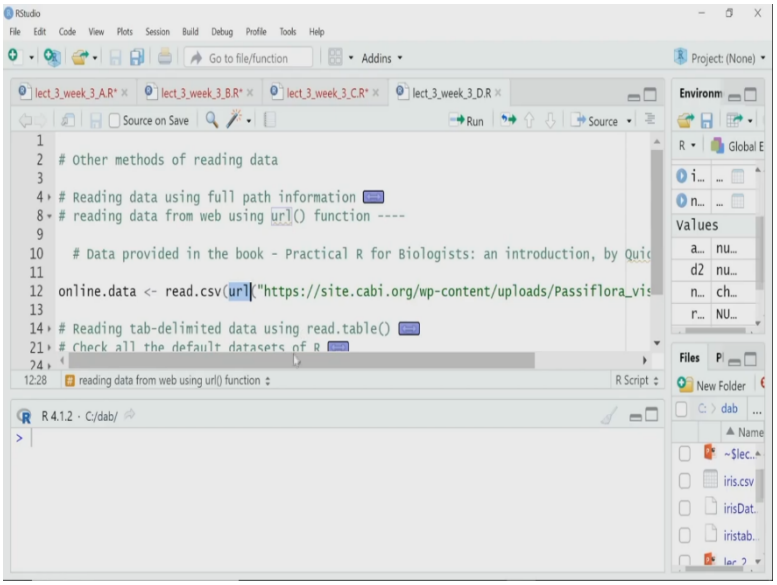
For example, what I have shown here I want to use read csv but as an argument I want to give the whole path of that particular file. For example, here

new.data ← read.csv("C://dab/iris.csv")

So, a new data has been created. You can check. It has the same data file.

Now, suppose, you have a data file, may be a csv file in the internet but you do not want to download it and you just want to directly import that data in your R console and work on it. So, that is also possible using the read csv function. You can easily do that. Let me show you how to write that. So, in that case, what I will do? Let me expand it.

(Refer Slide Time: 34:57)

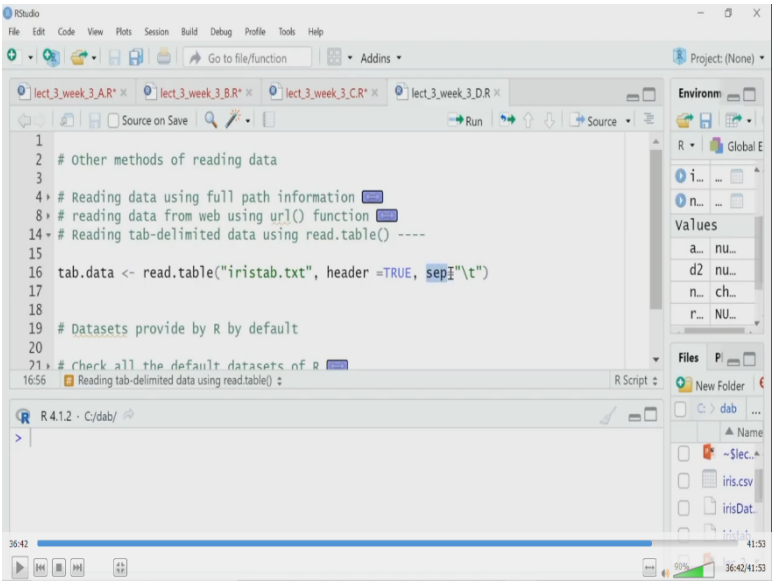


Online.data ← read.csv(url())

So, again I will use the read csv function. Now, as an argument I will write url and then url itself is a function. And as argument of that url function, I will write down this whole url the link that you have got for that data file. And then this whole thing will be assigned to a variable.

For example, I have given here the name online dot data and if you run that, this whole data will be fetched and it will be assigned to this online dot data variable. Then you can do analysis or extraction or writing the file, the way we have discussed now. I have till now, discuss only about csv files. You may be wondering why only csv file, there are many other data types also, data file formats also. One of the commonest one is tab delimited file. So, tab delimited file can also be read as a table format. So, what I will do?

(Refer Slide Time: 35:51)



I will use the read table function for that. So, the just like read.csv you can use read.table. So, here read.table is the function. It takes the first argument as the, first argument of this function is the name of the file. And I have a file here which is iristab, means I have saved that same excel file also as a tab delimited text file and that is why I have the extension .text, txt.

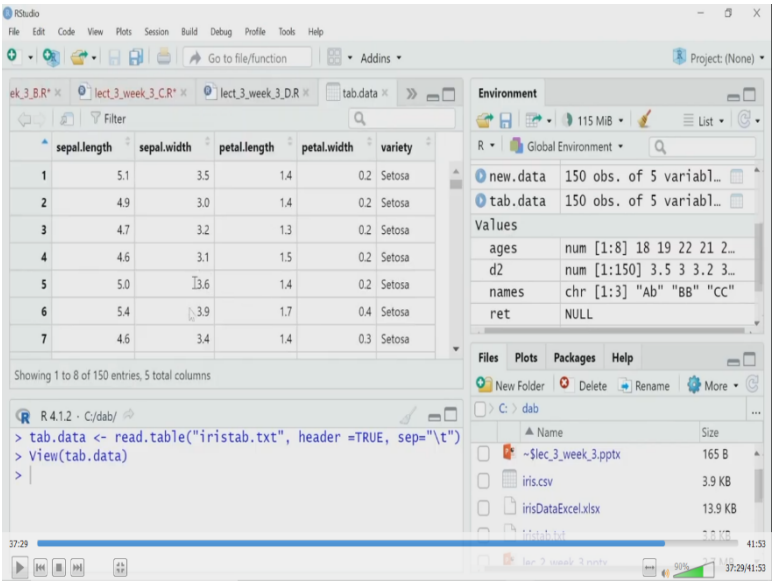
tab.data ← read.table("iristab.txt", header = TRUE, sep = "\t")

And notice here again I am putting the file name inside the apostrophe. Do not miss the name of the extension. Then I have to specify whether the header is there in my data or not. So, I have header in my data. So, I am writing header equal to TRUE. And the third operator argument is very important here.

The third argument here is separator. Means how the columns of the data is separated. In csv, by default, the separation is by comma. But, this is a read table generic function. So, it can read different types of data format. So, I have to tell it that my data in iristab.txt file is tab delimited means the columns are separated by tab.

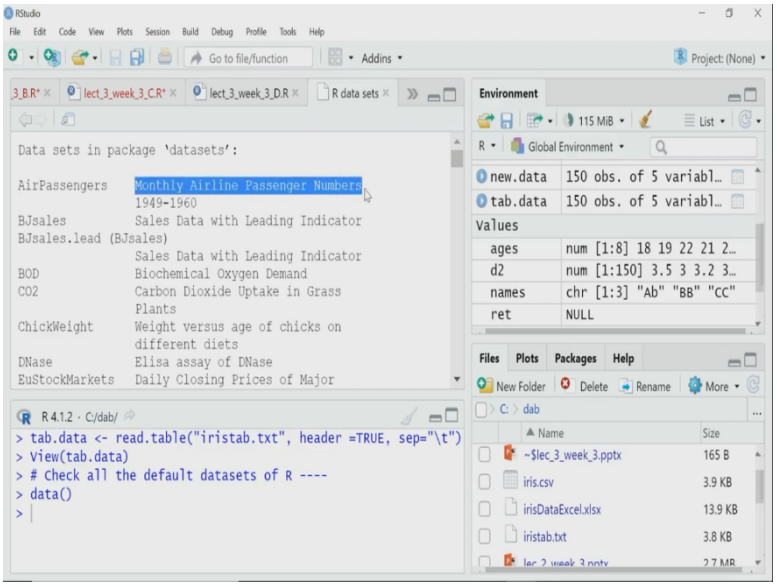
So, that is why I have written separations sep, sep equal to this \t, inside the apostrophe. So, if the R will understand that this table I have to read as a tab delimited table and I can run it. So, now, I have a new variable created. I have assigned this whole thing to tab dot data here.

(Refer Slide Time: 37:28)



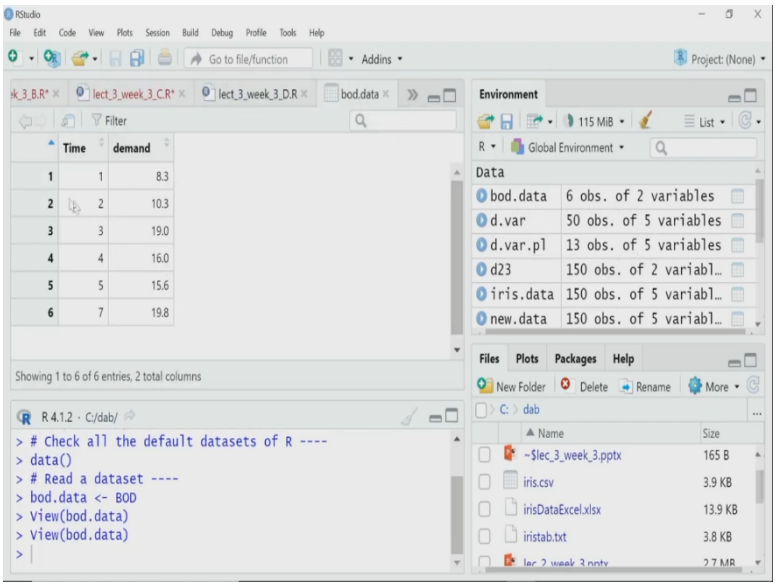
So, tab.data if I open, I can see the same data. Now, there is a very interesting thing which is very useful for students. R, when you install R, by default some data set which are commonly used for learning statistics, machine learning, machine learning are installed in your machine. So, this is the default data set.

(Refer Slide Time: 37:49)



You can actually check which are the default data set there in your machine. So, for that use a data function, just data with the round bracket. And it will give you a list. So, I have AirPassenger data which is data of monthly airline passenger, numbers from 1949 to 1960. Then you have the data for life cycle saving, all sort of data set. And you can call this data set and assign that data set to a variable and then perform your analysis. For example, what I will do?

(Refer Slide Time: 38:23)



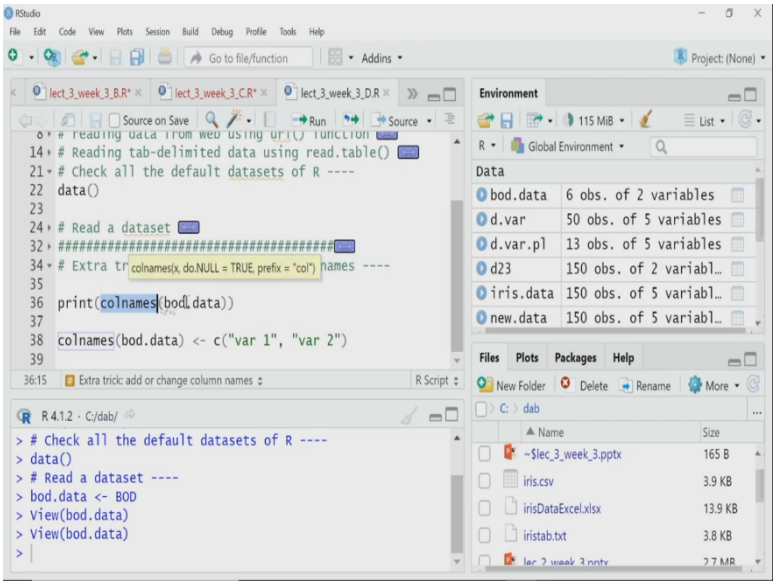
I will call this bod data, bod data to an assign that to a variable called bod.data using this assignment symbol. So, let me assign it. So, now, I have got a new variable bod.data. And if I

open that double clicking, it has 6 observations, 6 rows and it has 2 columns, Time and Demand.

So, Time is one variable, Demand is another variable. So, now, I can work on this data and perform analysis. So, this is very handy. If you are learning statistics and machine learning, you have not to go around finding and fetching data, you can use this readymade data set which you have already installed in your machine.

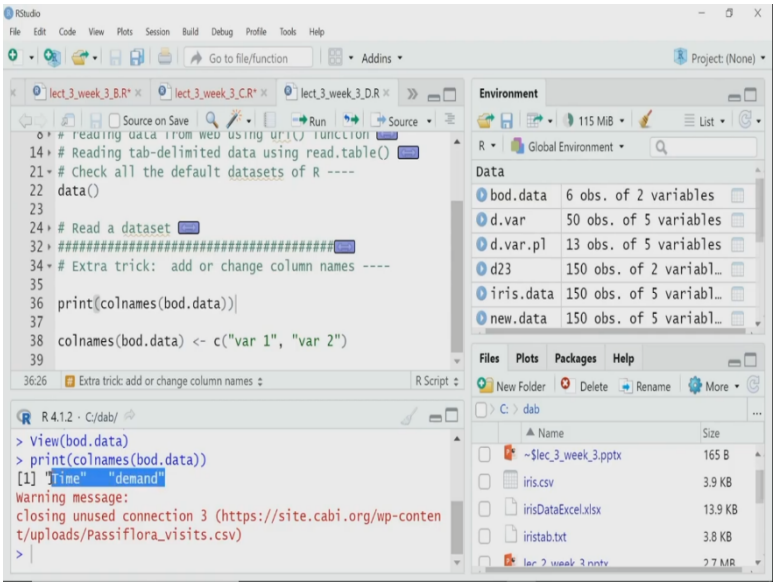
Now, this brings me to end of this lecture where we have learned how to create one-dimensional data, like a vector or a list and then we have learnt how to read a data file which is a table format then extract the data set and then write it. I will end it with a trick. Suppose, you want, you have extracted the data or you have imported the data and you want to change the name of the variable, that is you want to change the name of the header or suppose, your data set does not have header, you want to put the header using R. How to do that?

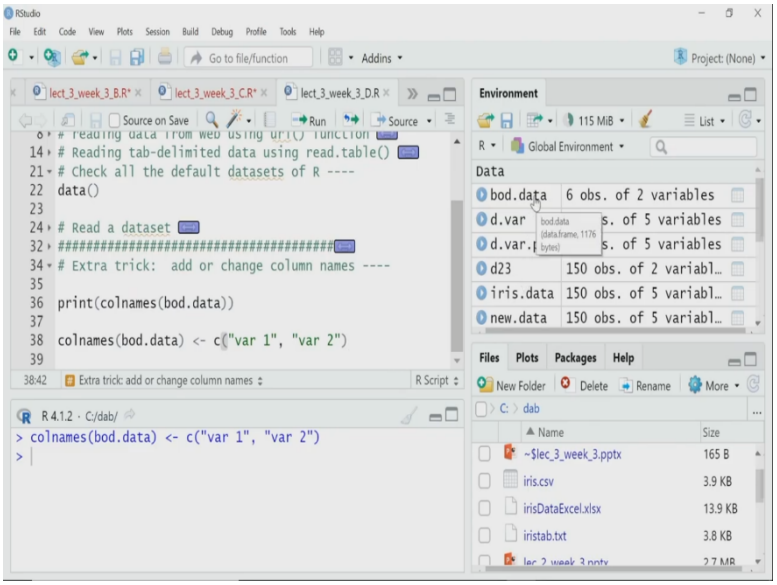
(Refer Slide Time: 39:47)



So, that is what I will do here. So, what I have to use? I have to use the function called column names. So, column names give me name of the column. Just now, I have imported the bod data as a variable called bod.data. So, let me extract that, extract other column names and print them, so that is why I have written print column names of bod data. So, if I run that. So, what I have got?

(Refer Slide Time: 40:18)





print(colnames(bod.data))

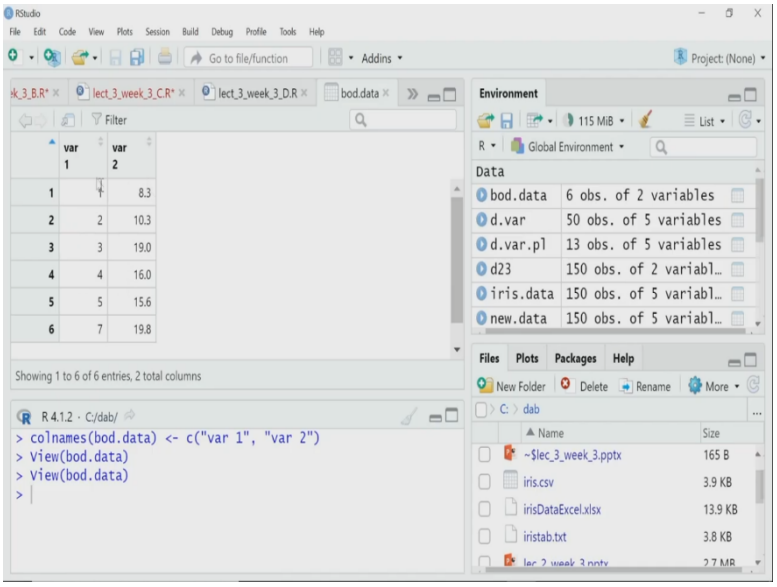
colnames(bod.data) ← c(“var 1”, “var 2”)

I have got names are time and demand. So, these are the two variable names. Now, I want to change this name. From time and demand, I want to change and put their name, some other name, for example, I want to call one of them variable 1, the other one is variable 2.

So, what I am writing? I am creating a list using the combination function c. So, c, var 1, var 2, these are the new name, obviously, in apostrophe. And I want to assign this list, to what? I want to assign these lists to the name that have been extracted now, using column names

function. So, if I run that, it has been done. Let me go back and check the bod dot data file again, variable again.

(Refer Slide Time: 41:06)



Now, you can see the variable names has changed. Earlier this were time and demand. Now, they are variable 1 and variable 2. This is the way you can actually add a new variable name, you add a new header or you change the names in the header, the variable names in the header. That is all for this video. Thank you for learning with me today. See you in the next video.