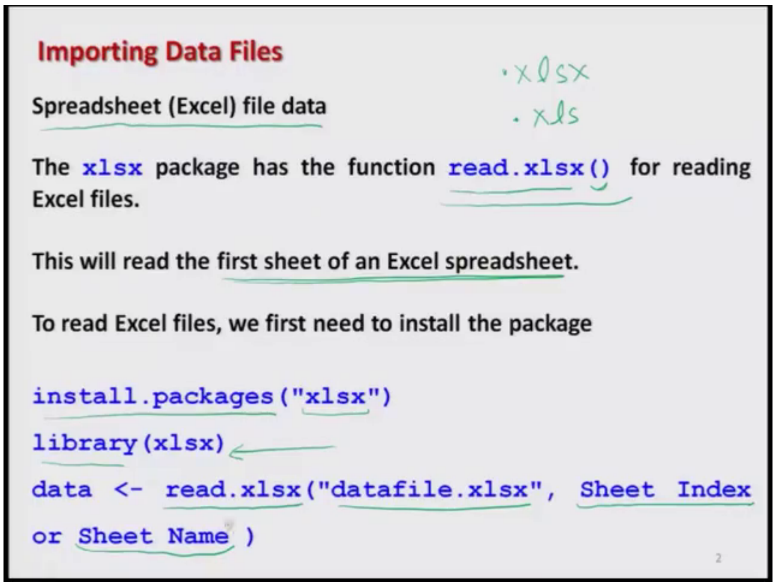
Introduction to R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 34
Importing Data Files of Other Software and Redirecting Output

Welcome to the next lecture on the course introduction to R software. You may recall that in the earlier lecture, we had discussed the aspect; how to import data sets from some external sources, right and we had discussed about different types of files structure like as dot CSV dot t x t and we discussed how to import them in your R software. Now we are going to continue our discussion and we are going to learn that how one can import a data set that was created in some other software, right.

(Refer Slide Time: 00:58)



Importing Data Files

Spreadsheet (Excel) file data

The `xlsx` package has the function `read.xlsx()` for reading Excel files.

This will read the first sheet of an Excel spreadsheet.

To read Excel files, we first need to install the package

```
install.packages("xlsx")
library(xlsx)
data <- read.xlsx("datafile.xlsx", Sheet Index
or Sheet Name )
```

So, let us start our discussion. So, first source which I am going to take is say how to import the data from a spreadsheet and one of the important and popular package to create a spreadsheet is say Microsoft Excel software, right and in that case the extension of the file is dot x l s x or this can also be dot x l s in the earlier versions of Microsoft Excel software.

So, here the question which we are going to address is how to import a data which is created in Excel software and has got an extension dot x l s x; actually dot x l s was the extension in the earlier version of the Microsoft Excel software. So, in order to read a file which is created in Excel is package, we have a command read dot x l s x and then inside the arguments we have to write down the file name, but when I try to use this command, this is going to read only the first sheet of the Excel spreadsheet when we are trying to create an Excel sheet in the Excel software, then it is possible to create different sheets inside the same file.

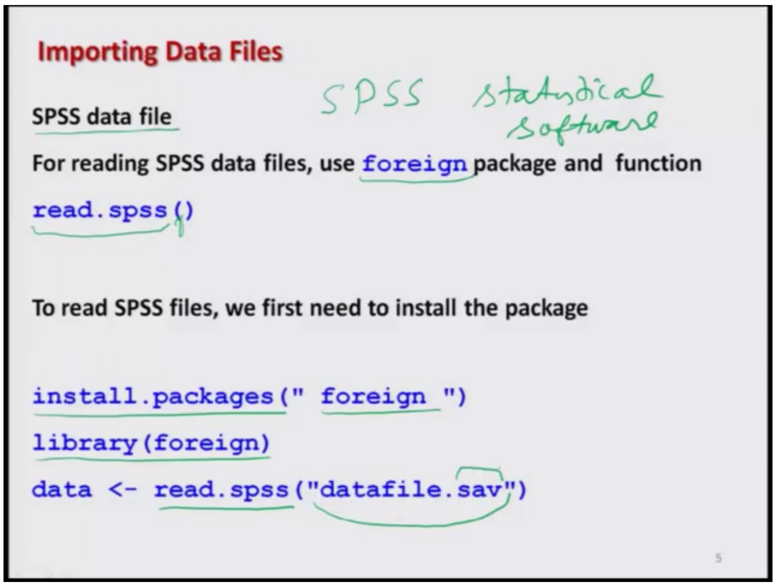
So, the objective can be whether to read a particular a spreadsheet which can be identified by the name of the spreadsheet or the number of the spreadsheet say spreadsheet number equal to 1, 2, 3, 4 or say its name. So, when I am not giving here any further extension and I am simply writing read dot x l s x, then it will read only the first sheet of an Excel file. This is an important point which you have to keep in mind, right, but in order to read the files from this package first step is to install the package whose name is x l s x and after this, you have to load this package and only then; you can read a file. So, after installing and uploading this package then we can use this command read x l s x, then name of the data file and after this you have to give the option whether you want to read the first sheet or say any other sheet.

So, in the next slide, I will show you how to give this option. So, now we have 2 option that suppose I want to read the sheet number 2 or I want to read a particular sheet which has got a name. So, in order to read a particular sheet, I have to use 2 options either sheet index; try to see here this all s h e e t; they are in small letters lowercase alphabets, then I is in the uppercase that is capital and all other letters are small or I can use the sheet name here in this case all letters except capital N; they are n small alphabets. So, sheet index is used to specify the number of the sheet and sheet name is used to specify the name of the sheet which we want to study inside that file. So, in that case the complete syntax will be read dot x l s x, then name of the data file that you want to read inside the double quotes then you have to write sheet index and then you have to write which file you want to read.

So, for example, here I have given here 2; that means, I want to read the sheet number 2 and similarly, in case if you want to read the sheet and you want to specify the name of the sheet, then the same command continues, but instead of here sheet index, I am using here the option sheet name is equal to for example, say marks and this is enclosed by the double quotes. So, if you want to read any particular sheet you can specify the number or the name of the sheet and using this package you can read this command there is always a possibility that you would like to read the file of m s Excel software that was created in the earlier version and in earlier version the extension of the files should used to be x l.

So, in order to use the older Excel files in dot x l s format, we have to use another package g data and then I have to use the command read dot x l s and rest everything remains the same, but in case if you try to use this read x l s this will again only read the first sheet of the file. So, in order to read a file of the format dot x l s; so, the first step is that we have to install the package called as g data after the g data package has been installed we need to upload it. So, I use the command library g data and this can be done exactly in the same way as we did in the earlier case and after this you want to read the file. So, for that use the command read dot x l s give the name of the file and here you have to specify the sheet index or sheet name exactly in the same way as we did in the case of dot x l s x format.

(Refer Slide Time: 07:35)



Importing Data Files

SPSS data file

SPSS statistical software

For reading SPSS data files, use `foreign` package and function

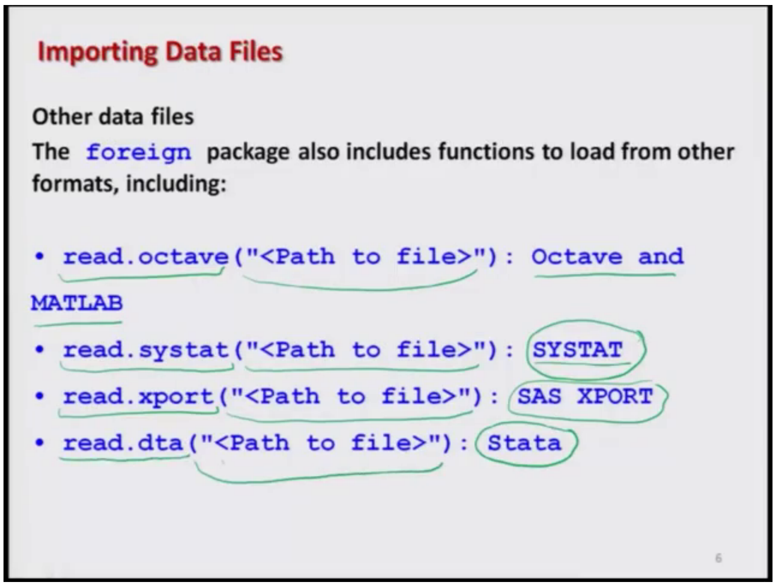
```
read.spss()
```

To read SPSS files, we first need to install the package

```
install.packages(" foreign ")
library(foreign)
data <- read.spss("datafile.sav")
```

So, that is pretty simple. Actually now we take another source from which the data can come suppose we are going to read a SPSS data file; SPSS is a very popular statistical software right in order to read a data file that is created in the SPSS package we need to install a special package which is called as foreign f o r e i g n; all in small letters. So, first we install this foreign package and then we use the function read dot SPSS and we give the name inside this argument. So, in case if you want to read a data file from SPSS package first step is to install the package foreign using this command then load this package using library command and then simply try to use the read dot SPSS command and inside the argument enclosed by the double quotes try to write down the name of the file.

(Refer Slide Time: 09:03)



Importing Data Files

Other data files

The `foreign` package also includes functions to load from other formats, including:

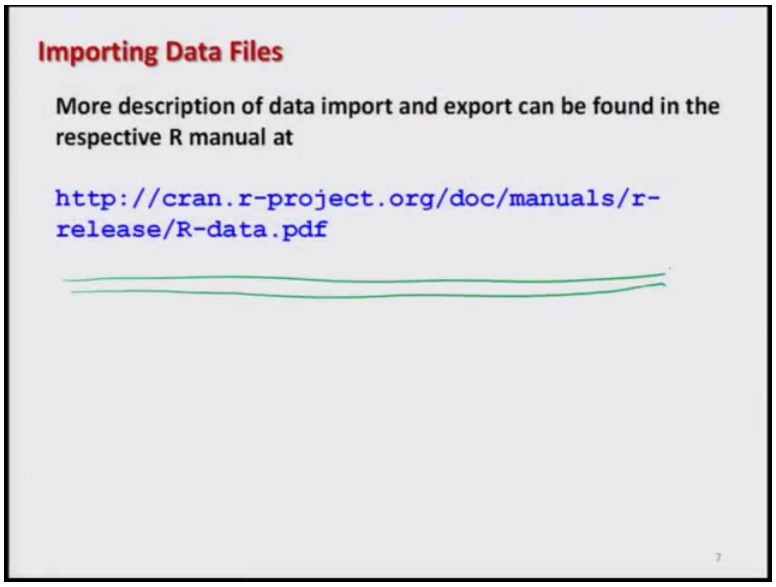
- `read.octave("<Path to file>")`: Octave and MATLAB
- `read.systat("<Path to file>")`: SYSTAT
- `read.xport("<Path to file>")`: SAS XPORT
- `read.dta("<Path to file>")`: Stata

6

And in the SPSS the extension of the data file is dot s a v, right. So, that is pretty simple exactly in the same way and there are some other options also to read the data file from some other software also and for that usually the package foreign works, right. So, for example, in case if you want to read the file from octave and MATLAB software, then the command is read dot octave and then inside the argument you have to give the name of the file including the path inside that double quotes. Similarly, if you want to read data file which is generated from this software SYSTAT, this is another statistical software, then you have to use the command read dot SYSTAT and then you have to give the file name and its path inside the double quotes in the arguments and then you can read this thing.

Similarly, if you want to read data file from say software SAS; SAS is another a statistical software a statistical analysis system and in that case we use the command read dot XPORT; X P O R T, right and then the same format inside the argument you have to specify the name of the file along with its paths inside the double quotes and similarly there is another statistical software what is called as a Stata. So, if you want to read a data file that was created in the software Stata, you just use the command read dot d t a and inside the arguments try to give the name of the file and its path inside a double quotes right.

(Refer Slide Time: 10:43)



Importing Data Files

More description of data import and export can be found in the respective R manual at

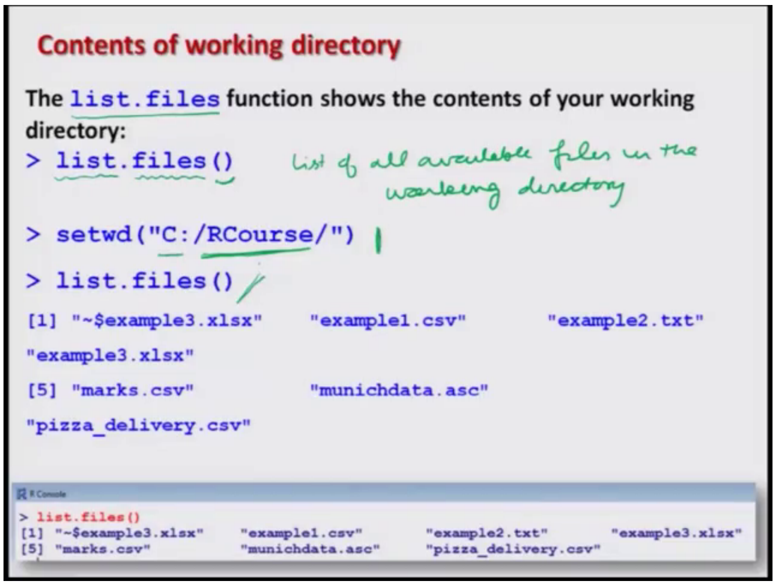
http://cran.r-project.org/doc/manuals/r-release/R-data.pdf

But here you can see what are we trying to do there are different resources which are trying to generate the data files and our objective is to read them inside the R software well it is difficult to know that which format can be read in the R software and which of the package has to be used. So, I would suggest you that whenever you want to read a particular type of data set please try to use the help and try to see what is the command to read the data and what additional package has to be installed and once you know this thing, then the steps are very simple first step install the package second step load the package using library command and third step is use the proper command to read the data file.

And then you can read it in case if you want to have some more description on the data import and export that can be also found in the R manual which is located at this data set on the website of the R project. So, means if you want to have a specific thing you can always read it from here. So, after learning that how we can read the data from different sources or in different formats the next objective is that whenever we are trying to run the program; how that program can be saved one option is that that we can see the outcome of the program on the screen and the second option is that we would like to save the outcome of the program inside a file and that file can also be of different type like a TXT or CSV or some a format.

So, now in the remaining lecture I am going to illustrate how one can save the outcome of a program inside a file, but in order to understand it first we also need to know that how to see the contents of the working directory because whenever you are trying to save the a file in a directory, you would also like to check whether that file is there or not or what are their contents.

(Refer Slide Time: 13:32)



Contents of working directory

The list.files function shows the contents of your working directory:

```
> list.files()
```

List of all available files in the working directory

```
> setwd("C:/RCourse/")

> list.files()
[1] "~$example3.xlsx"    "example1.csv"       "example2.txt"
"example3.xlsx"
[5] "marks.csv"          "munichdata.asc"
"pizza_delivery.csv"
```

```
> list.files()
[1] "~$example3.xlsx"   "example1.csv"     "example2.txt"        "example3.xlsx"
[5] "marks.csv"         "munichdata.asc"   "pizza_delivery.csv"
```

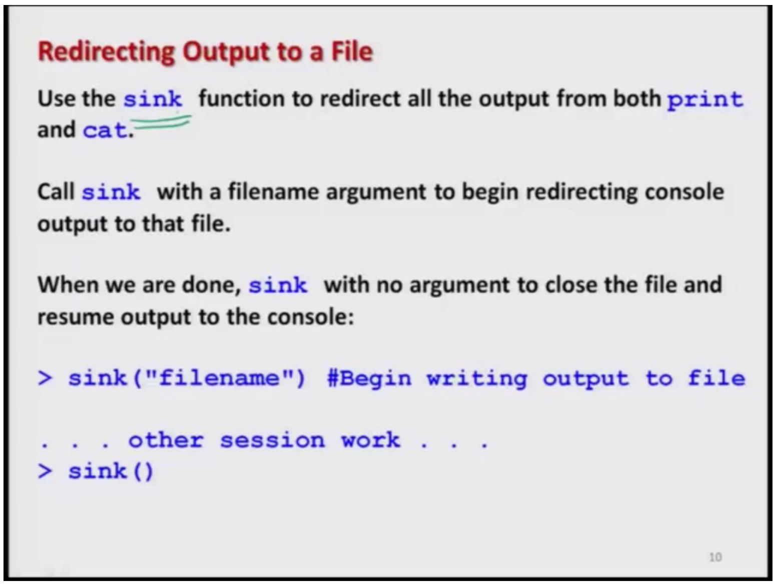
So, first of all I try to explain you here that how we can see the content of the working directory. So, in order to see the contents of the working directory we have a command here list dot files. So, we try to write down here l i s t dot f i l e s files and the arguments. So, when we try to do so.

This provides us the list of all available files in the working directory right now if you recall as some lectures back we had the constructed a directory for our course and this directory was located on the c drive and the name of the directory was R course. So, what I am going to do here that first I am going to set the working directory and then I would try to use this command list dot files and argument to see the contents of the file. So, you can see here that when we try to do it on the R console over here. So, first I tryto change my working directory and then I try to say here list files and you can see here that we have these many files and these many files if you try to see here on the R course directly also we have the same files, right.

So, yeah; I mean the font size will be little bit here small, but you still you can believe on me that these are the same files which are available over here right and yeah here also I have given the outcome and the screenshot. So, you can read it over here right. So, here is the outcome of the directory R course, but definitely this outcome was taken on a different computer. So, this will be a little bit different with what I have shown you here, but here is the screenshot of the same thing. So, that is a pretty simple thing to understand and now I come back to our main topic if you are going to cover in the remaining part of this lecture.

So, now my objective is that I am trying to run the program and I want to redirect the outcome of the program inside a file as I said whenever we are trying to run the program there is going to be an outcome and the outcome can be seen either on to the screen or that can be save to a file. So, now, first we try to see; what are the different possible commands to see the outcome and then we will try to see how we can redirect them towards a file. So, if you remember we had done a function here c a t cat and cat was used to demonstrate the outcome in a particular format, right.

(Refer Slide Time: 16:30)



Redirecting Output to a File

Use the `sink` function to redirect all the output from both `print` and `cat`.

Call `sink` with a filename argument to begin redirecting console output to that file.

When we are done, `sink` with no argument to close the file and resume output to the console:

```
> sink("filename") #Begin writing output to file

. . . other session work . . .
> sink()
```

10

So, for sample; if I try to take care of a very simple example; suppose, I try to add 6 and 8. So, and I try to save this outcome inside the new variable say a and s.

And suppose I want to print the outcome of a program something like the answer of six plus 8 is mean this 6plus 8 is going to be in the form of a string means a character that is not going to be added and then whatever is the value of this variable a n s and then by this backslash n inside the double quotes that is indicating that the next outcome is going to be on the next line. So, this is indicating a change of line that is newline and after that we have to write down the file is equal to and inside the double quotes I have to give the name of the file in which I want to store the outcome or the name of the file where I would like to store the outcome of the program and after this whatever is the output that will be saved in this file that is located inside a working directory, right. So, if you try to do it here in order to do it we need one more function.

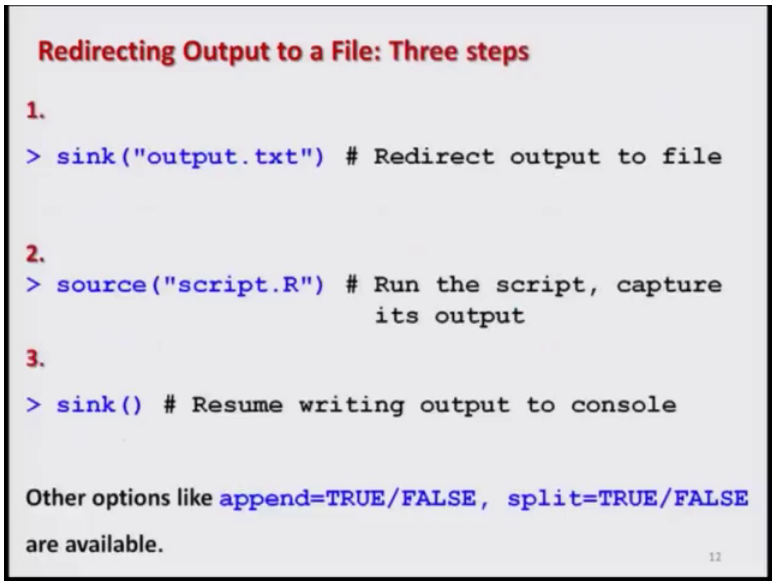
And this function is sink; this function sink helps us in redirecting the outcome of your here cat inside a file; cat function cannot work alone. So, this is how we are going to do it here. So, just for the sake of your understanding you may recall that we had done 2 functions print and cat to get the outcome in a particular format they had their own restrictions, but now we have learnt what is the difference between the commands print and cat. So, now, if I want to redirect the outcome of a cat; so, what we have to do here first that I have to call the sink function and I have to inform the R that; what is the file or what is the name of the file in which the outcome is going to be stored.

So, in order to do that thing we try to write down say here this command sink s I n k and inside the argument inside the double quotes I try to write down the name of the file here in which I want to store the outcome right and after that we try to execute the program over here; whatever is my program that is executed here and after that the outcome is saved to do this thing and after that we try to write once again sink and inside the argument there is blank.

That means, sink arguments this function will help us in changing the mode; that means, as soon as I give here the sink after this whatever is the outcome of my program that will always be diverted towards the file, but once I have done my job, then if I try to give here sink and this arguments, then again I am trying to inform the R software. Now please stop giving the outcome of the program inside the file, but now show the outcome of the file only on the R console that is on the screen.

So, this is a command to stop delivering the output to the file name go to the file whose name has been given earlier now I would like to take an example to illustrate this concept, but before that let me just briefly clarify the difference between the print and cat Functions.

(Refer Slide Time: 21:25)



Redirecting Output to a File: Three steps

1.

```
> sink("output.txt") # Redirect output to file
```

2.

```
> source("script.R") # Run the script, capture
                       its output
```

3.

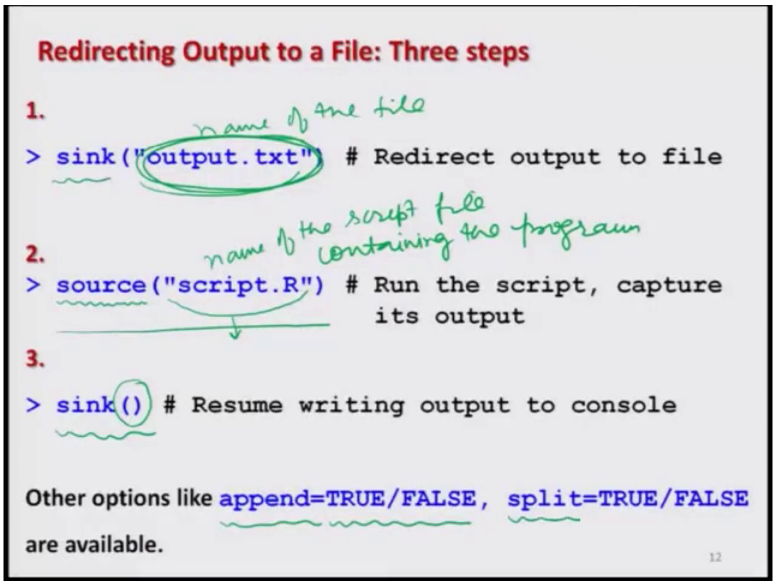
```
> sink() # Resume writing output to console
```

Other options like `append=TRUE/FALSE, split=TRUE/FALSE` are available.

12

So, if you try to recall the print and cat these are the functions which are use to write down the output of a program, but both these functions were used to write the output to the console; console in simple language, I would say that this is the say screen; that means, whenever we try to run the program inside the R software; whatever the output is coming on the screen that is the outcome on the console of the R software. Now what is the difference between cat and print; the cat function has the capability to redirect the outcome or the output of a program to a file provided, we have supplied a file name whereas; this capability is lacking in the print function print function cannot redirect its output to a file or to any file. Now we are going to use here a function sink which will force the cat function to redirect the outcome to a file cat function cannot do it alone.

(Refer Slide Time: 22:39)



Redirecting Output to a File: Three steps

1.

```
> sink("output.txt") # Redirect output to file
```

2.

```
> source("script.R") # Run the script, capture
                       its output
```

3.

```
> sink() # Resume writing output to console
```

Other options like `append=TRUE/FALSE`, `split=TRUE/FALSE` are available.

name of the file
name of the script file containing the program
12

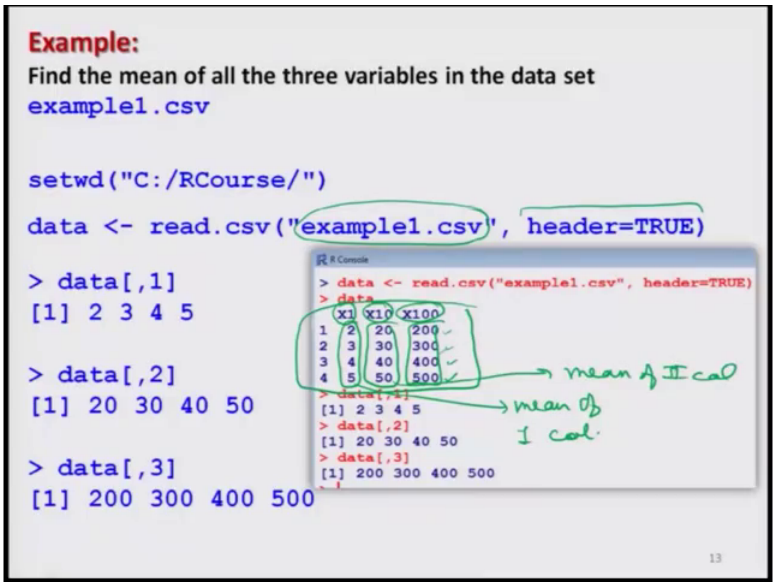
So, in order to do it there are 3 steps that we have to follow first of all I have to use the function here sink and inside the double quotes, I have to write down the name of the file in which we would like to store the outcome or in which we would like to redirect the outcome of a program now in the second step; I have to write down source s o u r c e and inside the double quotes I have to write the name of the file or the name of the script file containing the program right; that means, whenever we want to store the outcome first I need to write down my function inside the script file and that script file has to be saved inside the working directory that file which is containing the function that file is specified here the name of the file is mentioned here.

I have use the filename as a script dot R that is simply indicating the script of the program is written inside this file. So, as soon as I say here source and I try to give the program name; this will run the program or run the script and whatever is the outcome that is captured while running the program that is redirected to the file name the file name which is given here as say output dot t x t and once the program is over. Now we need to inform in the third step to my program. Now I do not want to redirect the outcome towards a file; please stop it and now start giving the outcome on the screen or the R console. So, in order to inform this thing to my R software, we will use here a function sink and then we write this argument.

So, this will inform the R software to resume writing the outcome to the R console when we are trying to do. So, there are some other options like as append that can be true or false. For example, if I am trying to run the program today and whatever outcome has been stored inside the file and tomorrow, I try to rerun the program, then whether the outcome should be appended or it should be overwritten on the outcome of the earlier day and so on. So, these options are there which we have to specify by writing true or false and then whether there is going to be split or not these several options are there.

But I am not going into that detail, but definitely I would like to take here a very simple example to illustrate all these things. So, what I try to do here I am trying to use a data set that we have created earlier and then I would simply try to write a small program although; we have not done this part, but here my objective is not to tell you how to write the program, but simply to show you that whatever is the outcome of the program that can be stored inside a file right.

(Refer Slide Time: 26:15)



Example:
Find the mean of all the three variables in the data set `example1.csv`

```
setwd("C:/RCourse/")

data <- read.csv("example1.csv", header=TRUE)

> data[,1]
[1] 2 3 4 5

> data[,2]
[1] 20 30 40 50

> data[,3]
[1] 200 300 400 500
```

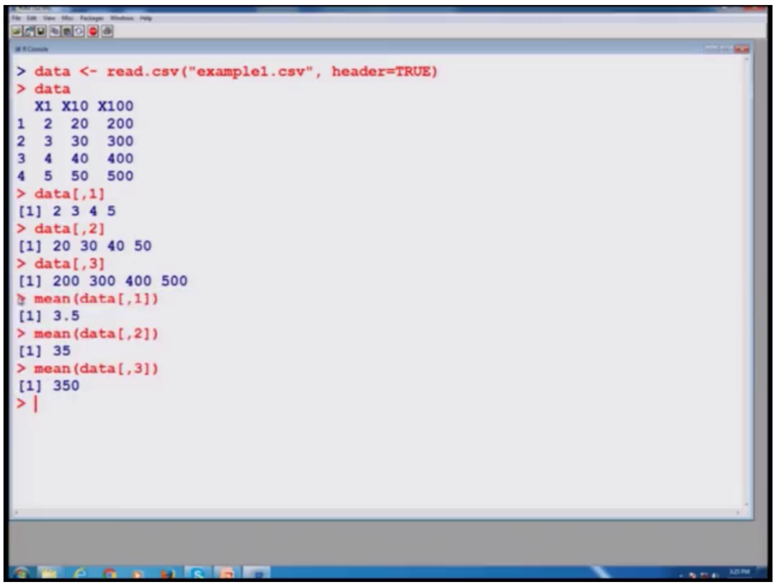
```
> data <- read.csv("example1.csv", header=TRUE)
> data
  X1 X10 X100
1  2  20  200
2  3  30  300
3  4  40  400
4  5  50  500
> data[,1]
[1] 2 3 4 5
> data[,2]
[1] 20 30 40 50
> data[,3]
[1] 200 300 400 500
```

So, if you try to remember some lectures back we had created a file example one dot c s v, right and in this file we have 3 variables that we had given as say here x 1, x 10, x 100 and they were containing 4 value like as 2, 3, 4, 5, 20, 30, 40, 50, 200, 300, 400 and 500.

And here I am using the option header is equal to true for the first line of this a data has been sacrificed as the header, right. So, in this file, if you try to see here this is my data set, I will try to show you on the R console also, but here if you try to see there is first column second column and third column; what I want to do is the following; I simply want to find out the means of the data set in the first second and third column. So, what is my objective? I want to find out the mean of say here first column then mean of second column and the mean of third column well I can do it separately also, but I would like to write a program in which I would like to store all the 3 means inside the file directly right. So, you can see here these 3 columns can be recalled by the names data inside the square bracket I have to write 1, 2 and here 3.

So, you can see here these are the data set which are coming over here and my objective is simply I want to find out the mean of this data set 2, 3, 4, 5 mean of this data set 20, 30, 40, 50 given here and the mean of this data set that is 200, 300, 400 and 500. So, before we try to do it, let us try to see this data set, right. So, I try to read this data file on the R console, right.

(Refer Slide Time: 28:27)

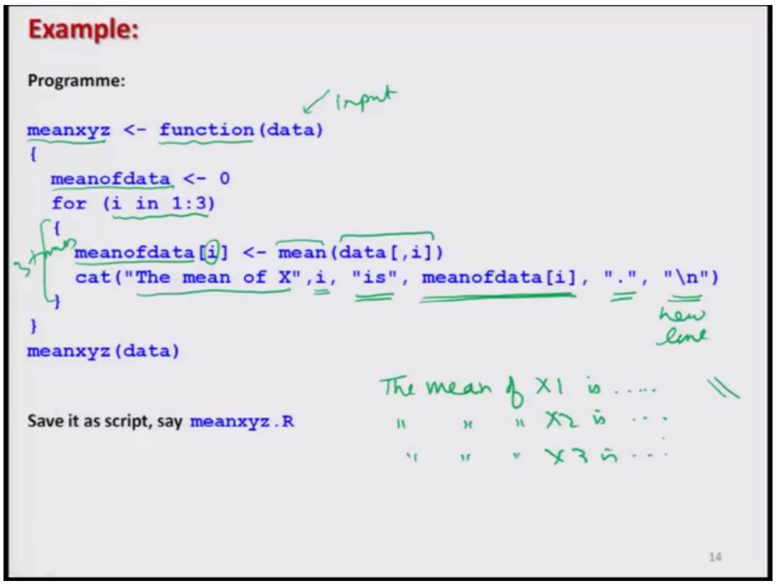


```
> data <- read.csv("example1.csv", header=TRUE)
> data
  X1 X10 X100
1  2  20  200
2  3  30  300
3  4  40  400
4  5  50  500
> data[,1]
[1] 2 3 4 5
> data[,2]
[1] 20 30 40 50
> data[,3]
[1] 200 300 400 500
> mean(data[,1])
[1] 3.5
> mean(data[,2])
[1] 35
> mean(data[,3])
[1] 350
> |
```

So, you can see here, this is my here the data right and now when I try to write down here data say one with. So, this gives me the first column this gives me the second column and this gives me the third column, right and now in case if you try to find out here the mean of this data, I can also find it out in this particular mean of data one; this is 3.5 mean of data 2 as a 35 and say mean of here data 3 as a 350.

But here if you try to see in this highlighted part, I have tried to find out the mean of the 3 columns, but that I have done manually one by one. Now I want to write down a small program to give me the outcome in a single shot more ever if you try to see the outcome of these 3 commands mean of data 1, data 2 and data 3 that is coming on my screen this is my R console and I do not want this data to be here on the screen, but I would like to save it inside a file. So, now, what I try to do here is the following right.

(Refer Slide Time: 29:43)



Example:

Programme:

```
meanxyz <- function(data)
{
  meanofdata <- 0
  for (i in 1:3)
  {
    meanofdata[i] <- mean(data[,i])
    cat("The mean of X",i, "is", meanofdata[i], ".", "\n")
  }
}
meanxyz(data)
```

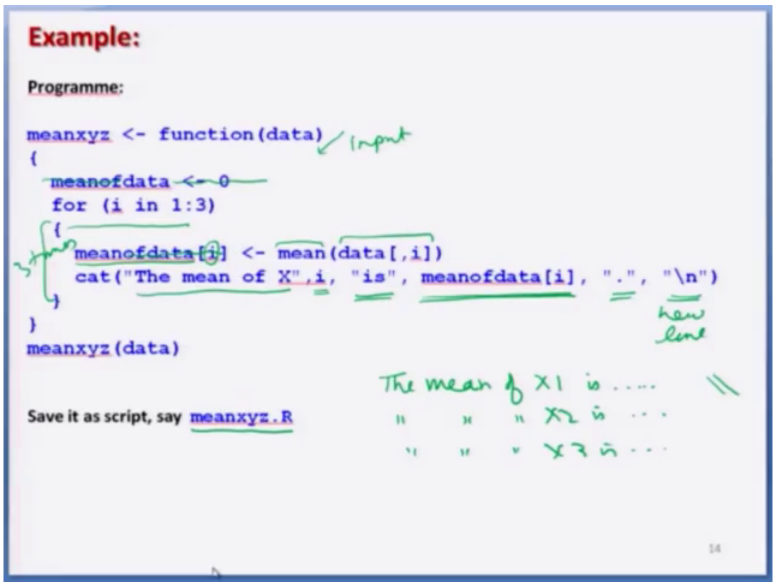
Save it as script, say `meanxyz.R`

I try to write down here a program say here whose name is mean x, y, z and this is given as a function and the input is coming from this here file data and I am; I simply try to define here a variable mean of data to be zero just to initialize the program and then I try to use here a loop in which I am saying that for i in 1, 2, 3; that means, I have a 3 columns data 1m data 2m data 3.

So, I would try to denote them by say here data comma i. So, I am trying to say here please try to read here the data column say data this bracket comma i and then try to find out the mean of this column and whatever is the mean of this i th column that has to be saved inside a new variable which is here mean of data and it has to be stored in the i th position and after that this program will run from i goes from 1 to 3. So, the program is going to run for 3 times and then there are going to be 3 outcomes mean of data one mean of data 2 and mean of data 3 and I want to express the outcome in this particular format, I want to write the mean of x and then I would write to write down 1, 2 or 3 that is x 1, x 2 or x 3 and then as a corrector I would say is and then whatever is the value of this here mean of data I it has to be printed here.

And then after this; there has to be a full stop and the next outcome has to be on the new line so; that means, I want to write down here something like the mean of x 1 is whatever is the value then there is a change of line and then and then the outcome will be the mean of x 2 is this whatever is the value, then the on the third line the mean of x 3 is like this. So, I try to write down this program and then I try to save this entire program inside a script file which is here mean x, y, z dot R right and then I try to do this thing.

(Refer Slide Time: 32:10)



Example:

Programme:

```
meanxyz <- function(data)
{
  meanofdata <- 0
  for (i in 1:3)
  {
    meanofdata[i] <- mean(data[,i])
    cat("The mean of X",i, "is", meanofdata[i], ".", "\n")
  }
}
meanxyz(data)
```

Save it as script, say meanxyz.R

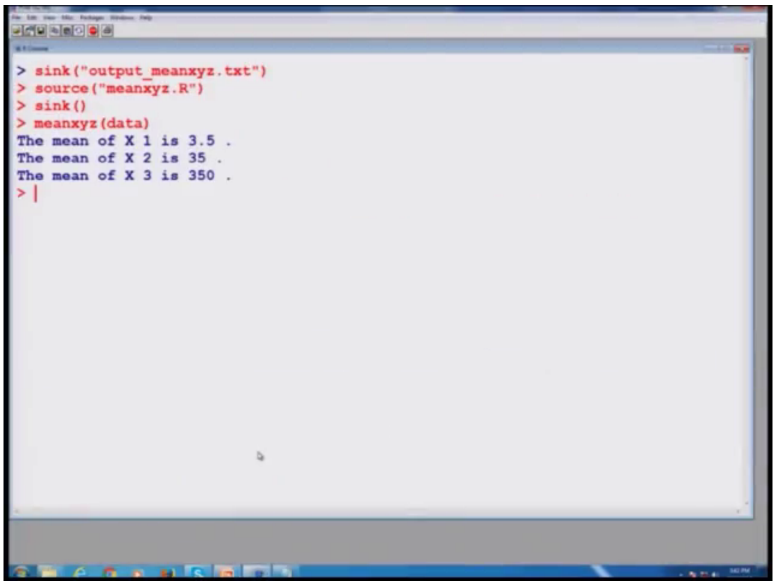
I try to create a file whose name is output underscored mean x y z dot t x t in which I am going to store the outcome this is a file which is going to be created to store the outcome of this program and now after this command here sink then please try to check this will create a blank file in your working directory.

And before doing it, this file will not be there that you can check yourself. Now after this I try to write down the second step source and I try to give here the name of the file in which I have written the program this file name is mean x y z dot capital R that is a R script file; do not worry, I will try to show you on the R console also as soon as I execute this command; this will write the output inside the file which file this file here, I have 2 option either I can use say source this thing or I can also write down the program say mean x y z data and it will also write down the outcome over there and then in that third step, I will write down here say sink and then this argument and this will resume writing the output to the console. So, now, let us try to do all this operation on the R console.

So, what we try to do here that first I try to create here a script file. So, I try to copy this function over here now. So, first we try to open here new script file. So, this file is you see here and I am trying to copy the same program over here. Now you can see here in this the directory there are here only 8 files and there is no file say call as mean x y z dot r. So, now, this file is here and I try to save this file here as a say mean x y z mean x y z and I try to save it in this directory over here. So, now, you can see here now we have here 9 file and this file what we have created here this is here which I here highlighted right and now I try to save this file now this file is save and I try to come on my first step that is I need to specify that where the outcome of this file is going to be stored.

So, I try to create here a file output underscore mean x y z dot t x t and you can see here that this file is not present in this working directory. So, as soon as I try to do here enter; now you can see here that this file is created over here this is here and you can see here that this file here is blank there is nothing. So, I try to close it here and then I am trying to run the program by writing here source and the name of the program.

(Refer Slide Time: 35:42)



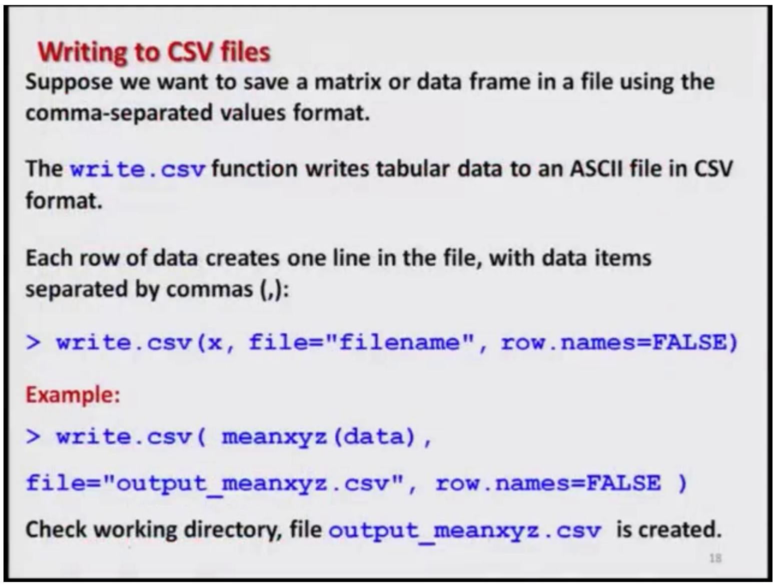
```
> sink("output_meanxyz.txt")
> source("meanxyz.R")
> sink()
> meanxyz(data)
The mean of X 1 is 3.5 .
The mean of X 2 is 35 .
The mean of X 3 is 350 .
> |
```

So, the name of the program is mean x y z dot R. So, I try to write down source and inside the double quotes, I try to write down the name of the R script which I want to run and as I try to give it here say enter there is no output here, but now if you try to see in this R working directory if you try to open now the same file; this file has this outcome; yes, I can show you here more clearly, right.

So, you can see here the outcome has come here the mean of x 1 is 3.5 the mean of x 2 is 35 and the mean of x 3 is 350. So, you can see here that this outcome has been written on this function. Now in case if you want to inform the R; now that please do not send anymore outcome to this file. So, you have to now say here sink and then arguments and after this, you will get here and after that you will inform the R console to not to send any outcome to the file, right. So, here I have shown you now that how you can redirect the outcome to the file and here you see I have given you the screenshot of all those things first, I try to change the directory and then I try to create a file name and if you try to open this file in the R course directory you will get this outcome.

And this is the screenshot which I have just shown you and this is another the screenshot of the same thing that first I try to write down here the program, then I try to see what is my program and then if I try to run the program over here, I will get this outcome right you can also see it here that if I try to write down here mean x y z dot data you will get here this outcome, right, now we come back to our slides and now in case if you want to write the outcome on a different type of file.

(Refer Slide Time: 37:54)



Writing to CSV files

Suppose we want to save a matrix or data frame in a file using the comma-separated values format.

The `write.csv` function writes tabular data to an ASCII file in CSV format.

Each row of data creates one line in the file, with data items separated by commas (,):

```
> write.csv(x, file="filename", row.names=FALSE)
```

Example:

```
> write.csv( meanxyz(data),
file="output_meanxyz.csv", row.names=FALSE )
```

Check working directory, file `output_meanxyz.csv` is created.

18

For example, suppose you want to write the outcome in a c s v format then briefly I can explain you here that the command to write the outcome in the c s v file is write dot c s v and the advantage or the feature of the c s v file is that that every row of the file will indicate one set of outcome of the data, right.

And those outcomes are separated by commas. So, the command to give the or to redirect the outcome towards the c s v file is that you try to use the command right dot c s v, then you try to give whatever the file name, you want to give where you want to store the names and then whether you want to have row names or not the different types of options are there. So, I am not going into that much detail, but I am simply trying to tell you that this option is also there for example, in the same example that we have done if I want to write down the outcome inside a c s v file, then the command will be right dot c s v and what outcome, we want to write means our script that is mean x y z and inside the argument the data on which I want to execute the program mean x y z and then the file.

For example output underscored mean x y z dot c s v that is the file name in which I want to store the outcome and here I want I do not want to give here the row name. So, that you have to give it here as a false and once you do this, then if you try to check the working directory, this will be created inside the R folder, right. So, now, in this lecture we would like to stop here and you may see that in this lecture, we have learnt that how we can import the data which is created in some other software and after that we also have learnt that whenever we are trying to execute the program, then how we can redirect the outcome to be stored inside the file.

Now, here I would like to conclude the topic on data handling and in the next lecture, we will try to come up with some other interesting topics; till then goodbye.